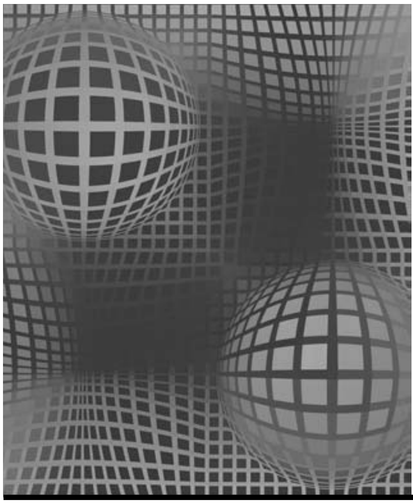

Capítulo 1

LÓGICA Y DEMOSTRACIONES

Lógica, lógica, lógica. La lógica es el principio de la sabiduría, Valeria, no el fin.

STAR TREK VI: EL PAÍS SIN DESCUBRIR

Lógica es el estudio del razonamiento; se refiere específicamente a si el razonamiento es correcto. La lógica se centra en la relación entre las afirmaciones y no en el contenido de una afirmación en particular. Considere, por ejemplo, el siguiente argumento:

Todos los matemáticos usan sandalias

Cualquiera que use sandalias es un algebrista

Por lo tanto, todos los matemáticos son algebristas

En el sentido técnico, la lógica no ayuda a determinar si alguna de estas afirmaciones es cierta; sin embargo, si las primeras dos afirmaciones son ciertas, la lógica asegura que la afirmación

WWW

todos los matemáticos son algebristas

también es cierta.

Los métodos lógicos se usan en matemáticas para demostrar teoremas y, en las ciencias de la computación, para probar que los programas hacen lo que deben hacer. Suponga, por ejemplo, que se asigna a un estudiante el desarrollo de un programa para calcular las trayectorias más cortas entre ciudades. Es necesario que el programa acepte como entrada un número arbitrario de ciudades y las distancias entre las ciudades con conexión directa por carretera, y que produzca como salida las trayectorias (rutas) más cortas entre cada par distinto de ciudades. Después de escribir el programa, es fácil para el estudiante probarlo con un número reducido de ciudades. Con papel y lápiz, puede enumerar todas las trayectorias posibles entre pares de ciudades y encontrar las más cortas. Esta solución por "fuerza bruta" se compara con la salida del programa. Sin embargo, para un número grande de ciudades, la técnica de la "fuerza bruta" sería tardada. ¿Cómo puede el estudiante estar seguro de que el programa trabaja bien para muchos datos (casi seguro el tipo de entrada con la que el profesor probaría el programa)? Él tendrá que usar la *lógica* para argumentar que el programa es correcto. El argumento puede ser informal o formal usando las técnicas presentadas en este capítulo; pero se requerirá un argumento lógico.

Entender la lógica también resulta útil para aclarar la escritura común. Por ejemplo, en una ocasión, se publicó el siguiente decreto en Naperville, Illinois: "Será ilegal que una

† Esta sección se puede omitir sin pérdida de continuidad.

persona tenga más de tres perros y tres gatos en su propiedad dentro de la ciudad". Un ciudadano que tenía cinco perros y ningún gato, ¿violaba el decreto? Piense en esta pregunta ahora y analícela (vea ejercicio 54, sección 1.1) después de leer la sección 1.1.

1.1 ➜ Proposiciones

¿Cuáles oraciones de la *a)* a la *e)* son verdaderas o falsas (pero no ambas)?

a) Los únicos enteros positivos que dividen† a 7 son 1 y el mismo 7.
b) Alfred Hitchcock ganó un premio de la Academia en 1940 por la dirección de "Rebeca".
c) Para todo entero positivo n, existe un número primo‡ mayor que n.
d) La Tierra es el único planeta en el universo que tiene vida.
e) Compra dos boletos para el concierto de rock "Unhinged Universe" del viernes.

