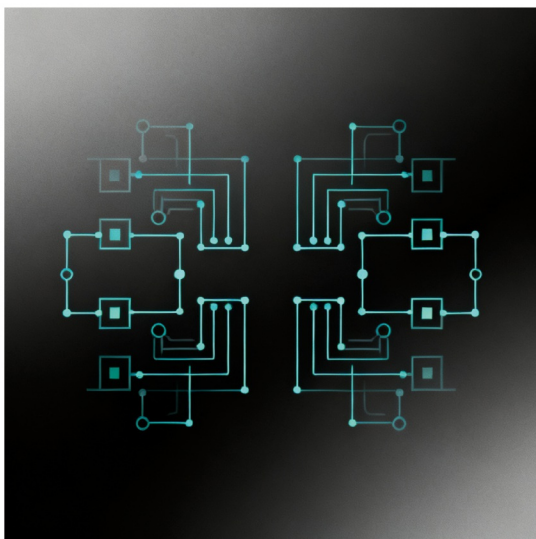


Lógica formal

Lógica proposicional, lógica de predicados,
lógica modal y lógicas no clásicas



Lucien Sina

Lógica formal

Lógica proposicional, lógica de predicados,
lógica modal y lógicas no clásicas

Lucien Sina

15.6.2026

Índice general

Prefacio	ix
0.1. Introducción y visión general	1
0.1.1. Motivación y relevancia	1
0.1.2. Contexto histórico	2
0.1.3. Objetivos de este libro	2
0.1.4. Procedimiento y estructura	3
0.1.5. ¿A quién está dirigido este libro? .	3
0.1.6. Objetivos de aprendizaje e indica- ciones de uso	4
0.1.7. Breves descripciones de los capítulos	6
0.2. Fundamentos	9
0.2.1. Notación y teoría de conjuntos . .	9
0.2.2. Estructuras y semántica	12
0.2.3. Principios de inducción	12
0.3. Estructura y estilo de los textos matemáticos	17
 1. Lógica proposicional	 21
1.1. Introducción y visión general	21
1.2. Sintaxis y semántica	24

1.2.1.	Introducción	24
1.2.2.	Signaturas	24
1.2.3.	Fórmulas	25
1.2.4.	Semántica: interpretaciones, satisfac- ción y modelos	26
1.2.5.	Tablas de verdad	29
1.2.6.	Observaciones	29
1.2.7.	Funciones de verdad	30
1.3.	Ejercicios con soluciones	33
1.4.	Inferencia	39
1.4.1.	Modelos como descripción de situa- ciones	39
1.4.2.	Operaciones sobre conjuntos de modelos	40
1.4.3.	Clases de satisfacibilidad de fórmulas	43
1.4.4.	Consecuencia semántica	47
1.4.5.	Clausura deductiva (¿qué se sigue de las premisas?)	48
1.4.6.	Teorías y la relación Th/Mod . . .	51
1.4.7.	Independencia respecto de átomos superfluos	54
1.4.8.	Breve lista de comprobación para argumentos semánticos	61
1.4.9.	Ejercicio breve	61
1.5.	Ejercicios y soluciones: razonamiento infe- rencial	62
1.6.	Equivalencia	66
1.6.1.	Fundamentos	66
1.6.2.	Equivalencias específicas	71

ÍNDICE GENERAL

1.6.3.	Formas normales	80
1.6.4.	Ejercicios y soluciones: equivalencia	91
1.7.	Consideración algorítmica	95
1.7.1.	Problemas de inferencia	96
1.7.2.	Cálculos: procedimientos sintácti- cos para inferir	100
1.7.3.	Cálculo de resolución	105
1.7.4.	Ejercicios y soluciones	118
1.8.	Solvers SAT	120
1.8.1.	Equivalencia de satisfacibilidad	121
1.8.2.	Codificación de Tseitin	125
1.8.3.	El algoritmo DPLL	134
1.8.4.	Ejercicios y soluciones	140
2.	Lógica de predicados	143
2.1.	Introducción y visión general	143
2.2.	Sintaxis y semántica	146
2.2.1.	Signaturas	148
2.2.2.	Términos y variables	152
2.2.3.	Fórmulas	155
2.2.4.	Semántica	161
2.2.5.	Ejercicios y soluciones	167
2.3.	Coincidencia, equivalencia y sustitución	170
2.3.1.	Lema de coincidencia	171
2.3.2.	Equivalencia	175
2.3.3.	Sustitución	193
2.3.4.	Clases de modelos	207
2.3.5.	Ejercicios y soluciones	212
2.4.	Formas normales y resolución básica	214

2.4.1.	Semántica para conjuntos de fórmulas	215
2.4.2.	Formas normales	218
2.4.3.	Modelos de Herbrand	231
2.4.4.	Teorema de Herbrand	238
2.4.5.	Resolución básica	246
2.4.6.	Ejercicios y soluciones	252
2.5.	Consecuencia lógica y resolución	253
2.5.1.	Consecuencia lógica	254
2.5.2.	Unificación	259
2.5.3.	Resolución en lógica de predicados	268
2.5.4.	Teorema de completitud de Gödel	278
2.5.5.	Clausuras deductivas y teorías . .	283
2.5.6.	Lema de Lindenbaum: existencia de teorías maximalmente consistentes	300
2.5.7.	La construcción de Henkin	303
2.5.8.	Teorema de Henkin	306
2.5.9.	El teorema de completitud de Gödel	311
2.5.10.	Del teorema de completitud a los teoremas de incompletitud	314
2.5.11.	Los teoremas de incompletitud de Gödel	329
2.5.12.	Computabilidad y complejidad . .	339
2.5.13.	Ejercicios y soluciones	344
2.6.	Lógica descriptiva	351
2.6.1.	Conceptos, roles y su interpretación	353
2.6.2.	Ejercicios y soluciones	388
2.7.	Lógica de predicados de segundo orden . .	391
2.7.1.	Sintaxis y semántica	392

2.7.2.	Propiedades y capacidad expresiva	398
2.7.3.	Lógica y complejidad descriptiva	414
2.7.4.	Ejercicios y soluciones	417

3. Lógica modal 422

3.1.	Lógica modal general	425
3.1.1.	Sintaxis y semántica	426
3.1.2.	Conceptos y propiedades	432
3.1.3.	Consecuencia lógica y cálculo de tableaux	441
3.1.4.	Ejercicios y soluciones	451
3.2.	Lógica temporal lineal	459
3.2.1.	Sintaxis y semántica	460
3.2.2.	Equivalencias y formas normales	465
3.2.3.	Relación con las estructuras de Kripke	477
3.2.4.	Distinción de estados en estructuras de Kripke	482
3.2.5.	Relación con la lógica de predicados	491
3.2.6.	Ejercicios y soluciones	497
3.3.	Lógica temporal ramificada	500
3.3.1.	Sintaxis y semántica	501
3.3.2.	Equivalencias y formas normales	509
3.3.3.	Bisimulación	523
3.3.4.	Relación con LTL	533
3.3.5.	Ejercicios y soluciones	539
3.4.	Lógica epistémica	545
3.4.1.	Restricciones de marcos	546
3.4.2.	Lógica modal para varios agentes	557

