Un semestre de MATEMÁTICA DISCRETA

Regino Criado y Roberto Muñoz

18 de mayo de 2007

Prólogo

El material que presentamos recoge una selección de contenidos de Matemática Discreta adecuada para el seguimiento de esta disciplina en, entre otros, los estudios de Ingeniería Técnica en Informática de Sistemas, Ingeniería Técnica en Informática de Gestión e Ingeniería Informática. Se ha utilizado como material docente en la Escuela Superior de Ciencias Experimentales y Tecnología de la Universidad Rey Juan Carlos desde el curso 2000/2001, en las titulaciones anteriormente señaladas. En el transcurso de este período de tiempo se han ido incorporando las reflexiones de numerosos profesores y alumnos que lo han utilizado. Este texto constituye, pues, la síntesis de un material trabajado y desarrollado desde distintas influencias y reflexiones.

Señalamos sus principales características.

- Es una selección de contenidos **realista** (de hecho se ha impartido en estos cursos) y **adecuada** (en el contexto de las titulaciones señaladas) pues sus contenidos se ajustan, en este nivel, a las últimas recomendaciones de algunas de las principales instituciones y asociaciones profesionales de informática y ciencias de la computación (*ACM* e *IEEE-CS*).

- Es un material globalmente **autocontenido** y sólo necesita conocimientos y herramientas elementales proporcionados por las matemáticas del bachillerato.

- Es un material **no exhaustivo** en el sentido de que los contenidos abordados no se desarrollan en su totalidad. Nuestro objetivo es dar fundamentos adecuados donde sustentar posteriores profundizaciones.

- Contiene una lista de **ejercicios resueltos** con las soluciones escritas al final de cada capítulo, lo que constituye una herramienta imprescindible en el aprendizaje de matemáticas.

- La mayoría de los resultados presentados van acompañados de su correspondiente **demostración**, escrita en una forma clara y sencilla, de modo que el enfoque adoptado en el texto no se limita a mostrar la matemática desde una perspectiva únicamente instrumental sino que la considera importante por su capacidad formativa.

- el texto aporta ejemplos diseñados con el programa de cálculo simbólico **Maple**.

- En el material se muestra una posible **secuencia temporal** para impartir sus contenidos. Esto resulta útil tanto al docente afanado en la preparación de cada clase como al estudiante presencial o, en su caso, seguidor a distancia del curso. Esta secuencia se ha llevado a la práctica en cursos sucesivos.

- La secuencia temporal adoptada se complementa con una propuesta de adecuación al **Espacio Educativo de Educación Superior** mediante su traslación a créditos *ECTS*.

- El texto se complementa con una **colección de exámenes** propuestos en años anteriores junto con su resolución, complemento que esperamos que sirva como un instrumento de medida de la preparación obtenida en el proceso de aprendizaje.

- En la **página web** de la asignatura

  http://www.escet.urjc.es/∼matemati/md_iti/md.html

  se encuentran materiales adicionales, exámenes recientes y prácticas de Maple complementarias.

Podemos decir que, en la experiencia de estos años, esta monografía se ha revelado como un instrumento útil en la preparación de los temas abordados en la misma y puede resultar adecuada para estudiantes y profesores de otras titulaciones que también necesitan de estos contenidos.

Queremos agradecer a los profesores que han impartido la asignatura en estos años sus constantes aportaciones. También a los alumnos que han señalado erratas, errores y han hecho acotaciones que mejoran el texto.

El segundo autor (R. M.) quiere agradecer a M. A. Fontelos su colaboración en la resolución de algunos de los ejercicios del texto.

El texto completo en su versión actualizada se puede encontrar en la página web

http://www.escet.urjc.es/∼matemati/materiales.html

y es intención de sus autores que sea un texto vivo que se someta a actualizaciones constantes.

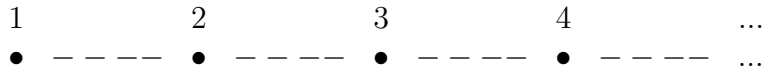
Introducción

Comenzamos esta introducción tratando de responder a la pregunta de **qué es la Matemática discreta**. La primera respuesta obvia es señalar que la matemática discreta es la parte de la matemática que estudia los objetos discretos. Esta respuesta no resulta muy satisfactoria pues seguimos sin conocer el significado del calificativo *discreto*. No resulta fácil, sin recurrir a definiciones formales que en este contexto resultarían muy oscuras, dar una definición de lo que es ser *discreto*, así que vamos a utilizar algunos ejemplos familiares para el lector.

Comenzamos con los números naturales, un objeto que, aunque matemáticamente precisa de cierto esfuerzo para su presentación, pertenece a nuestro bagaje cultural común:

$$\mathbb{N} = \{1, 2, 3, 4, 5, \dots\}.$$

Todos entendemos lo que es tener 5 sillas o que en clase haya 80 alumnos. Cuando los representamos solemos usar una semirrecta colocando a la derecha los números mayores:



Una vez que tenemos los números naturales, recurramos a un objeto matemático un poco más complejo, los números reales positivos $\mathbb{R}^+$. De nuevo éste es un conjunto conocido que contiene al conjunto de los números naturales, $\mathbb{N} \subset \mathbb{R}^+$, y es estrictamente más grande, $\mathbb{N} \neq \mathbb{R}$. Ahora contiene al cero, a las fracciónes y también a esos números *especiales* que no son fracciones (es decir que no se pueden escribir mediante un número decimal con una cantidad finita de cifras o con un período) como por ejemplo el número $\sqrt{2}$ o el número e.

También solemos representar a los números reales positivos mediante una semirrecta donde cada uno de sus puntos es un número real y hacia la derecha quedan ordenados de menor a mayor:

0 1 2 3 4 ...

Miremos con detenimiento un conjunto y otro. Hay cuantiosas diferencias entre ellos pero aquí vamos a centrarnos en las que establecen los conceptos de *discreto* (que se aplica a los números naturales) y de *continuo* (a los números reales).

Los números naturales son discretos, es decir, se disponen en la semirrecta que los representa *separados entre sí*. Del uno al dos hay un salto de una unidad. No hay ningún número natural entre ellos, es decir, no existe $n \in \mathbb{N}$ tal que $1 < n < 2$. Lo mismo ocurre entre el dos y el tres o entre dos cualesquiera números naturales consecutivos. Los conjuntos finitos o los números enteros $\mathbb{Z}$ (unión de los naturales el 0 y los negativos) tienen de igual manera esta propiedad.

Sin embargo, los números reales son muy distintos. Están todos *pegados*, no quedan huecos entre ellos, la recta que los representa está completamente punteada. Entre el uno y el dos no hay *agujeros*, siempre hay números entre ellos; está por ejemplo el 1.5 en medio. Entre el 1 y el 1.5 está por ejemplo el 1.2. Y entre el 1 y un número muy cercano a 1, pongamos por ejemplo el $1+10^{-10^7}$ (verdaderamente cercano ya que la distancia es cero coma un millón de ceros y luego un uno), hay números por medio, por ejemplo el $1 + 10^{-10^8}$. Valga esta aproximación a la noción de ser *continuo*: completamente pegados entre sí, sin espacios que los separan. De la formalización de este concepto es de donde parte la idea de límite y por tanto los conceptos de continuidad de funciones, derivada, integral y, en fin, todos los que constituyen un curso de Cálculo o Análisis Matemático, complementario a éste curso de Matemática Discreta en la formación de un ingeniero.

Con esta noción intuitiva acerca de qué es la matemática discreta la lista de contenidos del curso nos aclara un poco más:

- Lógica
- Algoritmos
- Aritmética y Aritmética modular

- Combinatoria
- Grafos
- Relaciones

Esto es, formalizar, diseñar procesos finitos para resolver problemas, analizar las propiedades fundamentales de los números enteros, contar, enumerar, relacionar conjuntos finitos, representar estas relaciones...

Una vez que hemos respondido (de forma intuitiva) a la pregunta sobre el concepto de matemática discreta se trata de responder ahora a la cuestión de **por qué estudiar matemática discreta**.

En primer lugar la matemática en general y la matemática discreta en particular tiene un carácter **instrumental**. Todas las ciencias usan la matemática como instrumento (para *medir, calcular, estimar, definir...*) e incluso como lenguaje de expresión (por ejemplo las ecuaciones diferenciales para la física). En concreto, la matemática discreta está ligada profundamente a las Ciencias de la Computación pues los objetos con los que trata un ordenador son objetos finitos (aunque pueden ser muy grandes) y por el triunfo de la tecnología digital, que en muchos aspectos es una representación discreta de la realidad. A los verbos anteriores objeto de la matemática discreta podemos asociar algún concepto relacionado con la informática: *formalizar* con *lenguajes de programación*, *procesos finitos* con *algoritmos*, *números enteros* con *criptografía*, *contar* con *estudiar la capacidad de un ordenador*, *enumerar* con *bases de datos*, *relacionar* con *redes*, *representar las relaciones* con *planos de redes...*

Por otro lado la matemática tiene un carácter **formativo** para un científico cualquiera y también para un ingeniero. Escribir con claridad, formalizar, adquirir destrezas para enfrentar situaciones nuevas, calcular y razonar con precisión, perserverar con constancia... son habilidades fundamentales para el científico o el ingeniero y para las que las matemáticas son un instrumento adecuado de desarrollo (evidentemente no el único).

Finalmente señalar que el curso está estructurado en 6 temas, cada tema tiene una pequeña introducción donde se explican los objetivos que pretendemos alcanzar, una relación autocontenida de los conceptos y resultados que hemos seleccionado, ejemplos ilustradores y ejercicios propuestos y resueltos. El capítulo 7 es una colección de exámenes propuestos y resueltos, material de apoyo para la preparación del curso.

0.1 Una propuesta de temporalización

Este curso de Matemática Discreta ha sido impartido en las titulaciones de Ingeniería Informática de Sistemas y Gestión en la Universidad Rey Juan Carlos desde el curso 2000/2001. Es un curso de 7.5 créditos que se imparte en 15 semanas, a razón de 5 horas semanales. Fundamentados en esta experiencia, proponemos una temporalización del mismo, indicando la materia que se puede explicar en cada sesión y actividades complementarias de tipo práctico (prácticas con Maple y ejercicios). En las sesiones prácticas resulta conveniente trabajar con grupos más pequeños de alumnos. Esta es la aportación más interesante, a nuestro entender, de este material: que responde a una selección cuidadosa de conceptos y herramientas que se puede impartir de forma efectiva en 75 horas de clases y es asimilable por el estudiante en el tiempo de un cuatrimestre. En este sentido el libro no tiene una vocación exhaustiva.

Pesentamos entonces el contenido de cada sesión (una propuesta que puede ser útil para el docente):

Sesión 1. (1 hora/ total 1 hora)

Presentación del curso, mediante el contenido de la página web de la asignatura:

http://www.escet.urjc.es/∼matemati/md_iti/md.html

Todos los materiales adicionales (hojas de ejercicios y prácticas con Maple) que se sugieren en esta temporalización pueden encontrarse en esta página.

Tema 1: Lógica. 11 horas

Sesión 2. (1 hora/ total 2 horas)

2.1) Lógica e informática
2.2) Lógica y ciencia
2.3) Lógica proposicional: definiciones básicas
2.4) Formas proposicionales

Sesión 3. (2 horas/ total 4 horas)

3.1) Tablas de verdad
3.2) Tautologías

3.3) Implicación lógica y equivalencia lógica
3.4) Razonamiento válido
3.5) Ejemplos

Sesión 4. (1 hora/ total 5 horas)

4.1) Demostración directa
4.2) Demostración por contrapositivo
4.3) Demostración por casos

Sesión 5. (2 horas/ total 7 horas)

5.1) Demostración por reducción al absurdo
5.2) Método de refutación
5.3) Lógica de predicados. Definiciones básicas
5.4) Cuantificadores existencial y universal. Contraejemplos
5.5) Modelización de frases del lenguaje natural

Sesión 6. (2 horas/ total 9 horas)

6.1) Formas de predicado
6.2) Asignación de valores de verdad a las formas de predicado
6.3) Ejercicios. Ejercicios adicionales: **Hoja 1**

Sesión 7. (1 hora/ total 10 horas)

Sesión de ejercicios.