La oración *a)*, que es otra manera de decir que el 7 es primo, es verdadera.

La oración *b)* es falsa. Aunque "Rebeca" ganó el premio de la Academia por la mejor película de 1940, John Ford ganó el premio por dirigir "Las viñas de la ira". Es un hecho sorprendente que Alfred Hitchcock nunca haya ganado un premio de la Academia por mejor dirección.

La oración *c)*, que es otra forma de decir que el número de primos es infinito, es verdadera.

La oración *d)* puede ser verdadera o falsa (pero no ambas), sin embargo en este momento se ignora.

La oración *e)* no es verdadera ni falsa [esta oración es una orden].

Una oración que es verdadera o falsa, pero no ambas, se llama una **proposición.** Las oraciones *a)* a la *d)* son proposiciones, mientras que la oración *e)* no es una proposición. Es común que una proposición se exprese como una oración declarativa (y no como pregunta, orden, exclamación, etcétera). Las proposiciones son los bloques básicos de construcción de cualquier teoría de lógica.

Se usarán variables, como p, q y r, para representar las proposiciones, casi como se usan letras en álgebra para representar números. También se usará la notación

$$p\text{: } 1 + 1 = 3$$

para definir que p es la proposición $1 + 1 = 3$.

Al hablar y escribir de forma normal, las proposiciones se combinan usando conectores como *y* y *o*. Por ejemplo, las proposiciones "está lloviendo" y "hace frío" se pueden combinar para formar la proposición "está lloviendo y hace frío". A continuación se dan las definiciones formales de *y* y *o*.

Definición 1.1.1 ▶

Sean p y q proposiciones.

La *conjunción* de p y q, denotada por $p \wedge q$, es la proposición

$$p \quad \text{y} \quad q.$$

La *disyunción* de p y q, denotada por $p \vee q$, es la proposición

$$p \quad \text{o} \quad q.$$ ◀

Un *operador binario* sobre un conjunto* X, asigna a cada par de elementos en X un elemento de X (vea la definición 2.2.44). El operador $\wedge$ asigna a cada par de proposiciones

†"Divide" se refiere a "división exacta". De manera más formal, se dice que un entero diferente de cero *d divide* a un entero m si existe un entero q tal que $m = dq$. A q se le llama el *cociente*. Se explorarán los enteros con detalle en el capítulo 5.

‡ Un entero $n > 1$ es *primo* si los únicos enteros positivos que dividen a n son 1 y el mismo n. Por ejemplo, 2, 3 y 11 son números primos.

* Un *conjunto* es una colección de objetos. Por ejemplo, el conjunto de enteros positivos consiste en los enteros 1, 2, … Los "conjuntos" se estudian con detalle en la sección 2.1.

p y q la proposición $p \wedge q$. Entonces, $\wedge$ es un operador binario sobre las proposiciones. El operador $\vee$ también es un operador binario sobre las proposiciones.

Ejemplo 1.1.2 ▶

Si

p: Está lloviendo,

q: Hace frío,

entonces la conjunción de p y q es

$p \wedge q$: Está lloviendo y hace frío.

La disyunción de p y q es

$p \vee q$: Está lloviendo o hace frío. ◀

El valor de verdad de la conjunción $p \wedge q$ está determinado por los valores verdaderos de p y q, y la definición se basa en la interpretación usual de "y". Considere la proposición

$p \wedge q$: Está lloviendo y hace frío

del ejemplo 1.1.2. Si está lloviendo (es decir, p es verdadera) y también hace frío (es decir, q también es verdadera), entonces la proposición

$p \wedge q$: Está lloviendo y hace frío

se consideraría verdadera. Sin embargo, si no está lloviendo (esto es, p es falsa) o si no hace frío (q es falsa) o ambas, entonces la proposición

$p \wedge q$: Está lloviendo y hace frío

se consideraría falsa.