3.4.3.	Anuncios públicos	563
3.4.4.	Ejercicios y soluciones	568
4.	Lógicas no clásicas	573
4.1.	Lógica intuicionista	576
4.1.1.	Semántica de Kripke	582
4.1.2.	Propiedades	590
4.1.3.	Semántica basada en pruebas . . .	600
4.1.4.	Ejercicios con soluciones	619
4.2.	Lógica paraconsistente	622
4.2.1.	Lógica proposicional trivaluada . .	623
4.2.2.	Inferencia con modelos minimales .	635
4.2.3.	Lógica proposicional tetravalente .	646
4.2.4.	Ejercicios y soluciones	661
4.3.	Lógica difusa	664
4.3.1.	Conjuntos difusos	665
4.3.2.	T-normas	672
4.3.3.	S-normas	679
4.3.4.	Negaciones en la lógica difusa . . .	687
4.3.5.	Semántica e inferencias	697
4.3.6.	Ejercicios con soluciones	707
4.4.	Lógica por defecto	711
4.4.1.	Conclusiones no monótonas	712
4.4.2.	Sintaxis y semántica de la lógica por defecto de Reiter	715
4.4.3.	Inferencia y semimonotonía	730
4.4.4.	Ejercicios y soluciones	736
4.5.	Programación de conjuntos de respuestas	741
4.5.1.	Programas lógicos extendidos . . .	742

ÍNDICE GENERAL

4.5.2. Modelos mínimos	747
4.5.3. Conjuntos de respuestas	759
4.5.4. Ejercicios y soluciones	769
5. Reflexiones finales	777
5.1. Resumen	777
5.2. Temas avanzados y aplicaciones	780
5.3. Recomendaciones bibliográficas	785
6. Aviso Legal	794

Prefacio

La lógica formal es uno de los fundamentos del pensamiento científico. Ofrece un lenguaje preciso para analizar proposiciones, conceptos, inferencias y demostraciones, y proporciona con ello herramientas que son de importancia central en la matemática, la informática, la filosofía, la lingüística y muchas otras disciplinas. Quien comprende las relaciones lógicas no solo adquiere una comprensión más profunda de los sistemas formales, sino también de la estructura de los argumentos y de los límites mismos del razonamiento.

Este libro desea abrir un acceso comprensible y, al mismo tiempo, sistemático a la lógica formal. Está dirigido a lectoras y lectores que quieran ocuparse de manera básica o más profunda de los procedimientos lógicos, y los acompaña desde los primeros conceptos de la lógica proposicional, pasando por la lógica de predicados, hasta las lógicas modales y algunas lógicas no clásicas seleccionadas. Se concede especial importancia a introducir los distintos temas no solo de forma rigurosa,

sino también mostrando las relaciones que existen entre ellos.

En el centro no se encuentra únicamente el aprendizaje de definiciones y teoremas, sino el desarrollo de una comprensión sólida de los métodos lógicos. Por ello, conceptos centrales como sintaxis, semántica, modelo, consecuencia, equivalencia y forma normal se construyen cuidadosamente y se complementan mediante ejemplos, ejercicios y soluciones. El recorrido conduce desde los fundamentos del razonamiento formal, pasando por procedimientos algorítmicos como la resolución y los SAT-solvers, hasta temas avanzados como la teoría de Herbrand, la lógica descriptiva, la lógica epistémica, la lógica difusa, la lógica por defecto y la programación de conjuntos de respuestas.

El libro sigue un enfoque didáctico que combina precisión formal con un desarrollo gradual. Muchas secciones están concebidas de tal modo que no solo presentan resultados individuales, sino que también permiten reconocer su motivación, su significado y su aplicación. De este modo, la lógica no debe aparecer como una colección de reglas aisladas, sino como un campo coherente, con una clara estructura interna y un amplio alcance práctico.

Mi agradecimiento va dirigido a todas las personas que han contribuido a la elaboración de este libro mediante sugerencias, comentarios e indicaciones críticas. Asimismo, agradezco a las lectoras y los lectores que se adentran en el viaje, a veces exigente pero siempre fructífero, por la lógica formal. Las observaciones, correcciones y propuestas de mejora son siempre bienvenidas, pues

ayudan a perfilar y mejorar aún más esta obra.

Les deseo una lectura enriquecedora y estimulante.

Lucien Sina

CAPÍTULO 0. PREFACIO

0.1. Introducción y visión general

La lógica es la disciplina que estudia sistemáticamente el razonamiento inferencial. En el centro se encuentran los lenguajes formales (sintaxis), que establecen las reglas para la formación de expresiones bien formadas, y las semánticas, que asignan significado a esa sintaxis. Con estas herramientas pueden formularse y analizarse con precisión cuestiones relativas a la verdad, la consecuencia y la representación del conocimiento. Una lógica proporciona así tanto una descripción de qué expresiones son válidas como un modelo de cómo deben interpretarse los enunciados respecto de estructuras, los llamados modelos.

0.1.1. Motivación y relevancia

El estudio de la lógica es central por varias razones. Por una parte, la lógica constituye el fundamento formal de la matemática y de muchos ámbitos de la informática. Conceptos como algoritmo, computabilidad y complejidad están históricamente estrechamente vinculados con la lógica formal; su formación conceptual y sus técnicas de demostración son fundamentales para comprender qué puede resolverse mediante procedimientos automáticos. Por otra parte, la lógica posee aplicaciones directas: especificación y verificación de software y hardware, representación del conocimiento en la inteligencia artificial, consultas en bases de datos, razonamiento automático en demostradores de teoremas, así como construcción de modelos en las formalizaciones de numerosas ciencias.

0.1.2. Contexto histórico

La lógica formal moderna tiene sus raíces en la filosofía y en la matemática de los siglos XIX y XX. Entre sus hitos destacan Frege, Russell e Hilbert, quienes impulsaron los sistemas formales y las presentaciones axiomáticas, así como Turing, Church y Gödel, cuyos resultados aclararon la relación entre computabilidad y demostrabilidad formal. De estos desarrollos surgieron conceptos centrales que aún hoy constituyen el marco para el tratamiento técnico y teórico de la lógica.

0.1.3. Objetivos de este libro

Este libro persigue tres objetivos que se complementan mutuamente:

1. Ofrecer una introducción precisa y autocontenida a las lógicas más importantes (lógica proposicional, lógica de predicados, lógicas modales, lógicas no clásicas seleccionadas y programación de conjuntos de respuestas).
2. Establecer un puente con la informática: se tratan aspectos algorítmicos, cuestiones de decidibilidad y complejidad, así como procedimientos concretos (por ejemplo, SAT-solvers, model checking y programación de conjuntos de respuestas).
3. Transmitir habilidades metodológicas: formular correctamente, demostrar por inducción (completa

y estructural), así como manejar con seguridad la sintaxis y la semántica.

0.1.4. Procedimiento y estructura

En este libro seguimos el siguiente patrón recurrente:

- **Sintaxis:** Definimos el lenguaje formal mediante signaturas, términos y fórmulas; además, presentamos reglas de formación y convenciones de notación.
- **Semántica:** Describimos cómo se interpretan las fórmulas en estructuras y definimos nociones de verdad, satisfacibilidad y validez.
- **Procedimientos de prueba y decisión:** Para cada lógica consideramos sistemas de prueba habituales (axiomáticos, deducción natural, resolución o similares), así como procedimientos algorítmicos y sus límites.
- **Ejemplos y aplicaciones:** Al final de cada capítulo central tratamos aplicaciones típicas y proponemos ejercicios.

0.1.5. ¿A quién está dirigido este libro?

El libro está dirigido a estudiantes avanzados de grado y máster de informática, matemáticas y disciplinas afines, así como a profesionales que buscan una comprensión sólida y matemáticamente fundamentada de la lógica. Se presuponen conocimientos básicos de teoría elemental

de conjuntos, matemática discreta y técnicas de demostración (en particular, inducción completa); cuando sea necesario, repasaremos brevemente estos fundamentos.