Sesión 8. (2 horas/ total 12 horas)

8.1) El principio de inducción
8.2) Demostración por inducción
8.3) Inducción completa
8.4) Inducción estructural
8.5) Ejemplos

Trabajo personal. Lectura de la sección de aplicaciones

Tema 2: Algoritmos. 9 horas

Sesión 9. (2 horas/ total 14 horas)

9.1) Relación cliente/informático
9.2) Algoritmos, definición.
9.3) Modelo computacional
9.4) Pseudocódigo

9.5) Ejemplos

Sesión 10. (2 horas/ total 16 horas)

10.1) Algoritmos de ordenación
10.2) Número de operaciones en el peor de los casos
10.3) Complejidad. La notación O
10.4) Ejemplos

Sesión 11. (1 horas/ total 17 horas)

11.1) Más consideraciones sobre $O(T)$
11.2) Complejidad de algoritmos
11.3) Ejemplos

Sesión 12. (1 hora/ total 18 horas)

12.1) Ejemplos de cómputo de complejidad: búsqueda secuencial, búsqueda binaria, ordenación burbuja...
12.2) Complejidad de un problema

Sesión 13. (1 hora/ total 19 horas)

Sesión de ejercicios. Ejercicios adicionales: **Hoja 2**

Sesión 14. (2 horas/ total 21 horas)

Práctica con Maple: **Práctica 1**

Tema 3: Aritmética modular. 14 horas

Sesión 14. (2 horas/ total 23 horas)

14.1) Axiomática de los naturales. Estructura algebraica
14.2) Axiomática de los enteros. Estructura algebraica
14.3) Ordenación en ambos conjuntos
14.4) Teorema de la división entera

Sesión 15. (1 horas/ total 24 horas)

15.1) Divisibilidad. Propiedades
15.2) Máximo común divisor
15.3) Ejercicios

Sesión 16. (2 horas/ total 26 horas)

16.1) Primos

16.2) Algorimo de Euclides
16.3) Lema de Bezout
16.4) Factorización

Sesión 17. (2 horas/ total 28 horas)

17.1) Unicidad en la factorización
17.2) Ejercicios
17.3) Congruencia módulo un entero mayor que 1
17.4) Artimética modular

Sesión 18. (1 hora/ total 29 horas)

18.1) El conjunto cociente $\mathbb{Z}_p$
18.2) Ejemplos de aritmética modular

Sesión 19. (2 horas/ total 31 horas)

19.1) Más cuentas módulo p
19.2) Congruencias lineales. Lema de existencia del inverso
19.3) Sistemas de congruencias lineales. Teorema chino de los restos
19.4) Aplicaciones

Sesión 20. (1 hora/ total 32 horas)

20.1) Bases de numeración
20.2) Cambios de base
20.3) Aritmética en base 2

Sesión 21. (1 hora/ total 33 horas)

Sesión de ejercicios. Ejercicios adicionales: **Hoja 3**

Sesión 22. (2 horas/ total 35 horas)

Práctica con Maple: **Práctica 2**

Tema 4: Combinatoria. 11 horas

Sesión 23. (2 horas/ total 37 horas)

22.1) Combinatoria: contar y enumerar
22.2) Cardinales de conjuntos finitos
22.3) Propiedades de los cardinales
22.4) Ejemplos

Sesión 24. (2 horas/ total 39 horas)

24.1) Variaciones. Variaciones con repetición
24.2) Permutaciones. Permutaciones con repetición
24.3) Combinaciones. Combinaciones con repetición

Sesión 25. (1 hora/ total 40 horas)

25.1) Propiedades de los números combinatorios
25.2) Ejercicios. Ejercicios adicionales: **Hoja 4**

Sesión 26. (2 horas/ total 42 horas)

26.1) Probabilidad. Nociones básicas. Regla de Laplace
26.2) Asignación de probabilidades
26.3) Probabilidad condicionada

Trabajo personal. Leer las sección correspondiente a variables aleatorias.

Sesión 27. (2 horas/ total 44 horas)

27.1) Ejercicios
27.2) Experimentos de Bernoulli
27.3) Más ejercicios

Sesión 28. (2 horas/ total 46 horas)

Práctica con Maple: **Práctica 3**

Tema 5: Grafos. 18 horas

Sesión 29. (2 horas/ total 48 horas)

29.1) Los 7 puentes de Konisberg
29.2) Definiciones básicas
29.3) Adyacencias e incidencias
29.4) Grado de un vértice
29.5) Grados de entrada y salida

Sesión 30. (2 horas/ total 50 horas)

30.1) Grafos con nombre. Número de vértices y aristas en cada caso
30.2) Subgrafos, unión, producto y partición

Sesión 31. (1 hora/ total 51 horas)

31.1) Matrices de adyacencias
31.2) Matrices de incidencias

31.3) Ejercicios. Ejercicios adicionales **Hoja 5**

Sesión 32. (2 horas/ total 53 horas)

32.1) Caminos en grafos
32.2) Conexión. Componentes conexas
32.3) Caminos y matrices
32.4) Criterio de conexión

Sesión 33. (2 horas/ total 55 horas)

33.1) Grafos etiquetados
33.2) Problema del camino mínimo. Algoritmo de Dijkstra.
32.3) Ejercicios

Trabajo personal. Leer las secciones de grafos eulerianos y hamiltonianos

Sesión 34. (1 horas/ total 56 horas)

Control de los temas 1, 2 y 3. Ejemplos de estos controles se pueden ver en la página web o en el capítulo 7.

Sesión 35. (1 horas/ total 57 horas)

35.1) Árboles. Definiciones básicas.
35.2) Árboles con raíz.
35.3) Teorema de las hojas de un árbol m-ario

Sesión 36. (2 horas/ total 59 horas)

36.1) Árboles de búsqueda
36.2) Árboles de decisión
36.3) Árboles generadores
36.4) Árbol generador mínimo

Sesión 37. (1 horas/ total 60 horas)

Sesión de ejercicios

Sesión 38. (2 horas/ total 62 horas)

38.1) Otros aspectos de teoría de grafos (seleccionar los que más interesen, el resto puede ser trabajo personal)

Sesión 39. (2 horas/ total 64 horas)

Práctica de Maple. **Práctica 4**

Tema 6: Relaciones. 11 horas

Sesión 39. (1 hora/ total 65 horas)

39.1) Relaciones. Definiciones básicas
39.2) Relaciones de equivalencia
39.3) Ejemplos

Sesión 40. (2 horas/ total 67 horas)

40.1) Representación de relaciones. Conjuntos. Tablas. Grafos dirigidos. Matrices.
40.2) Las propiedades en cada representación
40.3) Clausura reflexiva y simétrica

Sesión 41. (2 horas/ total 69 horas)

41.1) Clausura transitiva. Dos algoritmos
41.2) Ejercicios. Ejercicios adicionales: **Hoja 6**

Sesión 42. (2 horas/ total 71 horas)

42.1) Conjuntos parcialmente ordenados
42.2) Diagramas de Hasse
42.3) Elementos característicos de un *POSET*
42.4) Ejemplos

Sesión 43. (2 horas/ total 73 horas)

43.1) Inclusión de un orden parcial en uno total
43.2) Aplicación a la gestión de tareas
43.3) Ejercicios

Sesión 44. (2 horas/ total 75 horas)

Retículos y álgebras de Boole

Trabajo Personal. Práctica con Maple: **Práctica 5**

0.2 Créditos ECTS

En el nuevo Espacio Europeo de Educación Superior, resultado del acuerdo de Bolonia, la unidad de medida de la carga docente de una asignatura es el *Crédito ECTS*. Esta nueva forma de medir adopta la perspectiva del alumno

y del trabajo que debe desarrollar, más que en el número de horas de clase de la asignatura que va a recibir.

En nuestra experiencia de enseñanza de esta asignatura podemos presentar el siguiente cómputo de créditos ECTS:

Concepto	**Número de Horas**	**Créditos ECTS**
Horas de contacto	75	2.5
Trabajo clases teóricas	$40 \times 2 = 80$	3.2
Trabajo clases prácticas	$10 \times 2 + 25 \times 1 = 45$	1.8
Entrega de problemas	$5 \times 4 = 20$	0.8
Consulta de la bibliografía	12.5	0.5

Lo que constituye un total de **8.5 créditos ECTS**.

Y donde se aplican los siguientes criterios de conversión:

- 30 horas de clase constituyen 1 crédito ECTS.
- Cada hora de clase teórica necesita dos horas de trabajo personal por parte del alumno, esto es, un total de 80 horas de trabajo personal.
- 25 horas de trabajo personal constituyen 1 crédito ECTS.
- Cada clase de laboratorio necesita dos horas de trabajo extra en el laboratorio, esto es, 20 horas de trabajo en el laboratorio.
- Cada clase de problemas necesita una hora de trabajo por parte del alumno, esto es, 25 horas de trabajo sobre los ejercicios.
- 25 horas de trabajo práctico (ejercicios o laboratorio) constituyen 1 crédito ECTS.
- Cada hoja de problemas necesita 4 horas extraordinarias de preparación, entonces 20 horas de trabajo sobre los ejercicios.
- 25 horas de biblioteca constituyen 1 crédito ECTS.

Índice General

Capítulo 1

Lógica

En este capítulo se presenta una introducción a la lógica proposicional y de predicados, se define con precisión lo que es un razonamiento válido y se muestran distintos métodos de demostración, entre ellos la inducción matemática. El capítulo contiene algunas aplicaciones de los temas tratados a la informática.

Como objetivos se espera que los alumnos sean capaces de realizar con soltura las siguientes actividades:

- Construir tablas de verdad de proposiciones compuestas.
- Averiguar si dos proposiciones son lógicamente equivalentes.
- Traducir enunciados del lenguaje natural a expresiones lógicas.
- Verificar si un razonamiento es correcto.
- Aplicar el principio de inducción.

1.1 Lógica e informática

¿Qué relación existe entre la lógica y la informática? Los ordenadores son máquinas diseñadas para mecanizar trabajos intelectuales, entre otros, los cálculos basados en operaciones aritméticas o el almacenamiento, clasificación y búsqueda de datos.

Al intentar mecanizar tareas más complejas entramos en el campo de la informática conocido como inteligencia artificial, en el que se pretende que

el ordenador sea capaz de realizar ese tipo de razonamientos que el hombre efectúa de una manera un tanto informal, por lo que es necesario definir y analizar con precisión dichos razonamientos. En otras palabras, se trata de formalizar los razonamientos. Y de esto se ocupa la lógica.

También la lógica tiene relación con el mundo de la programación debido a la denominada *crisis del software*: los programas son cada vez más complejos, menos fiables y más difíciles de mantener. Se han propuesto y se utilizan diversas metodologías para la construcción de programas y su verificación (métodos para comprobar la coincidencia entre lo que se cree que hace el programa y lo que realmente hace). En palabras de R. Fairley *la verificación formal es una rigurosa demostración matemática de la concordancia del código fuente de un programa con su especificación.* En esta línea de acercamiento entre el código fuente de un programa y su especificación están los denominados lenguajes de programación declarativa (como contraposición a la visión tradicional conocida como programación imperativa). Desde esta perspectiva se pretende que los programas no sean una secuencia de instrucciones que le digan al ordenador, paso a paso, cómo resolver el problema (programación imperativa), sino más bien una especificación de lo que se pretende resolver dejando que sea el propio ordenador el que determine las acciones necesarias para ello. En este sentido la lógica puede verse como un lenguaje de especificación mediante el cual podemos plantear los problemas de forma rigurosa.

Por otra parte, desde el punto de vista estrictamente electrónico, el soporte tecnológico principal de los ordenadores lo constituyen los *circuitos de conmutación* o *circuitos lógicos,* denominados así por tener en común con las formas elementales de la lógica el modelo matemático conocido como Álgebra de Boole. Esta estructura se presentará en el capítulo 6.

Finalmente, otra relación de la lógica con la informática viene dada por el hecho de que el estudio matemático de los lenguajes es uno de los pilares de la informática, entendiendo por lenguaje un *sistema de símbolos y de convenios que se utiliza para la comunicación, sea ésta entre personas, entre personas y máquinas, o entre máquinas*; la Lógica Formal puede considerarse como un lenguaje, *el mejor hecho de los lenguajes*, en palabras de Ferrater Mora.

1.2 Lógica y modelos matemáticos

Adoptemos una perspectiva más general a la de la sección anterior. Las ciencias experimentales y las ciencias sociales se basan en la observación de la naturaleza y utilizan un razonamiento de tipo *inductivo* para poder formular teorías generales. Parten de lo particular y se conducen a lo general. Las teorías generales formuladas permiten entender y clasificar los resultados de observaciones particulares. Introducen también una capacidad predictiva sobre el resultado de futuros experimentos.

El conocimiento científico es una representación del mundo real y alcanza altos niveles de precisión y exactitud cuando dichos modelos son de naturaleza matemática. La utilidad de los modelos matemáticos reside en las tres características que, según entendemos, configuran su esencia:

1. Por una parte la necesaria **simplificación** que se realiza en los modelos; permite analizar con claridad aspectos y relaciones que se presentan en el mundo real, prescindiendo de otras relaciones superfluas que ocultan y complican los aspectos objeto de estudio.

2. La segunda característica se refiere a la **facilidad para hacer deducciones** dentro del modelo. Es posible que una de las mayores dificultades del trabajo científico consista en elegir entre los distintos modelos o distintas representaciones del fenómeno objeto de estudio aquel modelo que más facilite la labor de deducción.

3. La última característica esencial de un modelo es extrínseca al mismo, a diferencia de las dos anteriores, y consiste en **su capacidad predictiva**, o lo que es lo mismo, el grado de coincidencia obtenido del contraste entre lo que estaba previsto dentro del modelo y lo que sucede en el mundo real. El método científico incorpora la experimentación para realizar dicho contraste.

Si las ciencias experimentales siguen un *modelo inductivo*, las matemáticas siguen un *modelo deductivo* de razonamiento. Es decir, desde una colección inicial de verdades, denominadas **axiomas**, se van obteniendo, mediante reglas correctas de deducción, más hechos verdaderos **teoremas**. Se puede decir que el objeto de la lógica es el estudio sistemático de las condiciones generales de validez de estas deducciones. Los conceptos de *deducción* y

demostración de un teorema refieren a una argumentación lógicamente irrefutable que establece que dicho teorema es verdadero. La elección de los axiomas no es, en principio, arbitraria, sino que trata de explicar el mundo real. Así por ejemplo, los números naturales surgieron de la necesidad de *contar* objetos.

Si asumimos, como primera aproximación, que una sentencia es *verdadera* si es *verificable experimentalmente* resultará sencillo comprobar la validez de la sentencia

$$3 + 2 = 5$$

para lo cual se puede considerar suficiente coger tres peras y dos manzanas y contar el número de objetos obtenidos al reunirlas. Sin embargo, si ahora queremos comprobar *experimentalmente* la validez de la sentencia

$$3 \cdot 10^{10} + 2 \cdot 10^{10} = 5 \cdot 10^{10}$$

nos encontraremos ante un serio problema.

Para resolverlo necesitamos construir un modelo o representación *abstracta* de los números naturales de manera que la validez de la sentencia

$$3 \cdot 10^{10} + 2 \cdot 10^{10} = 5 \cdot 10^{10}$$

se pueda demostrar a partir de las propiedades *que postulamos* para la suma de números naturales en esa representación (profundizaremos sobre esto en el capítulo 3).

Se comienza entonces con un conjunto de postulados o **axiomas** que establecen una serie de propiedades básicas de los objetos matemáticos bajo estudio. Estas propiedades se admiten como verdaderas, no se demuestran y responden a la descripción de los objetos en cuestión y a las propiedades que debieran tener. Los axiomas deben ser:

i) **compatibles**: no debe poderse deducir lógicamente a partir de ellos que una cierta sentencia es simultáneamente verdadera y falsa,

ii) **independientes**: ningún axioma se debe poder demostrar a partir del resto y

iii) **suficientes**: todas las propiedades que entendemos que deben satisfacer los objetos que estamos representando deben poderse deducir a partir de los axiomas.