Los valores de verdad de las proposiciones, tales como conjunciones o disyunciones, se pueden describir por las **tablas de verdad**. La tabla de verdad de una proposición P, formada por las proposiciones individuales $p_1, \ldots, p_n$, enumera todas las posibles combinaciones de los valores de verdad para $p_1, \ldots, p_n$, donde V denota verdadero y F denota falso, y da la lista de valores de verdad de P para cada combinación. Se usa una tabla de verdad para dar la definición formal de los valores de verdad de $p \wedge q$.

Definición 1.1.3 ▶

Los valores de verdad de la proposición $p \wedge q$ se definen por la tabla de verdad

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

◀

Observe que en la tabla de verdad de la definición 1.1.3 se dan las cuatro combinaciones posibles (cuatro alternativas posibles) de asignaciones de verdad para p y q.

La definición 1.1.3 establece que la conjunción $p \wedge q$ es verdadera siempre que p y q sean ambas verdaderas; de otra manera, $p \wedge q$ es falsa.

Ejemplo 1.1.4 ▶

Si

p: Una década tiene 10 años,

q: Un milenio tiene 100 años,

entonces p es verdadera, q es falsa (un milenio tiene 1000 años) y la conjunción

$p \wedge q$: Una década tiene 10 años y un milenio tiene 100 años

es falsa. ◀

Ejemplo 1.1.5 ▶

Casi todos los lenguajes de programación definen "y" justo como la definición 1.1.3. Por ejemplo, en el lenguaje de programación Java, el "y" (lógico) se denota por `&&`, y la expresión

```
x < 10 && y > 4
```

es verdadera precisamente cuando el valor de la variable `x` es menor que 10 (esto es, `x < 10` es cierta) y el valor de la variable `y` es mayor que 4 (es decir, `y > 4` se cumple). ◀

El valor de verdad de la disyunción $p \vee q$ también está determinado por los valores de verdad de p y q, y la definición se basa en la interpretación "inclusiva" de "o". Considere la proposición

$p \vee q$: Está lloviendo o hace frío,

del ejemplo 1.1.2. Si está lloviendo (es decir, p es verdadera) o si hace frío (es decir, q es verdadera) *o ambas*, entonces se consideraría que la proposición

$p \vee q$: Está lloviendo o hace frío

es verdadera (esto es, $p \vee q$ es verdadera). El **or-inclusivo** de las proposiciones p y q es verdadero si ambas, p y q, son verdaderas. Si no está lloviendo (o sea, p es falsa) y si no hace frío (q también es falsa), entonces se consideraría que la proposición

$p \vee q$: Está lloviendo o hace frío,

es falsa (esto es, $p \vee q$ es falsa). También existe el **or-exclusivo** (vea el ejercicio 53) que define p *exor* q como falsa si ambas, p y q, son verdaderas.

Definición 1.1.6 ▶

El valor de verdad de la proposición $p \vee q$ se define por la tabla de verdad

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

◀

La definición 1.1.6 establece que la disyunción $p \vee q$ es verdadera siempre que p o q (o ambas) sean verdaderas; de otra manera, $p \vee q$ será falsa (es decir, sólo si p y q son falsas la disyunción será falsa).

Ejemplo 1.1.7 ▶

Si

p: Un milenio tiene 100 años,
q: Un milenio tiene 1000 años,

entonces p es falsa, q es verdadera y la disyunción

$p \vee q$: Un milenio tiene 100 años o un milenio tiene 1000 años

es verdadera. ◀

Ejemplo 1.1.8 ▶

Casi todos los lenguajes de programación definen un or (inclusivo) justo como en la definición 1.1.6. Por ejemplo, en Java, el or (lógico) se denota por `||` y la expresión

```
x < 10 || y > 4
```

es verdadera precisamente cuando el valor de la variable `x` es menor que 10 (esto es, `x < 10` es cierta) o el valor de la variable y es mayor que 4 (es decir, `y > 4` se cumple) o ambas. ◀

En el lenguaje común, las proposiciones que se combinan (es decir, p y q combinadas para dar la proposición $p \vee q$) suelen estar relacionadas; pero en lógica, no se requiere que estas proposiciones hagan referencia al mismo asunto. Por ejemplo, en lógica se permiten proposiciones como

$3 < 5$ o París es la capital de Inglaterra.