Lo que este libro no pretende ofrecer

Esta obra no es una colección exhaustiva de ejercicios de programación para herramientas lógicas específicas ni una exposición histórica completa de todas las escuelas de la lógica. Más bien, se concentra en los conceptos formales, los algoritmos típicos y la conexión con la práctica informática. Los aspectos especializados (por ejemplo, la implementación detallada de SMT-solvers o SAT-solvers) solo se tratan en la medida en que sirven para ilustrar afirmaciones teóricas.

0.1.6. Objetivos de aprendizaje e indicaciones de uso

Después de la lectura de los seis primeros capítulos, las lectoras y los lectores deberían estar en condiciones de:

- definir lenguajes formales y describir su semántica en estructuras;
- aplicar técnicas de demostración típicas de la lógica (inducción completa y estructural, demostración por contradicción, construcción directa);
- comprender en sus rasgos fundamentales procedimientos algorítmicos como la resolución SAT y el

model checking, así como llevar a cabo manualmente casos sencillos;

- reconocer los límites de los sistemas formales (indecidibilidad, clasificación de complejidad) y extraer de ellos consecuencias para la informática.

Indicaciones sobre la notación

Nos esforzamos por mantener una notación coherente; las convenciones centrales son:

- $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ designan los conjuntos numéricos usuales;
- las variables proposicionales o de fórmula se denotan normalmente por p, q, r, φ, ψ , y los individuos por x, y, z ;
- para las operaciones con conjuntos usamos $\cup, \cap, \setminus, \times$;
- los cuantificadores aparecen en la forma $\forall x$ o bien $\exists x$.

Procedimiento posterior

A continuación ofrecemos una introducción breve, pero completa, a la notación y repasamos la teoría de conjuntos y la teoría de la inducción necesarias. Después comenzamos con la lógica proposicional y avanzamos paso a paso hacia los temas más especializados.

En lo que sigue encontrará breves resúmenes de cada capítulo, que ofrecen una orientación sobre los contenidos y las competencias que se transmiten en las secciones correspondientes.

0.1.7. Breves descripciones de los capítulos

En los siguientes apartados ofrecemos una orientación concisa sobre los temas que tratan los distintos capítulos y sobre las competencias que la lectora o el lector debería adquirir a partir de ellos.

Capítulo 0 – Introducción y fundamentos. Este capítulo establece los fundamentos notacionales y conceptuales centrales: teoría de conjuntos y de relaciones, funciones, principios de inducción, construcción de lenguajes formales, así como conceptos básicos de la semántica (modelos, satisfacibilidad, validez, consecuencia lógica). Sirve como referencia a la que se recurrirá repetidamente en los capítulos siguientes.

Capítulo 1 – Lógica proposicional. La lógica proposicional constituye la entrada al razonamiento formal: introducimos variables proposicionales y conectivos (“y”, “o”, “no”, “implica”), definimos la evaluación semántica mediante tablas de verdad y estudiamos formas normales sintácticas (FNC, FND). Entre los temas principales se encuentran los procedimientos de prueba (métodos basados en resolución, deducción natural), la satisfacibilidad

(SAT) y las afirmaciones básicas sobre complejidad.

Capítulo 2 – Lógica de predicados. La lógica de predicados amplía la lógica proposicional mediante cuantificadores y predicados, de modo que puedan expresarse relaciones entre individuos. Tratamos signaturas, estructuras, interpretación de términos y fórmulas, así como resultados metateóricos centrales (completitud, incompletitud, Löwenheim–Skolem, cuestiones de decidibilidad), y mostramos sus consecuencias para la informática (por ejemplo, los límites de la automatización).

Capítulo 3 – Lógicas modales (incl. lógica temporal). Las lógicas modales permiten cualificar enunciados mediante operadores como “posible”, “necesario”, “siempre” o “en algún momento”. Introducimos la semántica de Kripke, estudiamos sistemas modales típicos (K, T, S4, S5) y describimos instancias especiales como las lógicas temporales (LTL, CTL) y sus aplicaciones en el model checking y en la especificación de sistemas.

Capítulo 4 – Otras lógicas no clásicas y programación de conjuntos de respuestas. Aquí consideramos una selección de lógicas no clásicas que representan extensiones o alternativas típicas a la lógica clásica: lógica intuicionista, lógica paraconsistente, lógicas multivaluadas y lógica difusa, lógicas por defecto y probabilísticas. Además, presentamos la programación de conjuntos de respuestas (Answer Set Programming, ASP) como una

técnica práctica de representación del conocimiento e inferencia no monótona.

Capítulo 5 – Resumen y perspectivas. El capítulo final resume resultados centrales, muestra conexiones entre los campos tratados y ofrece una breve perspectiva sobre el razonamiento automático, temas actuales de investigación y aplicaciones avanzadas en informática, inteligencia artificial y matemática.

Conceptos importantes que reaparecen a lo largo de este libro son:

- **Fórmula** – una expresión sintácticamente bien formada de un lenguaje lógico.
- **Estructura o modelo** – una interpretación que asigna valores a los símbolos del lenguaje.
- **Satisfacibilidad** – existencia de un modelo que hace verdadera una fórmula.
- **Consecuencia** – relación en la que una fórmula se sigue lógicamente de un conjunto de fórmulas.
- **Model checking** – comprobación algorítmica de si una estructura es modelo de una fórmula.

Este libro persigue el objetivo de ofrecer una introducción precisa y orientada a las aplicaciones a las lógicas centrales. Concedemos importancia a la precisión formal, a ejemplos motivadores y a aspectos algorítmicos como la computabilidad y la complejidad.

0.2. Fundamentos

0.2.1. Notación y teoría de conjuntos

Utilizamos la notación habitual de la teoría de conjuntos. Si X es un conjunto y x un objeto, entonces $x \in X$ significa que x es un elemento de X . Para dos conjuntos X e Y escribimos $X \subseteq Y$ cuando X es un subconjunto de Y .

Los conjuntos de los números naturales, enteros y reales se denotan por $\mathbb{N}$, $\mathbb{Z}$ y $\mathbb{R}$, respectivamente.

El producto cartesiano de dos conjuntos X e Y se denota por $X \times Y$. La potencia de un conjunto X la escribimos como $\mathcal{P}(X)$ o 2^X .

Conjuntos, operaciones y relaciones

Los conjuntos pueden representarse de manera implícita mediante condiciones de construcción. Por ejemplo,

$$D = \left\{ x \in \mathbb{N} \left| \begin{array}{l} \exists y \in \mathbb{N} : y \mid x \\ \text{y } 1 < y < x \end{array} \right. \right\}$$

es el conjunto de todos los números naturales compuestos.

Operaciones importantes con conjuntos son la unión $\cup$, la intersección $\cap$ y la diferencia $\setminus$. Para dos conjuntos X, X' se tiene:

$$X \cup X' = \{ x \mid x \in X \text{ o } x \in X' \},$$

$$X \cap X' = \{ x \mid x \in X \text{ y } x \in X' \},$$

$$X \setminus X' = \{ x \mid x \in X \text{ y } x \notin X' \}.$$

El producto cartesiano de X y X' es

$$X \times X' = \{ (x, x') \mid x \in X, x' \in X' \}.$$

De manera más general, escribimos $X^n = X \times \cdots \times X$ (n veces) para el producto cartesiano n -ario.

Una relación (binaria) R entre los conjuntos X y X' es un subconjunto $R \subseteq X \times X'$. En lugar de $(x, x') \in R$, escribimos a veces también xRx' . Una relación n -aria sobre X es un subconjunto de X^n .