Una vez establecidos los axiomas, el resto de *propiedades verdaderas* o *teoremas* se obtienen mediante deducciones lógicas o *demostraciones* a partir

de los axiomas o de otros teoremas previamente demostrados. Los términos lema, corolario y proposición se emplean para cierto tipo de teoremas. Un *lema* es un teorema más simple que se emplea para demostrar otro más complejo. Un *corolario* es un teorema que es consecuencia directa de otro teorema previo. El término *proposición* es sinónimo del término *teorema*. Se suele preferir la palabra teorema para hechos más relevantes y la palabra proposición para hechos más secundarios.

1.3 Lógica proposicional

1.3.1 Los conectivos lógicos

Al realizar razonamientos empleamos sentencias que están conectadas entre sí por conectivas lingüísticas. La lógica proposicional se ocupa del estudio de las conectivas lingüísticas entre proposiciones. Una **proposición** es una sentencia que puede ser verdadera, circunstancia que indicaremos asociándola el valor de verdad V, o falsa, en cuyo caso le asociaremos el valor de verdad F. Nuestro principal interés en relación con la lógica proposicional es dejar establecido el papel que juegan las conectivas lingüisticas y sus conectivos lógicos asociados en relación con los conceptos de *verdad* y *demostración*.

Las conectivas lingüísticas permiten construir proposiciones compuestas a partir de otras más simples. Así, si los símbolos p y q representan proposiciones genéricas, las conectivas lingüísticas más empleadas son las que aparecen en la siguiente tabla:

Conectiva lingüística	**Conectivo lógico**	**Símbolo**	**Se escribe**
no p	*Negación*	$\neg$	$\neg p$
p **y** q	*Conjunción*	$\wedge$	$p \wedge q$
p **o** q	*Disyunción*	$\vee$	$p \vee q$
Si p **entonces** q	*Implicación*	$\Rightarrow$	$p \Rightarrow q$
p **si y solo si** q	*Equivalencia*	$\Leftrightarrow$	$p \Leftrightarrow q$

- El significado del conectivo lógico negación, $\neg$, reside en que la proposición $\neg p$ es verdadera en el caso de que p sea falsa y recíprocamente que $\neg p$ es falsa cuando p es verdadera. Recogemos el significado del conectivo $\neg$ en la siguiente *tabla de verdad*:

p	$\neg p$
V	F
F	V

Del mismo modo podemos recoger el significado del resto de los conectivos:

- La sentencia $p \wedge q$ es verdadera sólo cuando p y q son verdaderas simultáneamente. La tabla de verdad correspondiente a este conectivo será, pues:

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

- La sentencia $p \vee q$, que se lee p *ó* q, debe entenderse en su acepción más amplia, es decir, p *ó* q *o ambos*, con lo que $p \vee q$ será verdadera en el caso de que p sea verdadera, q sea verdadera o ambas sean verdaderas simultáneamente:

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

- Para establecer el significado de la sentencia $p \Rightarrow q$, que se lee *si* p *entonces* q o también p *implica* q, debemos tener presente que en lenguaje natural la sentencia p *implica* q encierra una relación de causalidad, causalidad que no siempre aparece al utilizar este conectivo en el ámbito formal. Se pretende que el valor V o F de la sentencia $p \Rightarrow q$ dependa por completo de los valores de verdad de p y q, con independencia de

que exista o no alguna relación de causalidad con sentido entre p y q. En 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 y* p *verdadera*. Esto es, si el valor de verdad de p es V y el de q es F, el valor de verdad de $p \Rightarrow q$ es F; para el resto de posibles valores de p y q la sentencia será verdadera. Considérese el ejemplo siguiente:

Sea la proposición: *Si quemo madera, hay humo en el ambiente*, que representaremos por $p \Rightarrow q$. Entendemos que esta sentencia es verdadera, puesto que en nuestra experiencia no encontramos una situación en la que una madera esté ardiendo y no produzca humo. Sin embargo, la sentencia es verdadera independientemente de que se tenga madera a mano y no haya humo en el ambiente, de que no estemos quemando madera y haya humo debido a que estamos quemando papel, o incluso de que efectivamente estemos quemando madera y se esté llenando el ambiente de humo. En otras palabras $p \Rightarrow q$ sigue siendo verdadera aun en el caso en el que p sea falsa (no estemos quemando madera) y q sea falsa (no hay humo en el ambiente), p sea falsa y q sea verdadera o incluso que p sea verdadera y q sea verdadera. El significado de este conectivo lógico queda pues del siguiente modo:

p	q	$p \Rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Obsérvese que por ejemplo la sentencia *Si 2+2=4 entonces el sol es una estrella* es verdadera (según el significado del conectivo lógico establecido en la tabla anterior) y sin embargo no existe relación alguna de causalidad entre las proposiciones simples que en ella intervienen.

Es importante destacar que de la tabla anterior se sigue que para demostrar que la sentencia $p \Rightarrow q$ es verdadera, basta con estudiar el caso en el que p es verdadera, puesto que si p es falsa, la sentencia $p \Rightarrow q$ es verdadera.

A las sentencias p y q las denominaremos, respectivamente, **antecedente** y **consecuente** de la proposición $p \Rightarrow q$.

La sentencia $q \Rightarrow p$ es denominada **sentencia recíproca** de la sentencia $p \Rightarrow q$, y la sentencia $\neg q \Rightarrow \neg p$ **sentencia contrarrecíproca** de la sentencia $p \Rightarrow q$.

- La proposición $p \Leftrightarrow q$, que se lee p *si y solo* si q, o también p *equivale a* q es verdadera cuando p y q tienen el mismo valor de verdad, es decir:

p	q	$p \Leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

Para representar proposiciones emplearemos los símbolos p, q, r... o, lo que es lo mismo, emplearemos los símbolos p, q, r... como **variables proposicionales**.

1.3.2 Forma de una proposición

Uno de los descubrimientos más sorprendentes con el que nos encontramos al estudiar lógica consiste en que la validez de una deducción depende exclusivamente de la **forma** que ésta tenga, y no del posible significado de las proposiciones que en ella intervienen. Así por ejemplo, las deducciones:

keta es una roya o keta es larga
keta no es una roya
por lo tanto keta es larga

Cucufato estudia o Cucufato trabaja
Cucufato no estudia
por lo tanto Cucufato trabaja

son igualmente válidas, independientemente de que existan o no las ketas, de que sean o no largas, de que Cucufato estudie o trabaje o incluso de que exista alguien que se llame así.

Estas dos deducciones tienen la misma forma:

$$\frac{\begin{array}{c} p \vee q \\ \neg p \end{array}}{q}$$

Donde se está representando en cada línea una de las proposiciones, y se han separado las hipótesis de la conclusión por una línea horizontal. Sobre esto volveremos más adelante.

Veamos otro ejemplo para aclarar lo que entendemos por forma de una proposición:

las proposiciones *trabajo o no trabajo* y *estoy sentado o no estoy sentado* tienen la misma forma $p \vee \neg p$. La tabla de verdad de esta forma proposicional $p \vee \neg p$ es:

p	$\neg p$	$p \vee \neg p$
V	F	V
F	V	V

Es decir, cualquier proposición que tenga la forma $p \vee \neg p$ será verdadera independientemente del valor de verdad que tenga p.

La forma de las proposiciones va a estar determinada por cómo están dispuestos los conectivos $\{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\}$, por lo que vamos a introducir la definición de **forma proposicional** para estudiar aquellas propiedades que únicamente dependen de la manera en la que están colocados dichos conectivos lógicos:

Definición 1.3.1 *Llamaremos* **forma proposicional** *(o fórmula) a cualquier expresión formada por:*

a) variables proposicionales tales como p, q, r, ...

b) los conectivos lógicos $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

c) los paréntesis $(,)$

y construida utilizando las siguientes reglas:

1. *Cualquier variable proposicional es una forma proposicional*

2. *Si $\mathcal{A}$ y $\mathcal{B}$ son formas proposicionales, entonces también lo son* $(\neg\mathcal{A})$, $(\mathcal{A}\wedge\mathcal{B})$, $(\mathcal{A}\vee\mathcal{B})$, $(\mathcal{A}\Rightarrow\mathcal{B})$ *y* $(\mathcal{A}\Leftrightarrow\mathcal{B})$

En la práctica se suelen suprimir los paréntesis en aquellos casos en los que al hacerlo no se produce ambigüedad. Señalar también que, como veremos en la sección 1.5, la regla 2 puede ser simplificada prescindiendo de algunos de los conectivos.

Ejemplo 1.3.2 *La expresión*

$$\vee p\neg\vee\neg q$$

no es una forma proposicional (por ejemplo porque comienza por un conectivo $\vee$*), mientras que la expresión*

$$((p\wedge(\neg q))\Rightarrow r)$$

sí lo es porque sigue las reglas de su construcción.

Obsérvese que última forma proposicional puede ser representada sin ambigüedad suprimiendo paréntesis por

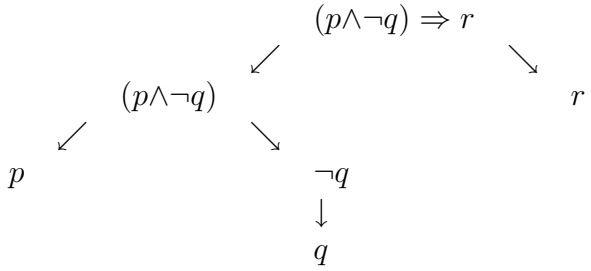
$$(p\wedge\neg q)\Rightarrow r.$$

Cada forma proposicional tiene asociada una tabla de verdad, que recoge los distintos valores de verdad de la forma al asignar valores V y F a las distintas variables involucradas. La tabla de verdad asociada a una forma proposicional se puede construir sistemáticamente a partir del procedimiento empleado para construir dicha forma proposicional:

Ejemplo 1.3.3 *Construcción de la tabla de verdad de la forma proposicional*

$$(p\wedge\neg q)\Rightarrow r$$

- Lo primero es determinar los pasos seguidos en su construcción. Obsérvese que según las reglas de construcción de formas proposicionales siempre se tiene un último conectivo que va separando la parte de la derecha y la de la izquierda que son proposiciones más simples (con menos conectivos):



- En segundo lugar se construye la primera fila de la tabla teniendo en cuenta dichos pasos:

p	q	$\neg q$	$(p\wedge\neg q)$	r	$(p\wedge\neg q)\Rightarrow r$

- En tercer lugar establecemos todas las posibles situaciones de verdad o falsedad de las primeras variables del enunciado que intervienen:

p	q	$\neg q$	$(p\wedge\neg q)$	r	$(p\wedge\neg q)\Rightarrow r$
V	V				
V	F				
F	V				
F	F				

- Finalmente se van rellenando el resto de las columnas teniendo en cuenta por una parte el *significado* de los conectivos lógicos $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$ y por otra, que cada vez que aparece una nueva variable proposicional es preciso comparar las situaciones de verdad o falsedad obtenidas con los dos posibles valores de verdad de la nueva variable proposicional:

(Paso intermedio)

p	q	$\neg q$	$(p \wedge \neg q)$	r	$(p \wedge \neg q) \Rightarrow r$
V	V	F	F	V	
V	F	V	V	V	
F	V	F	F	V	
F	F	V	F	V	
				F	
				F	
				F	
				F	

(Tabla final)

p	q	$\neg q$	$(p \wedge \neg q)$	r	$(p \wedge \neg q) \Rightarrow r$
V	V	F	F	V	V
V	F	V	V	V	V
F	V	F	F	V	V
F	F	V	F	V	V
V	V	F	F	F	V
V	F	V	V	F	F
F	V	F	F	F	V
F	F	V	F	F	V

Ejercicio 1 *Construir las tablas de verdad de las siguientes formas proposicionales:*

1. $\neg (p \vee q)$
2. $(p \wedge \neg p)$
3. $\neg (p \wedge \neg p)$
4. $p \Rightarrow (q \wedge \neg r)$
5. $\neg (p \wedge q) \Leftrightarrow ((\neg p) \vee (\neg q))$

1.3.3 Tautologías y razonamientos válidos

Definición 1.3.4 *Una forma proposicional es una* ***tautología*** *si toma el valor V cualquiera que sea la forma en que asignemos los valores V ó F a las*

variables proposicionales que en ella intervienen. Una forma proposicional es una ***contradicción*** *si toma el valor F cualquiera que sea la forma en que asignemos los valores V ó F a las variables proposicionales que en ella intervienen.*

Ejemplo 1.3.5 $(p \Rightarrow q) \Leftrightarrow ((\neg p) \vee q)$ *es una tautología según se sigue de su tabla de verdad:*

p	q	$p \Rightarrow q$	$\neg p$	$(\neg p) \vee q$	$(p \Rightarrow q) \Leftrightarrow ((\neg p) \vee q)$
V	V	V	F	V	V
V	F	F	F	F	V
F	V	V	V	V	V
F	F	V	V	V	V

Ejercicio 2 *Probar que las siguientes formas proposicionales son tautologías:*

1. $(p \wedge (p \Rightarrow q)) \Rightarrow q$
2. $((p \Rightarrow q) \wedge (q \Rightarrow r)) \Rightarrow (p \Rightarrow r)$
3. $((p \Rightarrow q) \wedge (\neg p \Rightarrow q)) \Rightarrow q$
4. $((p \vee q) \wedge \neg p) \Rightarrow q$
5. $(p \wedge \neg p) \Rightarrow q$

Ejercicio 3 *Probar que las siguientes formas proposicionales son tautologías:*

1. $((p \Rightarrow q) \wedge (q \Rightarrow p)) \Leftrightarrow (p \Leftrightarrow q)$
2. $(p \Rightarrow q) \Leftrightarrow ((\neg q) \Rightarrow (\neg p))$
3. $(p \Rightarrow q) \Leftrightarrow ((p \wedge \neg q) \Rightarrow (\neg p))$
4. $p \Leftrightarrow (\neg p \Rightarrow p)$
5. $p \Leftrightarrow ((\neg p) \Rightarrow (q \wedge \neg q))$

Definición 1.3.6 *Si $\mathcal{A}$ y $\mathcal{B}$ son formas proposicionales se dice que $\mathcal{A}$* **implica lógicamente a** *$\mathcal{B}$ si la forma proposicional $(\mathcal{A} \Rightarrow \mathcal{B})$ es una tautología.*

Otras formas habituales de decir $\mathcal{A}$ *implica lógicamente a* $\mathcal{B}$ son :

- $\mathcal{B}$ es *consecuencia lógica* de $\mathcal{A}$
- $\mathcal{A}$ es *condición suficiente* de $\mathcal{B}$
- $\mathcal{B}$ es *condición necesaria* de $\mathcal{A}$.