La lógica se ocupa de la forma de las proposiciones y de la relación de las proposiciones entre sí, no del tema. (La proposición anterior es verdadera porque $3 < 5$ es verdadera).

El operador final en una proposición p que analizamos en esta sección es la **negación** de p.

Definición 1.1.9 ▶ La *negación* de p, denotada por $\neg p$, es la proposición

no p.

El valor de verdad de esta proposición $\neg p$ se define por la tabla de verdad

p	$\neg p$
V	F
F	V

◀

Algunas veces escribimos $\neg p$ para decir "no ocurre que p". Por ejemplo, si

p: París es la capital de Inglaterra,

la negación de p se escribe como

$\neg p$: No ocurre que París es la capital de Inglaterra.

o más fácil como

$\neg p$: París no es la capital de Inglaterra.

Un *operador unario* sobre un conjunto X asigna a cada elemento de X un elemento de X (vea la definición 2.2.46). El operador $\neg$ asigna a cada proposición p la proposición $\neg p$. Entonces, $\neg$ es un operador unario sobre las proposiciones.

Ejemplo 1.1.10 ▶ Si

p: π se calculó con 1,000,000 de dígitos decimales en 1954,

la negación de p es la proposición

$\neg p$: π no se calculó con 1,000,000 de dígitos decimales en 1954.

No fue sino hasta 1973 que se calculó π con 1,000,000 de dígitos decimales; entonces p es falsa. (Desde entonces se han calculado más de 200 mil millones de dígitos decimales de π). Puesto que p es falsa, $\neg p$ es verdadera. ◀

Ejemplo 1.1.11 ▶

Casi todos los lenguajes de programación definen “no” justo como en la definición 1.1.9. Por ejemplo, en Java el “no” se denota por !, y la expresión

```
! (x < 10)
```

es verdadera precisamente cuando el valor de la variable `x` no es menor que 10 (es decir, `x` es mayor que o igual a 10). ◀

En las expresiones que incluyen algunos o todos los operadores $\neg$, $\wedge$ y $\vee$, en la ausencia de paréntesis, primero se evalúa $\neg$, después $\wedge$ y luego $\vee$. Esta convención se conoce como **precedencia del operador**. En álgebra, la precedencia del operador indica que se evalúan $\cdot$ y $/$ antes que $+$ y $-$.

Ejemplo 1.1.12 ▶

Puesto que la proposición p es falsa, la proposición q es verdadera y la proposición r es falsa, determine si la proposición

$$\neg p \vee q \wedge r$$

es falsa o verdadera.

Primero se evalúa $\neg p$, que es verdadera. Después se evalúa $q \wedge r$, que es falsa. Por último, se evalúa

$$\neg p \vee q \wedge r$$

que es verdadera. ◀

Ejemplo 1.1.13 ▶

Búsqueda en Internet

Se dispone de gran variedad de herramientas de búsqueda en Internet (como AltaVista, Google, Yahoo) que permiten al usuario introducir palabras clave que el portal de búsqueda intenta igualar con páginas Web. Por ejemplo, introducir *matemáticas* produce una lista (¡enorme!) de páginas que contienen la palabra “matemáticas”. Algunos sitios de búsqueda permiten al usuario incluir operadores como *AND*, *OR* y *NOT* (*y*, *o* y *no*) junto con paréntesis para combinar las palabras clave (vea la figura 1.1.1), lo que admite búsquedas