Si $R \subseteq X \times X$ (es decir, si $X' = X$), definimos:

- R es *reflexiva* si y solo si $(x, x) \in R$ para todo $x \in X$.
- R es *simétrica* si y solo si de $(x, y) \in R$ se sigue también $(y, x) \in R$.
- R es *transitiva* si y solo si de $(x, y) \in R$ y $(y, z) \in R$ se sigue también $(x, z) \in R$.

Si R es reflexiva, simétrica y transitiva, entonces R se llama una *relación de equivalencia*.

Una función f es una aplicación $f : X \rightarrow Y$. Para $x \in X$, denotamos por $f(x)$ la imagen de x . Toda función puede entenderse también como una relación especial $f \subseteq X \times Y$ que satisface la propiedad de funcionalidad: de $(x, y) \in f$ se sigue que no existe ningún $y' \in Y$ con $y' \neq y$ y $(x, y') \in f$.

Sea $f : X \rightarrow Y$.

- f es *inyectiva* (una aplicación uno a uno) si y solo si, para todos $x, x' \in X$, de $f(x) = f(x')$ se sigue siempre $x = x'$.
- f es *sobreyectiva* si y solo si, para todo $y \in Y$, existe un $x \in X$ con $f(x) = y$.
- f es *biyectiva* si y solo si es inyectiva y sobreyectiva.

La imagen inversa f^{-1} se define en general como una aplicación en el conjunto potencia:

$$f^{-1} : Y \rightarrow \mathcal{P}(X), \quad f^{-1}(y) = \{x \in X \mid f(x) = y\}.$$

Si f es una biyección, entonces f^{-1} es una aplicación $f^{-1} : Y \rightarrow X$ y, para cada $y \in Y$, $f^{-1}(y)$ es el único $x \in X$ tal que $f(x) = y$.

Si $f : X \rightarrow Y$ y $g : Y \rightarrow Z$ son aplicaciones, entonces $g \circ f : X \rightarrow Z$ denota la composición, definida por $(g \circ f)(x) = g(f(x))$ para todo $x \in X$.

Notación de crecimiento. Para funciones $f, g : \mathbb{N} \rightarrow \mathbb{N}$ se escribe $g \in O(f)$ si y solo si existen constantes $c > 0$ y $n_0 \in \mathbb{N}$ tales que

$$\forall n \geq n_0 : \quad g(n) \leq c \cdot f(n).$$

Intuitivamente, $g \in O(f)$ significa que g no crece asintóticamente más rápido que f .

0.2.2. Estructuras y semántica

Una estructura $\mathcal{A}$ para una signatura lógica consta de un conjunto base A y de una interpretación de los símbolos de la signatura. Una fórmula φ es verdadera en una estructura $\mathcal{A}$ si se cumple $\mathcal{A} \models \varphi$.

Conceptos importantes son:

- Satisfacibilidad: existe una estructura $\mathcal{A}$ tal que $\mathcal{A} \models \varphi$
- Validez: para todas las estructuras $\mathcal{A}$ se cumple $\mathcal{A} \models \varphi$
- Consecuencia lógica: $\Phi \models \psi$

0.2.3. Principios de inducción

Las demostraciones por inducción son técnicas de prueba centrales en la lógica y en la informática teórica. Aquí distinguimos dos tipos habituales: la *inducción completa* sobre los números naturales y la *inducción estructural* sobre objetos definidos sintácticamente (términos, fórmulas, árboles). Ambos procedimientos se basan en el principio de las estructuras bien ordenadas o bien fundadas, pero difieren en la formulación del paso inductivo.

Inducción completa

La inducción completa (también: inducción simple o fuerte, según la formulación) se aplica a enunciados sobre

números naturales. Para demostrar un enunciado $A(n)$ para todo $n \in \mathbb{N}$, se muestra:

1. (*Base de la inducción*) $A(1)$ es verdadero (o $A(0)$, según la convención adoptada).
2. (*Paso inductivo*) Para un $n > 1$ arbitrario, de la hipótesis de que $A(k)$ vale para todo $k < n$ se sigue que también vale $A(n)$.

La inducción completa se justifica entonces por el hecho de que no puede existir un conjunto de contraejemplos con un elemento mínimo.

Ejemplo 0.1. (Fórmula de suma de Gauss) Para todo $n \in \mathbb{N}$ se cumple

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}.$$

Demostración. Base de la inducción: Para $n = 1$ se tiene $\sum_{i=1}^1 i = 1$ y $\frac{1 \cdot (1+1)}{2} = 1$. Por tanto, la igualdad es correcta para $n = 1$.

Paso inductivo: Sea $n > 1$ y supóngase que la fórmula vale para $n - 1$, es decir,

$$\sum_{i=1}^{n-1} i = \frac{(n-1)n}{2}.$$

Entonces, para n , se obtiene:

$$\begin{aligned}
 \sum_{i=1}^n i &= \left(\sum_{i=1}^{n-1} i \right) + n \\
 &= \frac{(n-1)n}{2} + n \\
 &= \frac{n(n-1) + 2n}{2} \\
 &= \frac{n(n+1)}{2}.
 \end{aligned}$$

Con ello queda demostrado el paso inductivo y, por tanto, el enunciado para todo n . $\square$

Inducción estructural

La inducción estructural es adecuada para enunciados sobre objetos definidos recursivamente y sobre objetos con forma de árbol, como términos o fórmulas. Se toma como guía la definición recursiva del tipo de objeto: se muestra la propiedad para los objetos elementales (atómicos) y luego que dicha propiedad se conserva al componerlos según las reglas de construcción.

Formalmente: sea $\mathcal{F}$ el conjunto de las fórmulas de un lenguaje, definido por las reglas

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi)$$

con fórmulas atómicas p . Para mostrar una propiedad $P(\varphi)$ para toda $\varphi \in \mathcal{F}$, se demuestra:

- $P(p)$ para cada p atómica.
- De $P(\varphi)$ se sigue $P(\neg\varphi)$.
- De $P(\varphi)$ y $P(\psi)$ se sigue $P((\varphi \wedge \psi))$ (y análogamente para otros operadores binarios).

Ejemplo 0.2. Toda fórmula $\varphi \in \mathcal{F}$ tiene solo un número finito de subfórmulas.

Demostración. Demostramos la afirmación “ φ tiene finitamente muchas subfórmulas” para toda $\varphi \in \mathcal{F}$ mediante inducción estructural.

Base: Sea $\varphi = p$ atómica. Entonces el conjunto de sus subfórmulas es $\{ p \}$, y por tanto es finito.

Paso inductivo: Supongamos que, para φ y ψ , el conjunto de sus subfórmulas es finito en cada caso. Entonces las subfórmulas de $\neg\varphi$ son $\{\neg\varphi\} \cup \{\text{subfórmulas de } \varphi\}$, es decir, una unión finita de conjuntos finitos y, por tanto, un conjunto finito. Para la fórmula compuesta $(\varphi \wedge \psi)$ se tiene de modo análogo que el conjunto de sus subfórmulas

$$\{ (\varphi \wedge \psi) \} \cup \{\text{subfórmulas de } \varphi\} \cup \{\text{subfórmulas de } \psi\}$$

es una unión finita de conjuntos finitos y, por consiguiente, también es finito.