Analicemos el significado de esta definición:

si $(\mathcal{A} \Rightarrow \mathcal{B})$ es una tautología entonces es imposible asignar valores de verdad a las variables proposicionales de $\mathcal{A}$ y $\mathcal{B}$ de manera que $\mathcal{A}$ sea V y $\mathcal{B}$ sea F.

En otras palabras: Si $(\mathcal{A} \Rightarrow \mathcal{B})$ es una tautología y $\mathcal{A}$ es V, entonces necesariamente $\mathcal{B}$ es V.

Del ejercicio 2 tenemos que:

$p \wedge (p \Rightarrow q)$	*implica lógicamente* a	q
$(p \Rightarrow q) \wedge (q \Rightarrow r)$	*implica lógicamente* a	$(p \Rightarrow r)$
$(p \Rightarrow q) \wedge (\neg p \Rightarrow q)$	*implica lógicamente* a	q
$(p \vee q) \wedge \neg p$	*implica lógicamente* a	q
$(p \wedge \neg p)$	*implica lógicamente* a	q

Observación 1.3.7 *El hecho de que $p \wedge (p \Rightarrow q)$ implica lógicamente a q es la regla de inferencia que se suele conocer como* **modus ponens**.

La regla **modus tolens** *es el hecho de que $(p \Rightarrow q) \wedge \neg q$ implica lógicamente a $\neg p$.*

Definición 1.3.8 *Sean $\mathcal{A}$ y $\mathcal{B}$ dos formas proposicionales se dice que $\mathcal{A}$* **equivale lógicamente a** *$\mathcal{B}$ si la forma proposicional $(\mathcal{A} \Leftrightarrow \mathcal{B})$ es una tautología.*

Analicemos el significado de esta definición:

si $(\mathcal{A} \Leftrightarrow \mathcal{B})$ es una tautología entonces es imposible asignar valores de verdad a las variables proposicionales de $\mathcal{A}$ y $\mathcal{B}$ de manera que el valor de verdad de $\mathcal{A}$ sea diferente del valor de $\mathcal{B}$.

En otras palabras: si $(\mathcal{A} \Leftrightarrow \mathcal{B})$ es una tautología $\mathcal{A}$ es V si y solo si $\mathcal{B}$ es V, y $\mathcal{A}$ es F si y solo si $\mathcal{B}$ es F.

Ejercicio 4 *Probar que los siguientes pares de formas proposicionales son lógicamente equivalentes*

$\neg\neg p$	y	p	
$(p \wedge q) \wedge r$	y	$p \wedge (q \wedge r)$	*($\wedge$ es asociativa)*
$(p \vee q) \vee r$	y	$p \vee (q \vee r)$	*($\vee$ es asociativa)*
$(p \wedge q)$	y	$(q \wedge p)$	*($\wedge$ es conmutativa)*
$(p \vee q)$	y	$(q \vee p)$	*($\vee$ es conmutativa)*
$p \wedge (q \vee r)$	y	$(p \wedge q) \vee (p \wedge r)$	*($\wedge$ es distributiva respecto de $\vee$)*
$p \vee (q \wedge r)$	y	$(p \vee q) \wedge (p \vee r)$	*($\vee$ es distributiva respecto de $\wedge$)*
$\neg (p \vee q)$	y	$(\neg p) \wedge (\neg q)$	*ley de De Morgan*
$\neg (p \wedge q)$	y	$(\neg p) \vee (\neg q)$	*ley de De Morgan*
$(p \Leftrightarrow q)$	y	$(q \Leftrightarrow p)$	
$p \wedge V$	y	p	
$p \vee F$	y	p	
$(p \Rightarrow q)$	y	$(p \wedge \neg q) \Rightarrow F$	

Del ejercicio 3 tenemos que:

$(p \Rightarrow q) \wedge (q \Rightarrow p)$	*equivale lógicamente* a	$(p \Leftrightarrow q)$
$(p \Rightarrow q)$	*equivale lógicamente* a	$(\neg q) \Rightarrow (\neg p)$
$(p \Rightarrow q)$	*equivale lógicamente* a	$((p \wedge \neg q) \Rightarrow (\neg p))$
p	*equivale lógicamente* a	$(\neg p \Rightarrow p)$
p	*equivale lógicamente* a	$((\neg p) \Rightarrow (q \wedge \neg q))$

Por otro lado el ejemplo 1.3.5 indica que $p \Rightarrow q$ es equivalente lógicamente a $\neg p \vee q$.

Esta tabla de equivalencias lógicas pone de manifiesto los siguientes resultados:

1. La equivalencia $(p \Leftrightarrow q)$ es V si y solo si las implicaciones $(p \Rightarrow q)$ y $(q \Rightarrow p)$ son V.

2. La implicación $(p \Rightarrow q)$ es lógicamente equivalente a su *contrarrecíproca* $(\neg q) \Rightarrow (\neg p)$.

3. El *principio de reducción al absurdo*: la verdad de p es equivalente al hecho de que su negación $\neg p$ implique una contradicción. Este hecho se sustenta en la siguiente equivalencia lógica, presentada anteriormente:

   p *equivale lógicamente* a $((\neg p) \Rightarrow (q \wedge \neg q))$

Estas equivalencias lógicas nos permitirán presentar algunas técnicas de demostración. Antes introduzcamos con precisión el sentido que tiene el hecho de que un razonamiento sea correcto.

Definición 1.3.9 *Sean $\mathcal{A}_1, ..., \mathcal{A}_n$ y $\mathcal{B}$ formas proposicionales. Se dice que un razonamiento del tipo*

$$de \quad \mathcal{A}_1, ..., \mathcal{A}_n \quad se \quad deduce \quad \mathcal{B}$$

es **correcto** *o* **lógicamente válido** *si $\mathcal{A}_1 \wedge ... \wedge \mathcal{A}_n$ implica lógicamente $\mathcal{B}$ o, lo que es lo mismo, si es imposible encontrar un caso en el que siendo verdaderas las sentencias $\mathcal{A}_1$,...,$\mathcal{A}_n$ sea falsa la sentencia $\mathcal{B}$. A las sentencias $\mathcal{A}_1$,...,$\mathcal{A}_n$ las denominamos* **premisas** *o* **hipótesis** *y a la sentencia $\mathcal{B}$* **conclusión** *o* **tesis**.

Para distinguir las premisas de la conclusión, el razonamiento anterior se suele representar por

$$\begin{array}{c} \mathcal{A}_1 \\ \vdots \\ \mathcal{A}_n \\ \hline \mathcal{B} \end{array}$$

Ejemplo 1.3.10 *El siguiente razonamiento es correcto:*

$$\begin{array}{c} p \Rightarrow q \\ p \\ \hline q \end{array}$$

Ya que $((p \Rightarrow q) \wedge p) \Rightarrow q$ es una tautología.

Ejercicio 5 *Determinar si el siguiente razonamiento es correcto:*

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

Ejercicio 6 *Identificar adecuadamente las distintas conectivas lingüísticas y las variables proposicionales, obtener las premisas y la conclusión y estudiar la validez de los siguientes razonamientos:*

1. *H es un subgrupo si es no vacío, contiene al neutro y es cerrado para la operación. Si H es no vacío y cerrado para la operación, entonces H contiene al neutro. En consecuencia, si H es cerrado para la operación, H es un subgrupo.*

2. *Si a es un número perfecto entonces a es par y n es impar. a es par y n no es impar. Por consiguiente a no es perfecto.*

3. *Si no hay huelga, o el sindicato miente o el ministro tiene razón. Hay huelga o en caso contrario el ministro se equivoca. Por lo tanto el sindicato no miente.*

Ejercicio 7 *Determinar si el siguiente razonamiento es correcto:*

Si José ganó la carrera entonces Pedro fue el segundo o Ramón fue el segundo. Si Pedro fue el segundo entonces José no ganó la carrera. Si Carlos fue el segundo entonces Ramón no fue el segundo. José ganó la carrera. Luego Carlos no fue el segundo.

De esta manera, a partir de proposiciones verdaderas (que inicialmente conforman los axiomas) se pueden utilizar razonamientos válidos para demostrar que una nueva proposición es también verdadera. Veamos algunos ejemplos de demostraciones tipo:

- **Demostraciones directas:** se usan para probar que una sentencia de la forma $p \Rightarrow q$ es verdadera para lo cual utilizamos la siguiente tautología recursivamente:

$$((p \Rightarrow r) \wedge (r \Rightarrow s)) \Rightarrow (p \Rightarrow s).$$

  De esta manera si podemos construir una cadena finita de sentencias verdaderas de la forma $p \Rightarrow p_1$, $p_1 \Rightarrow p_2$, ..., $p_n \Rightarrow q$ habremos demostrado que $p \Rightarrow q$ es verdadera.

Ejemplo 1.3.11 *Demostremos que si x es un entero impar su cuadrado es impar.*
En efecto, si x es impar entonces existe un entero p tal que

$$x = 2p + 1$$

por la definición de ser impar. Formalmente hemos demostrado que si la proposición q es x *es impar y q_1 es la proposición* existe p *entero tal que $x=2p+1$ entonces tenemos la veracidad de:*

$$q \Rightarrow q_1.$$

De hecho, por definición de impar, $q \Leftrightarrow q_1$.
Aplicando las reglas para hacer el cuadrado tenemos:

$$x^2 = (2p+1)^2 = 4p^2 + 4p + 1.$$

Esta es una implicación $q_1 \Rightarrow q_2$ donde q_2 es la proposición existe p entero tal que x^2 se puede escribir como $x^2 = 4p^2 + 4p + 1$.
Sacando factor común obtenemos:

$$x^2 = 2(2p^2 + 2p) + 1.$$

Por lo que en efecto x^2 es impar.

Ejercicio 8 *Sean a, b dos números naturales $a, b \in \mathbb{N} = \{1, 2, 3...\}$. Se define que $a > b$ si existe un número natural $c \in \mathbb{N}$ tal que $a = b + c$. Demostrar que si $a > b$ y $b > c$ entonces $a > c$.*

- **Demostración por contrapositivo:** consiste en utilizar la equivalencia lógica de las sentencias $p \Rightarrow q$ y $\neg q \Rightarrow \neg p$.

Ejemplo 1.3.12 *Vamos a demostrar, usando la demostración por contrapositivo, la implicación recíproca del ejemplo anterior:*

$$x^2 \textit{ impar implica } x \textit{ impar.}$$

Debemos demostrar entonces que es verdadera la sentencia contrarrecíproca:

$$x \textit{ no impar implica } x^2 \textit{ no impar.}$$

O equivalentemente:

$$x \textit{ par implica } x^2 \textit{ par.}$$

Lo que se hace mediante una demostración directa idéntica a la del ejemplo anterior, cambiando la definición de impar por la de par.

- **Demostración por reducción al absurdo:** consiste en utilizar la equivalencia lógica escrita anteriormente

$$p \Leftrightarrow (\neg p \Rightarrow (q \wedge \neg q)).$$

Esto es, suponer que la hipótesis es verdadera, la conclusión es falsa y llegar a una contradicción.

Ejemplo 1.3.13 *Demostramos el siguiente hecho: si a es un número real, $a > 0$, entonces $\frac{1}{a} > 0$.*

Usamos la propiedad de los signos que indica que el producto de dos números reales es posisitivo si y solamente si o ambos son positivos o ambos son negativos.

Supongamos que la conclusión es falsa y la hipótesis es verdadera, esto es, $a > 0$ y $\frac{1}{a} \leq 0$. Es evidente que $\frac{1}{a} \neq 0$ pues $a\frac{1}{a} = 1$. Por tanto, como el producto $a\frac{1}{a} = 1$, entonces $a < 0$ en contradicción con la hipótesis $a > 0$.

- **Demostración por casos:** cuando se tiene una implicación del tipo $p \Rightarrow a_1 \vee a_2 \vee ... \vee a_n$ para demostrar que $p \Rightarrow q$ es suficiente demostrar que $a_1 \Rightarrow q$ y que ... $a_n \Rightarrow q$.

Ejercicio 9 *Determinar la tautología que justifica el método de demostración por casos. Hacer la tabla de verdad para $n = 2$.*

Ejemplo 1.3.14 *Sea $x \in \mathbb{N}$ demostramos que el resto de dividir x^2 entre 4 sólo puede tomar los valores 0 ó 1.*

Como se tiene que si $x \in \mathbb{N}$ entonces x es par o x es impar, entonces debemos probar que:

si x es par entonces se cumple la afirmación y que
si x es impar entonces también se cumple la afirmación.

En efecto, si x es par entonces existe $p \in \mathbb{N}$ de modo que

$$x = 2p$$

y por tanto

$$x^2 = 4p^2$$

de modo que x es múltiplo de 4, esto es, su resto al dividir por 4 es 0.

Y si x es impar entonces existe $p \in \mathbb{N}$ de modo que

$$x = 2p + 1$$

y por tanto

$$x^2 = 4p^2 + 4p + 1 = 4(p^2 + p) + 1$$

por lo que el resto de dividir por 4 es 1.

- **Demostración de dobles implicaciones:** La demostración de una doble implicación $p \Leftrightarrow q$ se convierte, mediante la equivalencia lógica con $(p \Rightarrow q) \wedge (q \Rightarrow p)$, en dos demostraciones. La de $p \Rightarrow q$ y la de $q \Rightarrow p$.