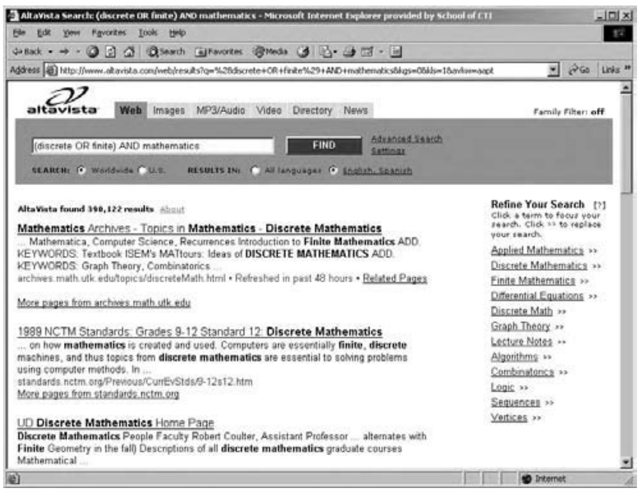

Figura 1.1.1 El portal de búsqueda AltaVista permite al usuario introducir expresiones con *AND*, *OR* y *NOT* junto con paréntesis. (En AltaVista, NOT debe ir precedido de otro operador como AND). En la figura, el usuario busca páginas que contengan “discrete mathematics” o “finite mathematics” (“matemáticas discretas” o “matemáticas finitas”) escribiendo *(discrete OR finite) AND mathematics*. Como se muestra, AltaVista encontró cerca de 390,000 páginas de Internet que contienen matemáticas discretas o matemáticas finitas.

más complejas. Por ejemplo, para buscar páginas que contengan las palabras clave “discretas” *y* “matemáticas”, el usuario escribiría *discretas AND matemáticas*. Para buscar páginas con las palabras clave “discretas” *y* “matemáticas” *o* las palabras clave “finitas” *y* “matemáticas”, el usuario podría introducir (*discretas OR finitas) AND matemáticas*. ◀

Sugerencias para resolver problemas

Aunque tal vez haya un camino más corto para determinar los valores de verdad de una proposición P formada al combinar las proposiciones $p_1, \ldots, p_n$ usando operadores como $\neg$ y $\vee$, la tabla de verdad siempre proporcionará todos los valores de verdad posibles de P para diferentes valores de las proposiciones que la constituyen $p_1, \ldots, p_n$.

Sección de ejercicios de repaso

†***1.*** ¿Qué es una proposición?

2. ¿Qué es una tabla de verdad?

3. ¿Qué es la conjunción de p y q? ¿Cómo se denota?

4. Proporcione la tabla de verdad para la conjunción de p y q.

5. ¿Qué es la disyunción de p y q? ¿Cómo se denota?

6. Proporcione la tabla de verdad para la disyunción de p y q.

7. ¿Qué es la negación de p? ¿Cómo se denota?

8. Proporcione la tabla de verdad para la negación de p.

Ejercicios

Determine si cada oración en los ejercicios 1 a 8 es una proposición. Si la oración es una proposición, escriba su negación. (No se piden los valores de verdad de las oraciones que son proposiciones).

1. $2 + 5 = 19$.

2. Mesero, ¿serviría las nueces, quiero decir, serviría las nueces a los invitados?

3. Para algún entero positivo n, $19340 = n \cdot 17$.

4. Audrey Meadows fue la “Alice” original de la serie “The Honeymooners”.

5. Pélame una uva.

6. La línea “Tócala otra vez, Sam” corresponde a la película “Casablanca”.

7. Todo entero par mayor que 4 es la suma de dos primos.

8. La diferencia de dos primos.

Los ejercicios 9 a 12 se refieren a una moneda que se lanza 10 veces. Escriba la negación de la proposición.

9. Salieron 10 caras.

10. Salieron algunas caras.

11. Salieron algunas caras y algunas cruces.

12. Salió al menos una cara.

Puesto que la proposición p es falsa, la proposición q es verdadera y la proposición r es falsa, determine si cada proposición en los ejercicios 13 a 18 es falsa o verdadera.