Así, por inducción estructural, se sigue que toda $\varphi \in \mathcal{F}$ posee solo finitamente muchas subfórmulas. $\square$

Comparación, indicaciones y casos típicos de aplicación

- **¿Cuándo usar cada tipo de inducción?** Usa la inducción completa cuando la afirmación esté formulada sobre números naturales (o directamente sobre $\mathbb{N}$); por ejemplo, en combinatoria, sumas o sucesiones recursivas. Usa la inducción estructural cuando las afirmaciones traten sobre objetos definidos recursivamente y con forma de árbol, como términos, fórmulas, árboles de prueba o árboles de sintaxis abstracta.
- **Forma del paso inductivo.** En la inducción completa, el paso inductivo suele referirse al predecesor inmediato $n - 1$ o a todos los $k < n$ (inducción fuerte). En la inducción estructural se muestra que la propiedad se conserva bajo las reglas de construcción.
- **Puntos en común.** Ambos métodos son aplicaciones del principio según el cual no existen sucesiones infinitas estrictamente descendentes en una relación bien fundada. Formalmente, ambos métodos pueden entenderse como casos particulares de la inducción bien fundada (well-founded induction).
- **Consejos prácticos.** Formula las afirmaciones inductivas de modo que la hipótesis de inducción contenga exactamente la información necesaria. En la inducción estructural suele ser recomendable

elegir afirmaciones inductivas fuertes (por ejemplo, afirmaciones sobre el conjunto de todas las subestructuras), para que las demostraciones en los casos compuestos resulten más sencillas.

0.3. Estructura y estilo de los textos matemáticos

Los textos matemáticos de este libro siguen una forma clara y reconocible: las definiciones y las notaciones fijan conceptos y convenciones precisos, los ejemplos ilustran la intuición, los resultados (lemas, proposiciones, teoremas, corolarios) formulan las afirmaciones centrales, y las pruebas justifican dichas afirmaciones de manera rigurosa. Esta estructura ayuda a la lectora o al lector a distinguir rápidamente entre enunciados formales, demostraciones y material ilustrativo.

Definiciones y notaciones

- **Definiciones.** Las definiciones formalizan nuevos conceptos. Deben ser concisas, completas y estar formuladas de tal modo que las propiedades necesarias puedan comprobarse directamente.
- **Notaciones.** Las notaciones reúnen abreviaturas y convenciones de escritura (por ejemplo, $\mathcal{P}(X)$, $X \times Y$, $O(\cdot)$). Deben usarse de forma uniforme y, cuando sea posible, resumirse en un lugar visible,

por ejemplo al comienzo del capítulo o en un recuadro de notación.

Ejemplos e ilustraciones

Los ejemplos sirven para orientar la intuición y muestran casos típicos o contrarios. Deben ser breves, demostrar el uso de nuevas definiciones y, cuando sea posible, remitir a ejercicios o aplicaciones posteriores.

Resultados: lema, proposición, teorema, corolario

- **Lema.** Un resultado auxiliar que se utiliza principalmente en la demostración de otro resultado.
- **Proposición.** Un resultado útil, por lo general menos central.
- **Teorema.** Un resultado central e importante.
- **Corolario.** Una consecuencia inmediata de un resultado anterior.

Todos los resultados deben contener una formulación precisa, las hipótesis cuando sean necesarias y una demostración clara.

Demostraciones

Las demostraciones forman parte de la argumentación matemática habitual. Deben estar estructuradas del modo

más claro posible:

1. Breve indicación inicial de lo que se va a demostrar.
2. Si es necesario: división en casos o en lemas auxiliares.
3. Al final, un signo claro de cierre, usualmente escrito como $\square$.

Observaciones, ejercicios y referencias bibliográficas

- **Observaciones** contienen a menudo información adicional, intuición o generalizaciones. Son más breves y menos formales.
- **Ejercicios** al final del capítulo consolidan la comprensión.
- **Referencias bibliográficas** remiten a fuentes complementarias y deben citarse con precisión.

Breve ejemplo modelo

Definición 0.3. Una aplicación $f : X \rightarrow Y$ se llama *inyectiva* si de $f(x) = f(x')$ se sigue necesariamente $x = x'$.

Teorema 0.4. Sea $f : X \rightarrow Y$ *inyectiva* y sea $g : Y \rightarrow Z$ *arbitraria*. Entonces $g \circ f$ es *inyectiva* si y solo si g es *inyectiva* sobre la imagen de f .

Demostración. La afirmación se obtiene comprobando directamente la definición: si $g \circ f$ es inyectiva y $g(f(x)) = g(f(x'))$, entonces $f(x) = f(x')$, y por la inyectividad de f se sigue $x = x'$. La otra dirección es igualmente directa. $\square$

Capítulo 1

Lógica proposicional

1.1. Introducción y visión general

La lógica proposicional es el sistema formal básico con el que pueden describirse con precisión enunciados simples y sus conexiones lógicas. En el centro se encuentran las proposiciones atómicas, cada una de las cuales puede asumir exactamente uno de dos valores de verdad: verdadero o falso. A partir de estos componentes atómicos se forman, mediante conectivos como *y*, *o* y *no*, fórmulas compuestas que permiten expresar también relaciones lógicas más complejas.

A pesar de su sintaxis elemental, la lógica proposicional tiene un alcance considerable. Constituye una base importante para muchas otras áreas de la lógica y desempeña al mismo tiempo un papel central en la informática, por ejemplo en la demostración automática, en el análisis

CAPÍTULO 1. LÓGICA PROPOSICIONAL

de circuitos lógicos, en la verificación y en procedimientos de la inteligencia artificial. Resulta especialmente útil que la lógica proposicional esté completamente formalizada: tanto la formación de fórmulas como su significado pueden especificarse de manera exacta.

Este capítulo introduce primero los fundamentos sintácticos y semánticos. Comenzamos con signatures, fórmulas e interpretaciones, y aclaramos después cuándo una fórmula es verdadera bajo una interpretación y qué significa que un conjunto de fórmulas posea un modelo. Luego consideramos el concepto de consecuencia lógica, así como la cuestión de cuándo una fórmula se sigue de otra o de un conjunto de fórmulas. Sobre esta base estudiamos la equivalencia lógica de fórmulas y presentamos reglas importantes de transformación, junto con formas normales que son especialmente relevantes para el procesamiento algorítmico.

En el desarrollo posterior del capítulo nos ocupamos de procedimientos que no solo describen semánticamente el razonamiento lógico, sino que también lo ejecutan de manera sintáctica. Entre ellos se encuentran los cálculos generales y, en particular, el cálculo de resolución, que constituye una herramienta central de la demostración automática. Finalmente examinamos con más detalle el lado algorítmico de la lógica proposicional: estudiamos en especial el problema de satisfacibilidad, explicamos por qué es difícil desde el punto de vista teórico y presentamos, con la codificación de Tseitin y el algoritmo DPLL, dos métodos importantes con los que en la práctica puede

comprobarse a menudo la satisfacibilidad de forma muy eficiente.

El capítulo está organizado del siguiente modo: en el apartado 1.1 introducimos la sintaxis y la semántica de la lógica proposicional. Allí tratamos primero las signaturas, después las fórmulas, así como las interpretaciones y los modelos. En el apartado 1.2 se estudian las propiedades de los modelos, el concepto de consecuencia y la independencia de la signatura. En el apartado 1.3 abordamos el tema de la equivalencia; esto incluye tanto la definición de equivalencia lógica como transformaciones equivalentes importantes y la representación de enunciados en formas normales. En el apartado 1.4 se presentan distintos cálculos para la inferencia automática, en particular el cálculo de resolución. Por último, el apartado 1.5 explica la reducción a problemas de satisfacibilidad, la codificación de Tseitin y el algoritmo DPLL como métodos centrales para resolver el problema de satisfacibilidad.