Ejemplo 1.3.15 *Mirando los ejemplos presentados en la demostración directa y en la demostración usando la sentencia contrarrecíproca se ha demostrado el hecho siguiente: x es impar si y solamente si x^2 es impar.*

1.3.4 El método de refutación

El método de refutación sirve para determinar si una forma proposicional es o no una tautología sin necesidad de obtener su tabla de verdad completa.

Este método consiste en intentar probar que la forma proposicional dada no es una tautología buscando aquellos valores de las variables proposicionales que hacen falsa la forma proposicional dada. Si vemos que esto es imposible, la forma proposicional objeto de estudio será necesariamente una tautología. En caso contrario no lo será y habremos encontrado unos posibles valores de verdad de las variables proposicionales que hacen que no lo sea.

Es importante señalar que este método es especialmente fecundo debido a que el mecanismo básico que emplea consiste en reducir la fórmula dada a una expresión cada vez más simple.

Notación. Si $\mathcal{A}$ es una forma proposicional entonces $\mathcal{A} : V$ (respectivamente $\mathcal{A} : F$) significa que consideramos la proposición $\mathcal{A}$ verdadera (respectivamente falsa). Cuando la forma es una variable, digamos p, a veces escribiremos $p := V$ (resp. $p := F$).

Ejemplo 1.3.16 *Vamos a demostrar que la siguiente proposición es una tautología utilizando el método de refutación:*

$$((p \Rightarrow q) \vee (r \Rightarrow q)) \Leftrightarrow ((p \wedge r) \Rightarrow q)$$

Supongamos que su valor de verdad es F y vamos a ir deduciendo los valores de verdad que deben tener las variables proposicionales p, q y r para

que eso ocurra. Si

$$((p \Rightarrow q) \vee (r \Rightarrow q)) \Leftrightarrow ((p \wedge r) \Rightarrow q) : F,$$

caben dos posibilidades, que iremos desarrollando paso a paso:

o bien	*o bien*
$(p \Rightarrow q) \vee (r \Rightarrow q) : V$	$(p \Rightarrow q) \vee (r \Rightarrow q) : F$
y $((p \wedge r) \Rightarrow q) : F$	*y* $(p \wedge r) \Rightarrow q : V$
de donde	*de donde*
$(p \Rightarrow q) \vee (r \Rightarrow q) : V$	$(p \Rightarrow q) : F$,$(r \Rightarrow q) : F$
$(p \wedge r) : V, q : F$	$(p \wedge r) \Rightarrow q : V$
luego	*luego*
$(p \Rightarrow F) \vee (r \Rightarrow F) : V$	$p : V$, $r : V$, $q : F$
$p : V$, $r : V$	$(p \wedge r) \Rightarrow q : V$
por lo que	*por lo que*
$(V \Rightarrow F) \vee (V \Rightarrow F) : V$	$(V \wedge V) \Rightarrow F : V$
es decir	*es decir*
F: V (contradicción)	$F : V$ *(contradicción)*

Luego falla el intento de probar que ***no*** *es una tautología, o lo que es lo mismo, no es posible asignar valores de verdad a las variables proposicionales que intervienen de manera que la forma proposicional* $((p \Rightarrow q) \vee (r \Rightarrow q)) \Leftrightarrow ((p \wedge r) \Rightarrow q)$ *sea falsa, por lo que concluimos que dicha forma proposicional es, efectivamente, una tautología.*

Ejemplo 1.3.17 *La sentencia*

$$((p \Rightarrow q) \wedge q) \Rightarrow p$$

no es una tautología. En efecto, tomamos $((p \Rightarrow q) \wedge q) : V$ *y* $p : F$. *Como* $((p \Rightarrow q) \wedge q) : V$ *entonces* $q : V$ *y* $(p \Rightarrow q) : V$. *Como* $p : F$ *entonces efectivamente* $(p \Rightarrow q) : V$. *De este modo* $p : F$ *y* $q : V$ *hacen que la sentencia sea* F *y por tanto no estemos ante una tautología.*

1.3.5 Conjuntos adecuados de conectivos en lógica de proposiciones

Es posible, utilizando las equivalencias lógicas, expresar cualquier forma proposicional mediante una forma proposicional equivalente en la que única-

mente aparezcan variables proposicionales y ciertos conectivos lógicos seleccionados como primitivos. Así por ejemplo, no es difícil demostrar que para cualquier forma proposicional $\mathcal{A}$ es posible encontrar una forma proposicional equivalente en la que los únicos conectivos que aparezcan sean los del conjunto $\{\vee, \wedge, \neg\}$. Por ejemplo, si $\mathcal{A}$ es la forma proposicional $\mathcal{B} \Rightarrow \mathcal{C}$, tendremos que la forma proposicional $\mathcal{A}$ es lógicamente equivalente a $(\neg\mathcal{B} \vee \mathcal{C})$. Si $\mathcal{A}$ es la forma proposicional $\mathcal{B} \Leftrightarrow \mathcal{C}$, tendremos que la forma proposicional $\mathcal{A}$ es lógicamente equivalente a $(\neg\mathcal{B} \vee \mathcal{C}) \wedge (\neg\mathcal{C} \vee \mathcal{B})$.

Por eso se dice que los conectivos $\{\vee, \wedge, \neg\}$ constituyen un **conjunto adecuado de conectivos**. Utilizando las leyes de De Morgan como axiomas es posible mostrar que tanto $\{\vee, \neg\}$ como $\{\wedge, \neg\}$ constituyen conjuntos adecuados de conectivos.

Existen conjuntos adecuados de conectivos que constan únicamente de un elemento. Sheffer introdujo en 1913 dos nuevos conectivos, que actualmente se conocen como *nand* y *nor*. Estos dos conectivos tienen gran utilidad en las aplicaciones a los circuitos de conmutación. El conectivo nor se representa con el símbolo $\downarrow$, siendo la sentencia $p \downarrow q$ lógicamente equivalente a $\neg(p \vee q)$. El conectivo nand se representa con el símbolo $|$, siendo $p \mid q$ lógicamente equivalente a $\neg(p \wedge q)$.

Ejercicio 10 *Comprobar que tanto* $\{\downarrow\}$ *como* $\{|\}$ *son conjuntos adecuados de conectivos. Para verlo expresar las formas proposicionales* $p \vee q$, $p \wedge q$ *y* $\neg p$ *en función, en primer lugar, del conectivo* $\{\downarrow\}$ *y, en segundo lugar, en función del conectivo* $\{|\}$.

1.4 Lógica de predicados

1.4.1 Introducción: predicados y objetos

Hay deducciones que se realizan habitualmente en matemáticas para las que no es posible analizar su validez dentro del ámbito de la lógica de proposiciones. Por ejemplo, si consideramos la deducción:

Todos los hombres son mortales
Sócrates es un hombre
Sócrates es mortal

e intentamos *formalizarla* (esto es, expresar su forma) en lógica de proposiciones, obtendríamos:

$$\begin{array}{l} p \\ q \\ \hline r \end{array}$$

por lo que no es un razonamiento válido (tomando $p : V$, $q : V$ y $r : F$) en contradicción con nuestra intuición. Necesitamos entonces una descripción más fina que permita distinguir los *hombres* (los objetos) de sus *propiedades* (predicados).
Consideremos ahora la deducción:

Todos los números primos son impares
3 es un número primo
3 es impar

Si separamos las sentencias:

3 es primo
5 es primo
5 es impar

observamos que, aun siendo distintas, aparecen en ellas partes comunes:

- en las dos primeras interviene la propiedad *ser primo*
- en las dos últimas interviene el objeto *5*.

Estas sentencias pueden ser simbolizadas del siguiente modo:

x es primo	*P(x)*
x es impar	*I(x)*
3 es primo	*P(3)*
5 es impar	*I(5)*
5 es primo	*P(5)*

O en el ejemplo del comienzo:

x es mortal	*M(x)*
x es hombre	*H(x)*

También es posible simbolizar de este modo propiedades que afectan a varios objetos:

x es el padre de y	*P(x,y)*
x estudia la carrera y	*E(x,y)*
x,y,z están alineados	*L(x,y,z)*
x es igual a y	$x=y$
x es mayor que y	$x > y$

En lógica de predicados se distingue, pues, entre las *propiedades* (también llamadas *predicados*) y los *objetos* a los que dichas propiedades se refieren.

En lógica, y en general, en matemáticas, se utilizan dos tipos de objetos:

- las **constantes**, que son objetos concretos y forman un universo de discurso, un conjunto U,
- las **variables**, que son objetos genéricos que normalmente denotamos con las letras $x, y, z....$ y que podrán sustituirse por objetos de U.

Por ejemplo, tomando el conjunto de los números naturales $U = \mathbb{N}$, el 1 es una constante y representa un elemento de $\mathbb{N}$, mientras que podemos escribir x, una variable que puede tomar cualquier valor natural.

La característica esencial que diferencia las variables de las constantes es que las variables pueden ser sustituidas por cualquier objeto del **universo de discurso** en el transcurso de una deducción.

Un **predicado** es entonces una sentencia que involucra variables de modo que al ser sustituidas por constantes se convierte en una proposición.

Por ejemplo, el predicado *x es primo* se convierte en una proposición al sustituir x por un número natural.

Los predicados se pueden representar por símbolos del tipo $P(x)$, $Q(x)$, $R(x)...$ que se denominan **funciones proposicionales** o también **sentencias abiertas**. Una vez que se haya asignado un valor adecuado a la variable x la función proposicional $P(x)$ da lugar a una proposición de la que es posible afirmar si es verdadera o falsa. Por ejemplo, si $P(x)$:=*x es primo* entonces $P(3)$ es verdadera, y $P(4)$ es falsa.

Por otra las funciones proposicionales pueden tener más de una variable. Así por ejemplo, si $Q(x, y, z)$ designa la sentencia $x + y = z$, tendremos que las sentencias $Q(1, 2, 3)$ y $Q(2, 2, 4)$ son verdaderas y que $Q(1, 2, 5)$ es falsa.

1.4.2 Los cuantificadores universal y existencial

Como hemos visto, una función proposicional (o sentencia abierta) puede dar lugar a una proposición (o, lo que es lo mismo, se puede cerrar) asignando valores concretos del universo de discurso a las variables que en ella intervienen. Introducimos ahora otra manera de cerrar las sentencias abiertas, usando los cuantificadores universal y existencial.

1) Consideremos la expresión: *Para todo objeto x de un universo de discurso U se verifica $P(x)$.* La frase *para todo objeto x de U* se denomina **cuantificador universal** y se representa por $\forall x \in U$.

Si la función proposicional $P(x)$ resulta ser una proposición verdadera al sustituir la variable x por cualquier elemento del universo de discurso U, diremos que *para todo elemento x de U la proposición $P(x)$ es verdadera.* Podemos escribir abreviadamente la siguiente sentencia que resulta ser verdadera:

$$\forall x \in U \ P(x).$$

2) Por otra parte, el hecho de que para al menos un elemento a del universo de discurso U la sentencia $P(a)$ sea una proposición verdadera, nos permite afirmar que la proposición *existe un elemento x de U tal que la proposición $P(x)$ es verdadera*, que expresamos de forma más breve por

$$\exists x \in U \ P(x)$$

es verdadera.

La frase *existe un objeto x del conjunto U* se denomina **cuantificador existencial** y se representa por $\exists x \in U$.

Así pues, para poder asociar un valor de verdad (o, lo que es lo mismo, interpretar) a una sentencia abierta, es necesario cerrarla previamente (se dice que una sentencia es cerrada si no tiene variables sin cuantificar). Como hemos visto una sentencia se puede cerrar asignando constantes a las variables que en ella intervienen o cuantificando dichas variables. Si las variables están cuantificadas, el valor de verdad depende del universo de discurso que se considere. Así por ejemplo, la sentencia

$$\exists x \in U \ (x^2 = 2)$$

es verdadera en el universo de discurso de los números reales ($U = \mathbb{R}$, siendo $x = \sqrt{2}$), pero es falsa en el universo de discurso de los números enteros ($U = \mathbb{Z}$ ya que $\sqrt{2} \notin \mathbb{Z}$).

Si el universo de discurso es finito, el cuantificador universal puede sustituirse por un número finito de conjunciones y el cuantificador existencial por un número finito de disyunciones.

Por ejemplo, si los valores posibles de la variable x son a, b, c, ($U = \{a, b, c\}$) la sentencia cerrada

$$\forall x \in U \ P(x)$$

es, por definición,

$$P(a) \wedge P(b) \wedge P(c)$$

y la sentencia

$$\exists x \in U \ P(x)$$

es, por definición,

$$P(a) \vee P(b) \vee P(c).$$

Estas definiciones muestran que los cuantificadores existencial y universal son una forma abreviada de escribir los conectivos $\vee$ y $\wedge$.

1.4.3 Forma de predicados

Como en el caso de la lógica proposicional en la lógica de predicados observamos que la validez o no de ciertos razonamientos está en su forma. Sea por ejemplo el razonamiento antes considerado, que intuitivamente se muestra como válido:

Todos los hombres son mortales
Sócrates es un hombre
Sócrates es mortal

Este razonamiento es el mismo, en su forma, que:

Todos los casos son resolubles
El caso Nécora es un caso
El caso Nécora es resoluble

Y se podría representar como:

$$\frac{\begin{array}{c}\forall x \in U\ P(x) \\ a \in U\end{array}}{P(a)}$$

Nuestra intuición señala que este razonamiento es válido siempre, en el sentido de que no depende del universo de discurso considerado. Resulta entonces natural escribir la siguiente expresión que consideraremos verdadera:

$$(\forall x\ P(x)) \Rightarrow P(a).$$

Obsérvese que no se ha escrito el universo de discurso.

De este modo es interesante construir una lógica formal usando predicados. Con este objeto vamos a dar un conjunto de reglas para definir *forma de predicados*, que es el equivalente, en la lógica de predicados, al concepto de *forma proposicional* en la lógica proposicional.