13. $p \vee q$

14. $\neg p \vee \neg q$

15. $\neg p \vee q$

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

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

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

Escriba la tabla de verdad de cada proposición en los ejercicios 19 a 26.

19. $p \wedge \neg q$

20. $(\neg p \vee \neg q) \vee p$

21. $(p \vee q) \wedge \neg p$

22. $(p \wedge q) \wedge \neg p$

23. $(p \wedge q) \vee (\neg p \vee q)$

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

25. $(p \vee q) \wedge (\neg p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee \neg q)$

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

En los ejercicios 27 a 29, represente la proposición indicada simbólicamente definiendo

$$p: 5 < 9, \quad q: 9 < 7, \quad r: 5 < 7.$$

Determine si cada proposición es verdadera o falsa.

27. $5 < 9$ y $9 < 7$.

28. No ocurre que ($5 < 9$ y $9 < 7$).

29. $5 < 9$ o no ocurre que ($9 < 7$ y $5 < 7$).

En los ejercicios 30 a 35, formule la expresión simbólica en palabras usando

p: Leo toma ciencias de la computación.

q: Leo toma matemáticas.

30. $\neg p$

31. $p \wedge q$

32. $p \vee q$

33. $p \vee \neg q$

34. $p \wedge \neg q$

35. $\neg p \wedge \neg q$

En los ejercicios 36 a 40, formule la expresión simbólica en palabras usando

p: Hoy es lunes.

q: Está lloviendo.

r: Hace calor.

36. $p \vee q$

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

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

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

† Los números de ejercicios en cursivas indican que se da una sugerencia o la solución al final del libro, después de la sección de referencias

40. $(p \wedge (q \vee r)) \wedge (r \vee (q \vee p))$

En los ejercicios 41 a 46, represente simbólicamente la proposición definiendo

p: Hay huracán.

q: Está lloviendo.

41. No hay huracán.

42. Hay huracán y está lloviendo.

43. Hay huracán, pero no está lloviendo.

44. No hay huracán y no está lloviendo.

45. Hay huracán o está lloviendo (o ambas).

46. Hay huracán o está lloviendo, pero no hay huracán.

En los ejercicios 47 a 52, represente simbólicamente la proposición definiendo

p: Oíste el concierto de rock de "Flying Pigs".

q: Oíste el concierto de rock de "Y2K".

r: Tienes los tímpanos inflamados.

47. Oíste el concierto de rock de "Flying Pigs" y tienes los tímpanos inflamados.

48. Oíste el concierto de rock de "Flying Pigs", pero no tienes los tímpanos inflamados.

49. Oíste el concierto de rock de "Flying Pigs", oíste el concierto de rock de "Y2K" y tienes los tímpanos inflamados.

50. Oíste el concierto de rock de "Flying Pigs" o el concierto de rock de "Y2K", pero no tienes los tímpanos inflamados.

51. No oíste el concierto de rock de "Flying Pigs" y no oíste el concierto de rock de "Y2K", pero tienes los tímpanos inflamados.

52. No ocurre que: oíste el concierto de rock de "Flying Pigs" o bien oíste el concierto de rock de "Y2K" o no tienes los tímpanos inflamados.

53. Proporcione una tabla de verdad para el or-exclusivo de p y q donde p *exor* q es verdadera si p o q, pero no ambas, son verdaderas.

54. En una ocasión se publicó el siguiente decreto en Naperville, Illinois: "Será ilegal que una persona tenga más de tres [3] perros y tres [3] gatos en su propiedad dentro de la ciudad". El señor Charles Marko tenía cinco perros y ningún gato, ¿violaba el decreto? Explique.

55. Escriba las instrucciones de búsqueda en Internet para encontrar parques nacionales en Dakota del Sur o del Norte.

56. Escriba las instrucciones de búsqueda en Internet para obtener información de enfermedades pulmonares que no sean cáncer.

57. Escriba las instrucciones de búsqueda en Internet para ver equipos de béisbol de las ligas menores que estén en la Liga del Medio Oeste.