De este modo, el capítulo vincula la base conceptual de la lógica proposicional con sus procedimientos algorítmicos centrales. Así establece la base para todo lo que se necesitará más adelante sobre razonamiento formal, formas normales y procedimientos automáticos de demostración.

1.2. Sintaxis y semántica

1.2.1. Introducción

Como toda lógica, la lógica proposicional se define mediante dos componentes:

- *Sintaxis*: la descripción de las expresiones bien formadas (fórmulas), es decir, qué cadenas de símbolos cuentan como objetos válidos, y
- *Semántica*: la asignación de significados (valores de verdad) a dichas expresiones.

En lo que sigue introduciremos primero las firmas y las fórmulas, y después definiremos la semántica modeloteórica.

1.2.2. Firmas

Definición 1.1. Una *firma proposicional* Σ es un conjunto (finito) de variables proposicionales (también llamadas átomos o proposiciones).

Ejemplo 1.2. Para modelar un sistema de alarma doméstico inteligente, se puede elegir, por ejemplo,

$$\Sigma_{\text{Alarma}} = \{\text{Humo}, \text{Calor}, \text{Movimiento}, \text{Alarma}, \text{Rociadores}\},$$

donde los significados previstos de las variables son, por ejemplo:

- Humo: se detecta humo

- Calor: se detecta una temperatura elevada
- Movimiento: se registra movimiento en la casa
- Alarma: la alarma está activada
- Rociadores: los rociadores están activados

Una fórmula típica en $L(\Sigma_{\text{Alarma}})$ sería, por ejemplo,

$$(\text{Humo} \vee \text{Calor}) \rightarrow \text{Alarma},$$

que expresa: «Si se detecta humo o una temperatura elevada, entonces se activa la alarma». ■

1.2.3. Fórmulas

Definición 1.3. Sea Σ una signatura proposicional. El conjunto $L(\Sigma)$ de las *fórmulas* sobre Σ se define inductivamente del siguiente modo:

1. toda variable proposicional $p \in \Sigma$ es una fórmula (fórmula atómica);
2. si φ es una fórmula, entonces $\neg\varphi$ es una fórmula;
3. si φ y ψ son fórmulas, entonces $(\varphi \wedge \psi)$ y $(\varphi \vee \psi)$ son fórmulas.

Alternativamente (BNF):

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \quad (p \in \Sigma).$$

Notación 1.4. Usamos las siguientes abreviaturas y convenciones:

- Conectores: $\neg$ (negación), $\wedge$ (conjunción), $\vee$ (disyunción).
- La implicación y la equivalencia se definen como $\varphi \rightarrow \psi := \neg\varphi \vee \psi$ y $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.
- Convención de precedencia: $\neg$ tiene mayor prioridad que $\wedge$, este tiene mayor prioridad que $\vee$, este mayor prioridad que $\rightarrow$, y este mayor prioridad que $\leftrightarrow$.
- Un *literal* es un átomo p o su negación $\neg p$.

Ejemplo 1.5. Con Σ_{Alarm} del ejemplo anterior, las siguientes son, por ejemplo, fórmulas de $L(\Sigma_{\text{Alarm}})$:

$$\begin{aligned} &(\text{Smoke} \vee \text{Heat}) \rightarrow \text{Alarm}, \\ &\neg \text{Smoke} \wedge (\text{Alarm} \vee \text{Sprinkler}), \\ &\text{Motion} \rightarrow \text{Alarm}. \end{aligned}$$

El conjunto de literales es

$$\begin{aligned} \text{Lit}(\Sigma_{\text{Alarm}}) = \{ &\text{Smoke}, \neg \text{Smoke}, \text{Heat}, \neg \text{Heat}, \\ &\text{Motion}, \neg \text{Motion}, \text{Alarm}, \neg \text{Alarm}, \\ &\text{Sprinkler}, \neg \text{Sprinkler} \}. \end{aligned}$$

■

1.2.4. Semántica: interpretaciones, satisfacción y modelos

Definición 1.6. Sea Σ una signatura. Una *interpretación* (o asignación) ω sobre Σ es una función $\omega : \Sigma \rightarrow$

$\{\text{true}, \text{false}\}$. Llamamos $\Omega(\Sigma)$ al conjunto de todas las interpretaciones.

Definición 1.7. La relación de satisfacción $\models_{\text{LP}} \subseteq \Omega(\Sigma) \times L(\Sigma)$ se define inductivamente:

$$\begin{aligned} \omega \models_{\text{LP}} p & \text{ ssi } \omega(p) = \text{true}, \\ \omega \models_{\text{LP}} \neg \varphi & \text{ ssi } \text{no } (\omega \models_{\text{LP}} \varphi), \\ \omega \models_{\text{LP}} \varphi \wedge \psi & \text{ ssi } \omega \models_{\text{LP}} \varphi \text{ y } \omega \models_{\text{LP}} \psi, \\ \omega \models_{\text{LP}} \varphi \vee \psi & \text{ ssi } \omega \models_{\text{LP}} \varphi \text{ o } \omega \models_{\text{LP}} \psi. \end{aligned}$$

Si $\omega \models_{\text{LP}} \varphi$, entonces ω se llama un *modelo* de φ .

Definición 1.8. Para una fórmula φ , definimos

$$\text{Mod}_{\text{LP}}(\varphi) = \{ \omega \in \Omega(\Sigma) \mid \omega \models_{\text{LP}} \varphi \},$$

la colección de todos los modelos de φ . Además, φ es *satisfacible* ssi $\text{Mod}_{\text{LP}}(\varphi) \neq \emptyset$, y φ es *válida* (una tautología) ssi $\text{Mod}_{\text{LP}}(\varphi) = \Omega(\Sigma)$.

Explicación. Si φ es una fórmula y ω es un modelo de φ ($\omega \in \text{Mod}_{\text{LP}}(\varphi)$), entonces ω puede entenderse intuitivamente como un *mundo posible* o como una *situación* en la que el enunciado φ resulta efectivamente verdadero. La interpretación ω determina, para cada enunciado atómico $p \in \Sigma$, si p es verdadero o falso en ese mundo; de este modo, ω explica *completamente* por qué φ es verdadera: por la asignación de valores de verdad a los átomos a partir de los cuales está construida φ .

La denominación *modelo* subraya precisamente este carácter explicativo: un modelo es un testimonio concreto que prueba que una fórmula es satisfacible. En consecuencia, el conjunto $\Omega(\Sigma)$ es el conjunto de todos los mundos posibles (todas las valuaciones posibles), y $\text{Mod}_{\text{LP}}(\varphi)$ está formado por aquellos mundos que hacen verdadera a φ .

Dos conceptos breves y útiles:

- *Satisfacibilidad*: La fórmula φ es satisfacible exactamente cuando $\text{Mod}_{\text{LP}}(\varphi) \neq \emptyset$. Un $\omega \in \text{Mod}_{\text{LP}}(\varphi)$ concreto se denomina también *modelo* o *asignación satisfactoria* de φ .
- *Contramodelo*: Para mostrar que una fórmula φ *no* es válida (no es tautológica), basta indicar un $\omega \in \Omega(\Sigma)$ tal que $\omega \notin \text{Mod}_{\text{LP}}(\varphi)$; tal ω se denomina *contramodelo*.

Observación. En aplicaciones (por ejemplo, en model checking), encontrar un modelo corresponde a menudo a demostrar que un comportamiento deseado del sistema puede producirse, mientras que proporcionar un contramodelo muestra que una propiedad deseada se viola bajo determinadas valuaciones.