Paso 1. Comenzamos primero considerando funciones proposicionales, de la forma $P(x), Q(x), ...$ (donde también se pueden considerar varias variables). Las funciones proposicionales son *formas de predicados.*

Paso 2. Se pueden construir nuevas formas de predicados usando los conectivos lógicos. Formas por ejemplo, del tipo:

$$P(x) \wedge Q(x)$$
$$P(x) \wedge Q(y)$$
$$\neg P(x)$$
$$P(x) \vee Q(x)$$
$$P(x) \Rightarrow Q(x)$$
$$P(x) \Leftrightarrow Q(x)$$

Paso 3. Dada una forma de predicados de las construidas hasta ahora, donde aparece la variable x, digamos por simplicidad $P(x)$, se pueden construir las expresiones:

$$\forall x\ P(x)$$

$$\exists x\ P(x).$$

Por ejemplo se construyen las formas:

$$\forall x\ (P(x) \wedge Q(x))$$

$$\exists x\ (P(x, y) \Leftrightarrow Q(x)).$$

En las formas construidas en el paso 3, la variable x afectada por el cuantificador universal o existencial se dice que queda *ligada*. El resto de posibles variables se dicen *libres*. Volveremos más adelante sobre esta definición.

Paso 4. Dos formas de predicado se pueden combinar usando conectivos lógicos. Los paréntesis son necesarios para evitar ambigüedades. Estamos considerando, por ejemplo, expresiones del tipo:

$$(\forall x\ P(x)) \Rightarrow (\exists x\ Q(x)).$$

Paso 5. Dada una forma de predicados $\mathcal{P}(x)$ donde aparece una variable libre x, se pueden construir expresiones del tipo:

$$\forall x\ \mathcal{P}(x)$$

$$\exists x\ \mathcal{P}(x).$$

Por ejemplo expresiones del tipo:

$$\forall x(\forall y\ Q(x,y))$$
$$\forall x(P(x) \Rightarrow (\forall y\ Q(y)))$$

Paso 6. Dada una forma de predicado, cualquiera de sus variables libres puede sustituirse por una constante arbitraria, dando lugar a una nueva forma de predicado.

Usaremos los términos *forma de predicados*, *sentencia de lógica de predicados* o *expresión de lógica de predicados* indistintamente.

Ejemplo 1.4.1 *Una expresión de lógica de predicados es, por ejemplo:*

$$(\forall x\ P(x)) \implies \neg(\exists x\ Q(x)).$$

O el ejemplo que ilustraba el comienzo de la sección:

$$(\forall x\ P(x)) \Rightarrow P(a).$$

Observación 1.4.2 *Cuando en un predicado aparece una variable afectada por un cuantificador y simultáneamente aparece también sin ser afectada por* ***ese mismo*** *cuantificador, debe interpretarse que se está utilizando el mismo nombre de variable para dos variables distintas.*

EL PREDICADO	DEBE INTERPRETARSE COMO
$(\forall x\ P(x)) \vee (\forall x\ Q(x))$	$(\forall x\ P(x)) \vee (\forall y\ Q(y))$
$(\exists x\ P(x)) \vee (\exists x\ Q(x))$	$(\exists x\ P(x)) \vee (\exists y\ Q(y))$

Observación 1.4.3 *Dado un universo de discurso U, la expresión construida en la sección 1.4.2, a saber, $\forall x \in U P(x)$, no es una forma de predicado según las reglas que acabamos de describir. Para componer una forma de predicado con ese significado definimos un predicado de la forma $U(x) := (x \in U)$ y escribimos:*

$$\forall x\,(U(x) \Rightarrow P(x))$$

o lo que es lo mismo

$$\forall x\,((x \in U) \Rightarrow P(x)).$$

Recuperamos la definición establecida anteriormente:

Definición 1.4.4 *Diremos que una variable x* ***aparece ligada*** *en una forma de predicado P si está afectada por algún cuantificador. En caso de que una variable no aparezca ligada diremos que* ***aparece libre.***

Por ejemplo

En	la(s) variable(s) aparece(n)
$N(x)$	x aparece *libre*
$x > y$	x e y aparecen *libres*
$\exists y\ (y > x)$	x aparece *libre* e y *ligada*
$N(x) \Rightarrow \exists y\ (y > x)$	x aparece *libre* e y *ligada*
$\forall x\ (N(x) \Rightarrow \exists y\ (y > x))$	x e y aparecen *ligadas*

Para poder asignar valores de verdad a las formas de predicado hay que, como se decía anteriormente, cerrar las formas, esto es, sustituir las variables libres por constantes y entender cómo se asignan valores de verdad a las variables cuantificadas.

- Si en una forma no aparecen cuantificadores, al sustituir las variables por constantes, la asignación de valores de verdad se hace exactamente como en lógica de predicados.

Ejemplo 1.4.5 *Por ejemplo, tomemos la forma:*

$$(P(x) \wedge Q(x)) \Rightarrow R(y),$$

y constantes a y b. Al sustituir tenemos:

$$(P(a) \wedge Q(a)) \Rightarrow R(b),$$

que es una expresión de lógica proposicional, donde $P(a)$, $Q(a)$ y $R(b)$ son variables proposicionales, y como tal se trata, pudiéndose construir su tabla de verdad.

- Dada una forma de predicado del tipo $\forall x\ P(x)$, donde $P(x)$ es una forma de predicado sin cuantificadores y con una sola variable x, se dice que *la expresión $\forall x\ P(x)$ es verdadera* si al sustituir x por una constante a cualquiera en cualquier universo de discurso, se tiene que $P(a)$ es verdadero.

Ejemplo 1.4.6 *La sentencia $\forall x\ (P(x) \vee (\neg P(x)))$ es V. En efecto, ya que para cada constante a en cada universo de discurso U se tiene que $P(a) \vee \neg P(a)$ es verdadera.*

- Dada una forma de predicado del tipo $\exists x\ P(x)$, donde $P(x)$ es una forma de predicado sin cuantificadores y con una sola variable x, se dice que *la expresión $\exists x\ P(x)$ es verdadera* si al sustituir x por una constante a en algún universo de discurso, se tiene que $P(a)$ es verdadero. Es decir, el valor de verdad de $\exists x\ P(x)$ es el mismo de $\neg(\forall x\ \neg P(x))$.

Ejemplo 1.4.7 *La forma $\exists x\ (P(x) \Rightarrow P(a))$ (donde x una variable y a una constante) es verdadera pues tomando la constante a, al sustituir la variable libre x por a se obtiene la expresión $(P(a) \Rightarrow P(a))$ que es verdadera.*

- Usando los significados de los distintos conectivos lógicos podemos asignar valores de verdad a nuevas formas de predicado. Veamos un ejemplo.

Ejemplo 1.4.8 *La siguiente sentencia es verdadera:*

$$(\forall x\ P(x)) \Rightarrow (\exists y\ P(y)).$$

En efecto si $\forall x\ P(x)$ es falso la implicación es verdadera. Y si $\forall x\ P(x)$ es verdadera entonces para cada constante a en cada universo, $P(a)$ es verdadera, por lo que $\exists y\ P(y)$ es verdadera.

Para asignar valores de verdad a cualquier forma de predicado necesitamos el concepto de *tautología* en lógica de predicados, que no es exactamente igual que el que manejamos en lógica proposicional, y usar coherentemente los significados de los conectivos lógicos:

Definición 1.4.9 *Diremos que una forma de predicado que contenga variables libres es una* **tautología** *si toma únicamente el valor V independientemente del universo de discurso y los objetos concretos de dicho universo que asignemos a las variables que aparecen libres.*

Ejemplo 1.4.10 *Siendo $P(x)$ una función proposicional, el predicado $P(x) \vee (\neg P(x))$ es una tautología. Podemos argumentar ese hecho del siguiente modo: puesto que, dependiendo del objeto concreto que coloquemos en lugar de la x, $P(x)$ será V ó F, tenemos la tabla:*

$P(x)$
V
F

Utilizando ahora $P(x)$ junto con los conectivos lógicos $\vee$ y $\neg$ construimos $P(x) \vee (\neg P(x))$. Teniendo en cuenta el significado de dichos conectivos, podemos completar la tabla de verdad obteniendo:

$P(x)$	$\neg P(x)$	$P(x) \vee (\neg P(x))$
V	F	V
F	V	V

Ejercicio 11 *Demostrar que la expresión*

$$(P(x) \wedge (\neg P(x))) \Rightarrow Q(x, y)$$

es una tautología.

- Podemos ahora asignar un valor de verdad a la forma $\forall x\ P(x)$, donde $P(x)$ es una forma cualquiera y x es una variable libre, de la siguiente manera: la sentencia $\forall x\ P(x)$ es V si $P(x)$ es una tautología, y F en caso contrario.

Ejemplo 1.4.11 *La sentencia (recordar que x es una variable y a, b son constantes):*

$$(\forall y(\forall x\ P(x,y))) \Rightarrow P(a,b) \quad es\ V,\ puesto\ que:$$

a) Si $(\forall y(\forall x\ P(x,y)))$ es V, $P(x,y)$ es una tautología, lo que significa que al sustituir la variable x (respectivamente la y) por cualquier objeto concreto, digamos a (respectivamente b), del universo de discurso que se considere obtenemos que $P(a,b)$ verdadera, y por consiguiente

$$(\forall y(\forall x\ P(x,y))) \Rightarrow P(a,b)$$

es V.
b) Si por el contrario $(\forall y(\forall x\ P(x,y)))$ es F, la implicación

$$(\forall y(\forall x\ P(x,y))) \Rightarrow P(a,b) \quad es\ V.$$

- Para asignar un valor de verdad a la expresión $\exists x\ P(x)$, donde $P(x)$ es una forma cualquiera y x es una variable libre , consideramos lo siguiente: la sentencia $\exists x\ P(x)$ es V si $\neg P(x)$ no es una tautología, y F en caso contrario.

Es decir, $\exists x\ P(x)$ es V si hay un universo de discurso U y una constante $a \in U$ tal que $P(a)$ es V.

Se establece entonces la equivalencia lógica siguiente:

$$\exists x\ P(x)$$

es lógicamente equivalente (en el sentido de que tienen el mismo valor de verdad) a

$$\neg(\forall x\ \neg P(x)).$$

a) Si $\exists x\ P(x)$ es V, $\neg P(x)$ no es una tautología, o lo que es lo mismo, $\forall x\ \neg P(x)$ es F, con lo que $\neg(\forall x\ \neg P(x))$ es V.

b) Recíprocamente, si $\neg(\forall x\ \neg P(x))$ es V, necesariamente $(\forall x\ \neg P(x))$ es F, o lo que es lo mismo, $\neg P(x)$ no es una tautología. Entonces existe un objeto a de un universo de discurso tal que $\neg P(a)$ es F, con lo que $P(a)$ es V. Esto implica que la sentencia $\exists x\ P(x)$ es V.

Por consiguiente en cualquier universo de discurso, para todo predicado P y para toda variable x, la siguiente sentencia es verdadera:

$$\exists x\ P(x) \Leftrightarrow \neg(\forall x \neg P(x)).$$

Ejercicio 12 *Probar que en cualquier universo de discurso, para toda función $P(x)$, la siguientes sentencias son verdaderas:*

$$\neg(\forall x P(x)) \Leftrightarrow (\exists x \neg P(x))$$

$$\neg(\exists x P(x)) \Leftrightarrow (\forall x \neg P(x))$$

El ejercicio anterior indica cómo negar adecuadamente las sentencias afectadas por cuantificadores. La negación de la frase *todo hombre es mortal* no es *todo hombre es inmortal* sino que *existe un hombre que es inmortal.*

Observación 1.4.12 *Para probar que $(\forall x \in U\ P(x))$ es F basta encontrar una constante a en U de modo que $P(a)$ es F. A dicho a lo denominaremos* **contraejemplo**. *Por ejemplo, sea U el conjunto de los números naturales, $U = \mathbb{N}$, y el predicado $P(x)$:=x es primo o producto de exactamente dos números primos; la sentencia $(\forall x \in U\ P(x))$ es F puesto que 30 no es primo ni producto de exactamente dos primos. El número 30 es un contraejemplo a la sentencia $\forall x \in U\ P(x)$.*

1.4.4 Ejemplos de sentencias verdaderas. Contraejemplos

Veamos algunos ejemplos más:

Proposición 1.4.13 *En cualquier universo de discurso establecido y para cualquier predicado $P(x, y)$ la siguiente sentencia es V:*

$$(\exists x \forall y P(x, y)) \Rightarrow (\forall y \exists x P(x, y))$$

Demostración. Razonaremos por reducción al absurdo, vamos a suponer que la sentencia escrita es falsa y obtendremos una contradicción:

Supongamos que $(\exists x \forall y P(x, y)) \Rightarrow (\forall y \exists x P(x, y))$ es F. En ese caso se tiene que $(\exists x \forall y P(x, y))$ es V y $(\forall y \exists x P(x, y))$ es F.

De la primera condición se sigue que existe un objeto concreto a de un universo de discurso tal que $\forall y P(a,y)$ es V. Por tanto $P(a,y)$ es una tautología.

De la segunda condición se sigue que existe un objeto concreto b de un universo de discurso tal que $\exists x P(x,b)$ es F, o lo que es lo mismo, $\neg\left(\forall x\left(\neg P(x,b)\right)\right)$ es F. En consecuencia tendremos que $\forall x\left(\neg P(x,b)\right)$ es V con lo que $\neg P(x,b)$ también es una tautología. Ahora bien, por ser $P(a,y)$ una tautología $P(a,b)$ es V, y por ser $\neg P(x,b)$ una tautología, $\neg P(a,b)$ también es V, con lo que hemos obtenido la contradicción buscada.