Ejemplo 1.9. Consideremos Σ_{Alarma} y la interpretación ω dada por $\omega(\text{Humo}) = \text{verdadero}$, $\omega(\text{Calor}) = \text{falso}$, $\omega(\text{Movimiento}) = \text{verdadero}$, $\omega(\text{Alarma}) = \text{verdadero}$, $\omega(\text{Rociadores}) = \text{falso}$. Entonces se cumple, por ejemplo, $\omega \models_{\text{LP}} \text{Humo}$, $\omega \models_{\text{LP}} \neg \text{Calor}$, $\omega \models_{\text{LP}} (\text{Humo} \vee \text{Calor}) \rightarrow$

Alarma (porque $\omega(\text{Humo}) = \text{verdadero}$ y $\omega(\text{Alarma}) = \text{verdadero}$), $\omega \models_{\text{LP}} \text{Movimiento} \rightarrow \text{Alarma}$ (ya que $\omega(\text{Movimiento}) = \text{verdadero}$ y $\omega(\text{Alarma}) = \text{verdadero}$) y $\omega \models_{\text{LP}} \neg \text{Rociadores}$. ■

1.2.5. Tablas de verdad

Los conectivos pueden describirse mediante tablas de verdad; presentamos aquí la forma compacta (1 = verdadero, 0 = falso):

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\neg \varphi$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

1.2.6. Observaciones

- Si $|\Sigma| = n$, entonces $|\Omega(\Sigma)| = 2^n$ interpretaciones.
- La inducción definitoria de la semántica (satisfacción) es equivalente a la especificación directa de una función de verdad $\llbracket \cdot \rrbracket_\omega$.
- La semántica modeloteórica constituye la base de conceptos como la *consecuencia lógica* $\Phi \models \psi$ (toda interpretación que satisface todas las fórmulas de Φ satisface también ψ).

1.2.7. Funciones de verdad

Definición 1.10. Sea Σ una signatura de la lógica proposicional y sea $\omega \in \Omega(\Sigma)$ una interpretación. La *función de verdad* $\llbracket \cdot \rrbracket_\omega : L(\Sigma) \rightarrow \{\text{true}, \text{false}\}$ se define inductivamente del siguiente modo:

$$\llbracket p \rrbracket_\omega = \omega(p) \quad (p \in \Sigma),$$

$$\llbracket \neg \varphi \rrbracket_\omega = \begin{cases} \text{true}, & \text{si } \llbracket \varphi \rrbracket_\omega = \text{false}, \\ \text{false}, & \text{en otro caso}, \end{cases}$$

$$\llbracket \varphi \wedge \psi \rrbracket_\omega = \begin{cases} \text{true}, & \text{si } \llbracket \varphi \rrbracket_\omega = \text{true} \text{ y } \llbracket \psi \rrbracket_\omega = \text{true}, \\ \text{false}, & \text{en otro caso}, \end{cases}$$

$$\llbracket \varphi \vee \psi \rrbracket_\omega = \begin{cases} \text{false}, & \text{si } \llbracket \varphi \rrbracket_\omega = \text{false} \text{ y } \llbracket \psi \rrbracket_\omega = \text{false}, \\ \text{true}, & \text{en otro caso}. \end{cases}$$

Proposición 1.11. Sea Σ una signatura, sea $\omega \in \Omega(\Sigma)$ una interpretación y sea $\varphi \in L(\Sigma)$. Entonces se cumple

$$\omega \models_{\text{LP}} \varphi \quad \text{ssi} \quad \llbracket \varphi \rrbracket_\omega = \text{true}.$$

Demostración. Sea Σ una signatura proposicional, sea $\omega \in \Omega(\Sigma)$ una interpretación y sea $\varphi \in L(\Sigma)$ una fórmula. Demostramos la equivalencia

$$\omega \models_{\text{LP}} \varphi \quad \text{ssi} \quad \llbracket \varphi \rrbracket_\omega = \text{true}$$

por inducción estructural sobre la fórmula φ .

Caso base. Sea $\varphi = p$, con $p \in \Sigma$, un átomo. Por definición de la relación de satisfacción se cumple que

$\omega \models_{\text{LP}} p$ exactamente cuando $\omega(p) = \text{true}$. Por definición de la función de verdad se tiene $\llbracket p \rrbracket_{\omega} = \omega(p)$. De ello se sigue inmediatamente $\omega \models_{\text{LP}} p \iff \llbracket p \rrbracket_{\omega} = \text{true}$.

Paso inductivo. Sea ahora φ una fórmula, y supongamos que la afirmación ya vale para todas las subfórmulas propias de φ (hipótesis de inducción). Consideramos los distintos casos de construcción.

- $\varphi = \neg\psi$.
 $\Rightarrow$: Supongamos que $\omega \models_{\text{LP}} \neg\psi$. Entonces, por definición, $\omega \not\models_{\text{LP}} \psi$. Por la hipótesis de inducción se sigue que $\llbracket \psi \rrbracket_{\omega} \neq \text{true}$, es decir, $\llbracket \psi \rrbracket_{\omega} = \text{false}$. Por definición de la función de verdad se obtiene entonces $\llbracket \neg\psi \rrbracket_{\omega} = \text{true}$.
 $\Leftarrow$: Supongamos que $\llbracket \neg\psi \rrbracket_{\omega} = \text{true}$. Entonces $\llbracket \psi \rrbracket_{\omega} = \text{false}$. Por la hipótesis de inducción se sigue que $\omega \not\models_{\text{LP}} \psi$, y por tanto $\omega \models_{\text{LP}} \neg\psi$.
- $\varphi = \psi \wedge \xi$.
 $\Rightarrow$: Supongamos que $\omega \models_{\text{LP}} \psi \wedge \xi$. Entonces se cumple que $\omega \models_{\text{LP}} \psi$ y $\omega \models_{\text{LP}} \xi$. Por la hipótesis de inducción se tiene, por tanto, $\llbracket \psi \rrbracket_{\omega} = \text{true}$ y $\llbracket \xi \rrbracket_{\omega} = \text{true}$. Por consiguiente, $\llbracket \psi \wedge \xi \rrbracket_{\omega} = \text{true}$.
 $\Leftarrow$: Supongamos que $\llbracket \psi \wedge \xi \rrbracket_{\omega} = \text{true}$. Entonces $\llbracket \psi \rrbracket_{\omega} = \text{true}$ y $\llbracket \xi \rrbracket_{\omega} = \text{true}$. Por la hipótesis de inducción se sigue que $\omega \models_{\text{LP}} \psi$ y $\omega \models_{\text{LP}} \xi$, luego $\omega \models_{\text{LP}} \psi \wedge \xi$.
- $\varphi = \psi \vee \xi$.
 $\Rightarrow$: Supongamos que $\omega \models_{\text{LP}} \psi \vee \xi$. Entonces se

cumple que $\omega \models_{\text{LP}} \psi$ o $\omega \models_{\text{LP}} \xi$. Sin pérdida de generalidad, sea $\omega \models_{\text{LP}} \psi$. Por la hipótesis de inducción, $\llbracket \psi \rrbracket_{\omega} = \text{true}$, de modo que $\llbracket \psi \vee \xi \rrbracket_{\omega} = \text{true}$. (El otro caso es análogo.)

$\Leftarrow$: Supongamos que $\llbracket \psi \vee \xi \rrbracket_{\omega} = \text{true}$. Entonces $\llbracket \psi \rrbracket_{\omega} = \text{true}$ o $\llbracket \xi \rrbracket_{\omega} = \text{true}$. Por la hipótesis de inducción se sigue que $\omega \models_{\text{LP}} \psi$ o $\omega \models_{\text{LP}} \xi$, y por tanto $\omega \models_{\text{LP}} \psi \vee \xi$.

Como todos los casos de construcción han sido cubiertos, la afirmación se sigue para toda fórmula φ . $\square$

Los conectivos también pueden indicarse de forma compacta mediante tablas de verdad. Primero, la variante con las etiquetas true/false:

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\neg \varphi$
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

Y aquí la representación compacta 0/1 (0 = falso, 1 = verdadero):

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\neg \varphi$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Observación. La definición anterior de la función de verdad es equivalente a la definición inductiva de la relación de satisfacción $\models_{LP}$. En las demostraciones, una u otra formulación puede resultar más práctica según el contexto: la relación de satisfacción destaca la perspectiva de los modelos ($\omega \models \varphi$), mientras que la función de verdad indica explícitamente el valor de la fórmula en una interpretación dada ($\llbracket \varphi \rrbracket_\omega \in \{\text{verdadero, falso}\}$).

Perspectiva

En la siguiente sección trataremos las formas normales sintácticas, las reglas concretas de transformación (por ejemplo, FNC, FND) y los primeros procedimientos de prueba (resolución, deducción natural), que son relevantes tanto desde el punto de vista teórico como en la práctica (por ejemplo, para los SAT-solvers).

1.3. Ejercicios con soluciones

Ejercicio 1. Sea la signatura proposicional $\Sigma = \{P, Q, R, S, T, U\}$. Sean dados los enunciados

$$\varphi_1 = S \wedge \neg(P \leftrightarrow \neg R),$$

$$\begin{aligned} \omega_2 : P \mapsto \text{true}, \quad Q \mapsto \text{false}, \quad R \mapsto \text{true}, \\ S \mapsto \text{false}, \quad T \mapsto \text{true}, \quad U \mapsto \text{false}. \end{aligned}$$

y

$$\varphi_3 = \neg T \wedge \neg(\neg T \vee R).$$

CAPÍTULO 1. LÓGICA PROPOSICIONAL

Decida si las siguientes afirmaciones son **verdaderas** o **falsas** (justifique brevemente cada respuesta):

1. $\neg P \vee R \in L(\Sigma)$.
2. La fórmula indicada

$$\overline{\varphi_1} = \neg S \vee (\neg P \vee \neg R) \wedge (R \vee P) \vee \neg Q$$

es el complemento (la negación) de φ_1 .

3. $\llbracket \neg T \vee (\neg R \leftrightarrow P) \rrbracket_{\omega_2} = \text{false}$.
4. $|\text{Mod}_{\text{LP}}(\varphi_3)| = 0$.

Soluciones y explicaciones.

1. **Verdadero.** $\neg P \vee R$ es una expresión bien formada, construida a partir de átomos de Σ mediante los conectores $\neg$ y $\vee$; por tanto, es una fórmula de $L(\Sigma)$.
2. **Falso.** La fórmula afirmada no es la negación de φ_1 . En primer lugar, contiene el átomo Q , que no aparece en φ_1 ; en segundo lugar, por las leyes de De Morgan, la estructura sintáctica de una negación de $\varphi_1 = S \wedge \neg(P \leftrightarrow \neg R)$ es equivalente a

$$\neg S \vee \neg \neg(P \leftrightarrow \neg R) = \neg S \vee (P \leftrightarrow \neg R),$$

no a la fórmula complicada indicada, que contiene $\neg Q$ y el término adicional $(\neg P \vee \neg R) \wedge (R \vee P)$.

3. **Verdadero.** Calculamos la fórmula $\neg T \vee (\neg R \leftrightarrow P)$ bajo ω_2 . Bajo ω_2 se tiene $T = \text{true}$, luego $\neg T = \text{false}$. Además, $R = \text{true}$, por lo que $\neg R = \text{false}$. Como $P = \text{true}$, $\neg R \leftrightarrow P$ es la equivalencia proposicional entre false y true , es decir, false . Por tanto, $\neg T \vee (\neg R \leftrightarrow P)$ es igual a $\text{false} \vee \text{false} = \text{false}$. Así, la afirmación es correcta.

4. **Verdadero.** Simplificamos $\varphi_3 = \neg T \wedge \neg(\neg T \vee R)$. La subfórmula $\neg(\neg T \vee R)$ es equivalente a $\neg\neg T \wedge \neg R$, es decir, a $T \wedge \neg R$. Por tanto, $\varphi_3 \equiv \neg T \wedge (T \wedge \neg R)$. Esta fórmula es claramente insatisfacible, pues exigiría simultáneamente $\neg T$ y T . Por ello, φ_3 no tiene *ningún* modelo y, en consecuencia, $|\text{Mod}_{\text{LP}}(\varphi_3)| = 0$.

Retroalimentación breve:

- (1) La fórmula pertenece al lenguaje sobre Σ : correcto.
- (2) La fórmula de negación indicada es falsa: contiene un átomo ajeno y no tiene la estructura adecuada (no es equivalente a la negación de φ_1).
- (3) Verdadero: la evaluación da como resultado false .
- (4) Verdadero: φ_3 es contradictoria y, por tanto, insatisfacible.

Ejercicio avanzado. Varíe la signatura Σ (por ejemplo, usando solo tres átomos) y determine, para todas las $2^{|\Sigma|}$ interpretaciones, qué fórmulas del Ejercicio 1 son verdaderas. Compare el número de modelos de las fórmulas correspondientes.

Solución modelo (ejemplo con $\Sigma' = \{P, Q, R\}$)

Elegimos $\Sigma' = \{P, Q, R\}$ y definimos las fórmulas análogas a las del ejercicio (adaptando el ejercicio original a la signatura más pequeña):

$$\varphi_1 := Q \wedge \neg(P \leftrightarrow \neg R),$$

$$\alpha := \neg P \vee R,$$

$$\beta := \neg Q \vee ((\neg P \vee \neg R) \wedge (R \vee P)) \quad (\text{candidata para } \neg\varphi_1),$$

$$\gamma := \neg R \vee (\neg P \leftrightarrow Q),$$

$$\delta := \neg P \wedge \neg(\neg P \vee R).$$

La fórmula $(\neg P \vee \neg R) \wedge (R \vee P)$ es equivalente a $(P \leftrightarrow \neg R)$, por lo que β representa exactamente $\neg Q \vee (P \leftrightarrow \neg R)$; es decir, formalmente, la negación de $\varphi_1 = Q \wedge \neg(P \leftrightarrow \neg R)$.

La tabla de verdad completa (V = verdadero, F = falso):

P	Q	R	φ_1	α	β	γ	δ
F	F	F	F	V	V	V	F
F	F	V	F	V	V	F	F
F	V	F	V	V	F	V	F
F	V	V	F	V	V	V	F
V	F	F	F	F	V	V	F
V	F	V	F	V	V	V	F
V	V	F	F	F	V	V	F
V	V	V	V	V	F	F	F

De la tabla obtenemos el número de modelos, es decir, el número de interpretaciones en las que la fórmula es verdadera:

- $|\text{Mod}(\varphi_1)| = 2$ (verdadera para las asignaciones $(P, Q, R) = (F, V, F)$ y (V, V, V))
- $|\text{Mod}(\alpha)| = 6$
- $|\text{Mod}(\beta)| = 6$
- $|\text{Mod}(\gamma)| = 6$
- $|\text{Mod}(\delta)| = 0$ (insatisfacible)

Interpretación y conclusiones

- $\alpha = \neg P \vee R$ no es tautológica (no es verdadera en todas las interpretaciones), pero se satisface en 6 de los 8 casos; por tanto, es frecuente, aunque no universalmente válida.