Proposición 1.4.14 ***No*** *es cierto que para cualquier predicado $P(x,y)$ la siguiente sentencia es V independientemente del universo de discurso:*

$$(\forall x \exists y P(x,y)) \Rightarrow (\exists y \forall x P(x,y))$$

Demostración. Basta encontrar un ejemplo en el que la forma escrita no sea verdad, esto es, un **contraejemplo:**

Sea el predicado $P(x,y)$:=*x es estrictamente menor que y* en el universo de los números naturales. En este universo, con este predicado se tiene:

$$\forall x \in U\ \exists y \in U\ P(x,y):\ V$$

ya que para cada número natural x siempre existe un número natural y (por ejemplo $y = x+1$) estrictamente mayor. Pero:

$$\exists y \in U\ \forall x \in U\ P(x,y):\ F$$

ya que no existe un número natural mayor que el resto de los números naturales. Por tanto:

$$(\forall x \exists y P(x,y)) \Rightarrow (\exists y \forall x P(x,y))$$

es falsa en este ejemplo.

Proposición 1.4.15 *En cualquier universo de discurso y para cualesquiera que sean los predicados P y Q la siguiente sentencia es verdadera:*

$$\exists x\left(P(x) \vee Q(x)\right) \Leftrightarrow (\exists x P(x)) \vee (\exists x Q(x)).$$

Demostración. Como se trata de una dobre implicación tenemos que probar:

$$\text{i) } \exists x\,(P(x) \vee Q(x)) \Rightarrow (\exists x P(x)) \vee (\exists x Q(x)) \text{ es } V$$
$$\text{y que}$$
$$\text{ii) } (\exists x P(x)) \vee (\exists x Q(x)) \Rightarrow \exists x\,(P(x) \vee Q(x)) \text{ es } V$$

i) Supongamos verdadera la sentencia $\exists x\,(P(x) \vee Q(x))$. Sea a tal que

$$(P(a) \vee Q(a))$$

es verdadera. Existen dos posibilidades:

- Si $P(a)$ es V entonces $(\exists x P(x))$ es V y por tanto $(\exists x P(x)) \vee (\exists x Q(x))$ es V.
- Si $Q(a)$ es V se razona análogamente.

ii) Supongamos verdadera la sentencia $(\exists x P(x)) \vee (\exists x Q(x))$. En ese caso existen también dos posibilidades:

- Si $(\exists x P(x))$ es V entonces, siendo a un objeto tal que $P(a)$ es V, resulta que $P(a) \vee Q(a)$ es V y por tanto $\exists x\,(P(x) \vee Q(x))$ es V.
- Si $(\exists x Q(x))$ es V se razona análogamente.

Proposición 1.4.16 *En cualquier universo de discurso y para cualquier par de predicados $P(x)$ y $Q(x)$ la siguiente sentencia es verdadera*

$$((\forall x P(x)) \vee (\forall x Q(x))) \Rightarrow \forall x\,(P(x) \vee Q(x))\,.$$

Sin embargo ***no*** *es cierto que en cualquier universo de discurso y para cualquier par de predicados $P(x)$ y $Q(x)$ la siguiente sentencia sea verdadera:*

$$\forall x\,(P(x) \vee Q(x)) \Rightarrow ((\forall x P(x)) \vee (\forall x Q(x)))$$

Demostración. Para demostrar la primera parte razonamos por reducción al absurdo: supongamos que $((\forall x P(x)) \vee (\forall x Q(x)))$ es V y que $\forall x\,(P(x) \vee Q(x))$ es F. De la segunda condición se deduce que existe un objeto a tal que $(P(a) \vee Q(a))$ es F, con lo que $P(a)$ es F y $Q(a)$ es F. En consecuencia $(\forall x P(x))$ es F y $(\forall x Q(x))$ es F, es decir, $((\forall x P(x)) \vee (\forall x Q(x)))$ es F.

Para demostrar la segunda parte pondremos un contraejemplo: sea el universo de los números naturales y los predicados $P(x)$:=*x es par* y $Q(x)$:=*x es impar*. Sea la sentencia *todo número natural es o bien par o bien impar*, es decir $\forall x(P(x) \vee Q(x))$. Esta sentencia es V. Sin embargo la sentencia *todo número natural es par o todo número natural es impar*, esto es, $\forall x P(x) \vee \forall x Q(x)$ es F.

Proposición 1.4.17 *En cualquier universo de discurso y para cualquier predicado $P(x, y)$ la siguiente sentencia es verdadera*

$$\forall x \forall y P(x, y) \Leftrightarrow \forall y \forall x \ P(x, y)$$

Demostración. Demostramos primero la implicación de izquierda a derecha.

Razonamos por reducción al absurdo: supongamos que $\forall x \forall y P(x, y)$ es V y que $\forall y \forall x P(x, y)$ es F. De la segunda condición se sigue que existe b tal que $\forall x P(x, b)$ es F. Por ello existe a tal que $P(a, b)$ es F. Luego $\forall y P(a, y)$ es F y en consecuencia $\forall x \forall y P(x, y)$ es F.

Para la otra implicación, o bien se razona análogamente a la implicación que acabamos de probar, o bien se aplica el resultado anterior a la sentencia $\forall y \forall x \ P(x, y)$.

Ejercicio 13 *Demostrar el siguiente hecho usando una demostración por contrapositivo: Si $a \geq 0$ es un número real tal que para todo número real positivo, $\epsilon > 0$, se tiene $0 \leq a < \epsilon$, entonces $a = 0$.*

1.4.5 Modelización de expresiones en forma simbólica

Veamos algunos ejemplos de como se expresan en lenguaje formal frases del lenguaje natural.

La sentencia *Todos tenemos exactamente un alma gemela* se puede modelizar de la siguiente forma:

Sea $G(x, y)$ la sentencia *y es el alma gemela de x*. La sentencia que queremos modelizar dice que para cualquier persona x, existe otra y que es su alma gemela y que cualquier otra persona z no es el alma gemela de x:

$$\forall x \ \exists y \ (G(x, y) \wedge \forall z \ ((z \neq y) \Rightarrow \neg G(x, z)))$$

Observación 1.4.18 *Hay que tener cuidado con la negación de una sentencia cuando viene afectada por un cuantificador. Supongamos que queremos negar la sentencia*

Todos los alumnos de esta clase han aprobado algún examen en febrero.

Para ello simbolicemos por $C(x)$ el predicado *x es un alumno de esta clase*, por $A(x, y)$ el predicado *x ha aprobado el examen y* y por $F(y)$ *y se realizó en febrero.* Así, la sentencia dada se puede escribir

$$\forall x \ [C(x) \Rightarrow (\exists y \ (A(x, y) \wedge F(y)))] \, .$$

Su negación sería

$$\neg \forall x \ [C(x) \Rightarrow (\exists y \ (A(x, y) \wedge F(y)))]$$

que, según sabemos, es lógicamente equivalente a

$$\exists x \ \neg [C(x) \Rightarrow (\exists y \ (A(x, y) \wedge F(y)))]$$

que a su vez es lógicamente equivalente a

$$\exists x \ \neg [\neg C(x) \vee (\exists y \ (A(x, y) \wedge F(y)))]$$

que a su vez es lógicamente equivalente a

$$\exists x \ \ [\neg(\neg C(x)) \wedge \neg(\exists y A(x, y) \wedge F(y))]$$

que a su vez es lógicamente equivalente a

$$\exists x \ \ [C(x) \wedge (\forall y \ \neg(A(x, y) \wedge F(y))]$$

que a su vez es lógicamente equivalente a

$$\exists x \ \ [C(x) \wedge (\forall y \ (\neg A(x, y) \vee \neg F(y))] \, .$$

Es decir, la negación de la sentencia es

Existe algún alumno de esta clase que o no ha aprobado ningún examen $(\forall y \ (\neg A(x, y)))$ *o, si lo ha aprobado, no ha sido en febrero.*

Si preferimos quedarnos con la sentencia lógicamente equivalente a ésta de la penúltima línea se podría leer

existe algún alumno de esta clase que no ha aprobado ningún examen en febrero.

Ejercicio 14 *Escribir de forma simbólica las siguientes sentencias y su negación, de manera que en la expresión final no haya ningún cuantificador precedido del símbolo negación:*

1. *No todos los españoles son periodistas*
2. *Si algún caminante bosteza, todos los caminantes bostezan*
3. *Todo el mundo conoce a algún modisto*
4. *Entre dos números números reales distintos cualesquiera existe algún número racional*
5. *No existe un primo mayor que el resto de los números primos*

Ejercicio 15 *Sea x_n una sucesión de números reales. Sea $\mathbb{R}^+$ el conjunto de los números reales positivos y $\mathbb{N}$ el conjunto de los números naturales. Se dice que*

$\lim(x_n) = a$ *si* $\forall \varepsilon \in \mathbb{R}^+ \ (\exists n \in \mathbb{N}(\forall m \in \mathbb{N}(m > n \Rightarrow |x_m - a| < \varepsilon)))$.

Escribir una sentencia que indique formalmente que $\lim(x_n) \neq a$.

1.5 El razonamiento por inducción

En esta sección revisamos una de las técnicas más potentes que emplearemos con frecuencia para demostrar propiedades de objetos discretos (complejidad de algoritmos, teoremas sobre grafos, etc...)

El conjunto $\mathbb{N}$ de los números naturales es un conjunto caracterizado por, entre otras, la siguiente propiedad (volveremos sobre esto en el capítulo 3):

Si $A \subset \mathbb{N}$ *es tal que*

$1 \in A$ y

$\forall n \in \mathbb{N} \ (n \in A \Rightarrow (n+1) \in A)$,

entonces $A = \mathbb{N}$.

Esta propiedad es la base de lo que se conoce como **razonamiento por inducción:**

Proposición 1.5.1 *Sea $P(x)$ una propiedad de manera que*

1. *$P(1)$ es verdadera*

2. $\forall n \in \mathbb{N}$ *se verifica que* $(P(n) \Rightarrow P(n+1))$ *es* V

En estas circunstancias, la sentencia $(\forall n \in \mathbb{N}\ P(n))$ *es* V.

Demostración. Sea $A = \{n \in \mathbb{N} | P(n)\ es\ V\ \}$. La demostración consiste en ver que $A = \mathbb{N}$.

De las hipótesis se sigue que $1 \in A$. Para ver que $A = \mathbb{N}$ sólo queda por demostrar que si $n \in A$ entonces $(n+1) \in A$. Supongamos que $n \in A$; en ese caso $P(n)$ es V, y por la hipótesis 2 $(P(n) \Rightarrow P(n+1))$ es V, con lo que $P(n) \wedge (P(n) \Rightarrow P(n+1))$ es V, y puesto que la sentencia $p \wedge (p \Rightarrow q)$ implica lógicamente a la sentencia q, concluimos que $P(n+1)$ es V o, lo que es lo mismo, que $(n+1) \in A$. Esto concluye que $\mathbb{N} = A$.

El razonamiento por inducción se puede ver intuitivamente del siguiente modo: como $P(1)$ es V, y $P(1) \Rightarrow P(2)$ es V, ya que es un caso particular de $(P(n) \Rightarrow P(n+1))$, podemos concluir que $P(2)$ es V. Siendo $P(2)$ verdad, como $P(2) \Rightarrow P(3)$ es V, podemos concluir que $P(3)$ es V y así sucesivamente. Como lo que se pretende es probar que $P(n)$ es V para cada n natural es suficiente observar que, por muy grande que sea n, se puede repetir el argumento anterior tantas veces como sea necesario, pero siempre un número finito de veces (exactamente n), y concluir que $P(n)$ es V.

Observación 1.5.2 *Es usual, al hacer un razonamiento por inducción, emplear la siguiente jerga:*

- $P(1)$ *es* V *es la* ***base de inducción***
- $\forall n \in \mathbb{N}\ (P(n) \Rightarrow P(n+1))$ *es el* ***paso de inducción***
- *Normalmente se hace una demostración directa del paso de inducción. La suposición de que* $P(n)$ *es* V *para cierto* n *fijo es conocida como* ***hipótesis de inducción.***

Ejemplo 1.5.3 *Cualquier tablero cuadriculado de* $2^n \times 2^n$ *casillas iguales al que se le ha quitado una casilla puede cubrirse con piezas de tres casillas en forma de L sin que se solapen.*

Para demostrarlo razonamos por inducción sobre n:

Base de inducción: *P(1) es la proposición: Un tablero 2x2 al que se le ha quitado una casilla puede cubrirse con piezas de tres casillas en forma*

de L. Está claro que en este caso con una sola pieza basta y la sentencia es verdadera.

***Paso de inducción**: Supongamos que para tableros de $2^n \times 2^n$ casillas hay solución. Si ahora consideramos un tablero de tamaño $2^{n+1} \times 2^{n+1}$, podemos descomponerlo en cuatro tableros de tamaño $2^n \times 2^n$ dividiéndolo por la mitad longitudinal y transversalmente. Como en el tablero inicial falta una casilla, en una de las cuatro partes en que lo hemos dividido falta una casilla. En las otras tres partes quitamos la casilla de la esquina que se encuentra en el centro del tablero. De esta forma las tres casillas que quitamos forman una L. Ahora en cada una de las cuatro partes tenemos un tablero de tamaño $2^n \times 2^n$ al que le falta una casilla. Por tanto, por la hipótesis de inducción, podemos rellenar cada uno de ellos con piezas en forma de L por separado. Para acabar de rellenar el tablero original solo basta añadir una L en el hueco central formado por las tres casillas que hemos quitado.*

Ejercicio 16 *Demostrar, razonando por inducción, la validez de las siguientes fórmulas:*

$$\forall n \in \mathbb{N}, \quad \sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

$$\forall n \in \mathbb{N}, \quad \sum_{k=1}^{n} (2k-1) = n^2$$

$$\forall n \in \mathbb{N}, \quad 2^n \leq (n+1)!.$$

Ejercicio 17 *Demostrar por inducción que en efecto es una tautología la señalada en el ejercicio 9.*

1.5.1 Razonamiento por inducción completa

El razonamiento por inducción completa es equivalente al razonamiento por inducción, en el sentido de que cualquier demostración por inducción se puede realizar por inducción completa y recíprocamente, aunque hay casos en los que, por simplicidad, es conveniente emplear uno u otro método. Se trata de demostrar la validez de una sentencia de la forma

$$\forall n \in \mathbb{N}\ P(n).$$

El razonamiento por inducción completa consta de los siguientes pasos (observa que la única diferencia con el razonamiento por inducción está en el paso de inducción):

- **Base de inducción**: $P(1)$ es V.
- **Paso de inducción**: Se demuestra que $\forall n \in \mathbb{N}\ (P(1) \wedge ... \wedge P(n)) \Rightarrow P(n+1)$ es V.

Como en el caso del razonamiento por inducción la conclusión es que la sentencia $\forall n \in \mathbb{N}\ P(n)$ es V.

Ejemplo 1.5.4 *Vamos a demostrar utilizando el método de inducción completa, que todo número entero mayor que 1 es primo o producto de números primos:*

***Base de inducción**: P(2) es verdadera pues 2 es primo.*

Paso de inducción: *Supongamos que los números 2, ..., n son primos o producto de primos. Si n+1 es primo, hemos acabado. En caso contrario n+1 es compuesto, es decir n+1 = pq , con* $1 < p < n+1$ *y* $1 < q < n+1$. *Por hipótesis de inducción p y q son primos o producto de primos, y, en consecuencia n+1 también lo es .*

Obsérvese que en este caso la demostración por inducción en lugar de inducción completa es más complicada ya que no podríamos afirmar, por hipótesis de inducción, que p y q son primos o producto de primos. Esto sólo lo podríamos afirmar para n pues la hipótesis de inducción es $P(n)$ y no $P(1) \wedge ... \wedge P(n)$.

1.5.2 Inducción estructural

El método de inducción ordinaria se apoya en el hecho de que dado un número natural n, su sucesor $n+1$ es único. La inducción estructural es una generalización de la inducción ordinaria que tiene lugar cuando, teniendo en cuenta la estructura de los objetos que se están considerando, el paso de inducción de un elemento al siguiente puede producirse en más de una dirección.

Este método es una herramienta utilísima para el tratamiento de muchos objetos que aparecen en el ámbito de las ciencias de la computación, en

particular en la teoría de lenguajes formales y en la semántica de los lenguajes de programación.

El método consiste en definir en primer lugar una cierta construcción, como por ejemplo *un programa en Pascal* inductivamente, y demostrar alguna propiedad del programa por inducción sobre el número de aplicaciones de las reglas del programa.

Ejemplo 1.5.5 *Los números enteros vienen determinados por las siguientes reglas:*

$$\begin{aligned} 0 &\in \mathbb{Z}. \\ Si\ n &\in \mathbb{Z},\ entonces\ (n+1) \in \mathbb{Z} \wedge (n-1) \in \mathbb{Z}. \end{aligned}$$

En este caso el paso de inducción tiene dos direcciones, es decir, para probar el paso de inducción habría que comprobar que si n satisface la propiedad objeto de estudio, entonces también la satisfacen $(n+1)$ y $(n-1)$.

Veamos un ejemplo de una demostración por inducción estructural en este contexto.

Dado $a \in \mathbb{R}$, definimos a^n para $n \in \mathbb{Z}$ del siguiente modo:

$$\begin{aligned} a^0 &= 1 \\ a^{n+1} &= a^n \cdot a,\ a^{n-1} = \frac{a^n}{a}. \end{aligned}$$

En estas condiciones, para comprobar la ley

$$a^{m+n} = a^m \cdot a^n$$

fijamos m arbitrario y razonamos por inducción (estructural) en n:

Base de inducción: *para $n = 0$,*

$$a^{m+0} = a^m = a^m \cdot 1 = a^m \cdot a^0.$$

***Paso de inducción**: Supongamos válida la ley para n. Tenemos que demostrar que en ese caso la ley también es válida para $(n+1)$ y para $(n-1)$:*

$$a^{m+(n+1)} = a^{(m+n)+1} = a^{m+n} \cdot a =$$

$$= (a^m \cdot a^n) \cdot a = a^m \cdot (a^n \cdot a) = a^m \cdot a^{n+1},$$

donde la igualdad tercera es la hipótesis de inducción. Y también:

$$a^{m+(n-1)} = a^{(m+n)-1} = \frac{a^{m+n}}{a} =$$

$$= (\frac{a^m \cdot a^n}{a}) = a^m \cdot (\frac{a^n}{a}) = a^m \cdot a^{n-1},$$

donde de nuevo la igualdad tercera es la hipótesis de inducción. Con esto concluimos, como consecuencia del razonamiento por inducción estructural efectuado, que la ley es válida $\forall n \in \mathbb{Z}$.

Ejemplo 1.5.6 *Dada una forma proposicional* $\mathcal{A}$*, definimos la forma proposicional* **complementaria** *de* $\mathcal{A}$ *del siguiente modo:*

$$\begin{aligned} \textit{Si } p \textit{ es una variable proposicional, } comp(p) &= \neg p,\ comp(\neg p) = p, \\ comp(\mathcal{A} \vee \mathcal{B}) &= comp(\mathcal{A}) \wedge comp(\mathcal{B}), \\ comp(\mathcal{A} \wedge \mathcal{B}) &= comp(\mathcal{A}) \vee comp(\mathcal{B}). \end{aligned}$$

Teorema: *Para toda forma proposicional* C *se verifica:*

$$comp(C) = \neg C.$$

Demostración. Razonamos por inducción estructural:

Base de inducción: *Los únicos casos (casos básicos) para los que se puede aplicar directamente la definición de forma proposicional complementaria son: si* $C = p$,

$$comp(C) = comp(p) = \neg p = \neg C$$

y si $C = \neg p$

$$comp(C) = comp(\neg p) = p = \neg\neg p = \neg C.$$

Antes de hacer el paso de inducción recordar que $\{\wedge, \vee, \neg\}$ *es un conjunto adecuado de conectivos por lo que toda forma proposicional se puede construir mediante las reglas de construcción de formas proposicionales y sólo usando estos conectivos.*

Paso de inducción. *Hipótesis de inducción: Supongamos que cualquie forma proposicional con n conectivos lógicos del tipo* $\wedge$ *o* $\vee$ *satisface el teorema. Sea* C *una forma proposicional con n+1conectivos del tipo* $\wedge$ *o* $\vee$ *.*

Si $C = \mathcal{A} \vee \mathcal{B}$, *tendremos que*

$$\begin{aligned} comp(\mathcal{A} \vee \mathcal{B}) &= comp(\mathcal{A}) \wedge comp(\mathcal{B}) = \\ &= \neg\mathcal{A} \wedge \neg\mathcal{B} = (De\ Morgan) = \\ &= \neg(\mathcal{A} \vee \mathcal{B}) = \neg C. \end{aligned}$$

Donde la segunda igualdad es verdadera por hipótesis de inducción.

Si $C = \mathcal{A} \wedge \mathcal{B}$, *tendremos que*

$$\begin{aligned} comp(\mathcal{A} \wedge \mathcal{B}) &= comp(\mathcal{A}) \vee comp(\mathcal{B}) = \\ &= \neg\mathcal{A} \vee \neg\mathcal{B} = (De\ Morgan) = \\ &= \neg(\mathcal{A} \wedge \mathcal{B}) = \neg C. \end{aligned}$$

Donde la segunda igualdad es verdadera por hipótesis de inducción.

Si $C = \neg\mathcal{A}$ *entonces por las leyes de De Morgan* $\neg\mathcal{A}$ *admite una forma equivalente, que denotamos igual, con el mismo número de conectivos del tipo* $\wedge$ *o* $\vee$. *De este modo, por hipótesis de inducción:*

$$comp(C) = comp(\neg\mathcal{A}) = \neg(\neg\mathcal{A}) = \mathcal{A} = \neg C.$$

Por consiguiente, concluimos, como queríamos demostrar, que para cualquier forma proposicional se tiene:

$$comp(C) = \neg C.$$

1.6 Aplicaciones

1.6.1 Corrección de algoritmos y verificación de programas

Como estudiaremos con más detalle en el siguiente capítulo, un **algoritmo** es una *rutina* o procedimiento mecánico que acepta un cierto conjunto de *inputs*, para cada uno de los cuales obtiene un *output* tras un número finito de pasos. Por un **programa** entenderemos una expresión o una secuencia finita de expresiones en algún lenguaje de programación. El concepto de programa está relacionado con el concepto de algoritmo, pero no es exactamente igual, pues un programa es la realización concreta de un algoritmo, pero un

algoritmo puede ser programado (o implementado) en diferentes lenguajes de programación. En el siguiente capítulo estudiaremos el concepto de eficiencia relacionado con los algoritmos (tiempo de ejecución y necesidades de memoria). En esta sección nos ocuparemos de la siguiente pregunta: ¿Cuándo podemos asegurar que un algoritmo (o el programa que lo implementa) es correcto (i.e., que hace aquello para lo que fue diseñado)? Probar que un algoritmo es correcto requiere métodos de demostración similares a los estudiados en el desarrollo del capítulo. En cualquier caso, verificar que un algoritmo es correcto puede ser una tarea muy difícil. Sólo algoritmos relativamente simples pueden ser verificados utilizando las técnicas que vamos a describir a continuación. En cualquier caso, la comprensión de estas técnicas es seguro que servirá de ayuda para diseñar algoritmos de una manera mejor.

Por otra parte es imposible diseñar un método que nos permita verificar la corrección de cualquier programa. Esto es una consecuencia del teorema de incompletitud de Gödel sobre los límites del razonamiento formal. No obstante, este resultado nos permite garantizar que no es posible desarrollar una maquinaria general de verificación de la corrección de programas, pero el tipo de programas que aparecen en la práctica no son, en cualquier caso, totalmente generales, así que podemos desarrollar métodos de verificación en algunos casos importantes. Más aún, al hacerlo estaremos mejorando nuestra metodología de programación. El ideal sería diseñar programas junto con las demostraciones de su corrección.

La verificación de programas está muy relacionada con lo que se denomina *Demostración automática de teoremas*. De hecho, para ser riguroso con el concepto de verificación, necesitaríamos una descripción precisa de aquello que se pretende que el programa haga, y esto debe expresarse en un lenguaje diseñado para este propósito. A este tipo de lenguajes se les denomina **lenguajes de especificación**.

En cualquier caso el objetivo de este apartado es remarcar el papel que juega la inducción en las iteraciones y recursiones que se emplean en muchos lenguajes de programación, para lo que incluimos algunos ejemplos para ilustrar las ideas.

Ejemplo 1.6.1 *Consideremos el siguiente fragmento de programa, diseñado para calcular el producto de dos números naturales a y b utilizando sólo la suma, la multiplicación por 2 y la división por 2:*

$r := 0$ $m := a$ $n := b$

while $m > 0$
    if m es impar then $r := r + n$
    $m := m/2$
    $n := 2n$

En las sentencias anteriores, $m/2$ representa el número parte entera del cociente de m entre 2. La intención es que si a y b son los valores iniciales de las variables m y n, entonces el programa debería terminar con la variable r con el valor ab. Por ejemplo, para $a = 11$ y $b = 26$, el programa haría lo siguiente:

m	n	r
11	26	0
5	52	26
2	104	78
1	208	78
0	416	286

es decir, el output es 286=11·26.

Este programa particular tiene una estructura muy simple: después de asignar el valor 0 a la variable r, ejecuta la iteración *while* hasta que la variable m toma el valor 0. Probaremos por inducción sobre k que, después de k ejecuciones de la iteración *while*, los valores que alcanzan m, n y r satisfacen la igualdad:

$$m \cdot n + r = ab$$

Base de inducción: si $k = 0$, es decir, si estamos justo antes de que la iteración while sea ejecutada por primera vez, los valores de m, n y r son, respectivamente, a, b y 0, con lo que el resultado es inmediato.

Paso de inducción: Supongamos el resultado para k, estrictamente menor que el número de veces que se repite el bucle. Sean a_1, b_1 y c_1 los valores de m, n y r antes de la iteración $k+1$, y a_2, b_2 y c_2 los valores de m, n y r después de dicha iteración. Por hipótesis de inducción

$$a_1 \cdot b_1 + c_1 = ab$$

y lo que tenemos que demostrar es que

$$a_2 \cdot b_2 + c_2 = ab.$$

Distingamos dos casos:

a) si a_1 es par, entonces $a_2 = \frac{a_1}{2}$, $b_2 = 2b_1$ y $c_2 = c_1$, por lo que

$$a_2 \cdot b_2 + c_2 = \frac{a_1}{2} \cdot 2b_1 + c_1 = ab.$$

b) si a_1 es impar, entonces $a_2 = \frac{(a_1-1)}{2}$, $b_2 = 2b_1$ y $c_2 = c_1 + b_1$, por lo que

$$a_2 \cdot b_2 + c_2 = \frac{(a_1 - 1)}{2} \cdot 2b_1 + c_1 + b_1 = a_1 b_1 - b_1 + c_1 + b_1 = ab$$

con lo que la demostración por inducción está completa.

Una vez realizada esta demostración podemos asegurar que el programa (en este caso el fragmento de programa) es **correcto parcialmente** ya que **si el programa termina,** entonces da la respuesta correcta. El otro ingrediente de la demostración de corrección es la **terminación**. Para que el programa sea correcto, tenemos que demostrar que el programa siempre termina.

Examinando la clausula *while*, vemos que el programa termina sólo cuando m toma el valor 0, y por otro lado en cada iteración de la clausula *while* el valor de m decrece. Esto significa que, como máximo en la repetición m-ésima, tomará el valor 0, por lo que podemos asegurar que el programa es correcto.

1.6.2 Ingeniería del conocimiento: sistemas expertos

Un *sistema experto* es un sistema informático diseñado para resolver problemas en un área específica, y al que de algún modo se le ha dotado de una competencia similar a la de un experto humano de ese área. Así por ejemplo, MYCIN es un sistema experto en diagnóstico y tratamiento de un número muy reducido de enfermedades infecciosas de la sangre, y PROSPECTOR es un sistema experto en determinar la probabilidad de la existencia de yacimientos de ciertos minerales a partir de las pruebas realizadas sobre el terreno.

Según el caso, el objetivo de un sistema experto puede ser el de sustituir al experto humano o el de ayudar a los expertos humanos a tratar con volúmenes de información que desbordan su capacidad. En cualquier caso el objetivo final es el mismo de todas las aplicaciones informáticas: relevar al hombre de tareas mecánicas y proporcionarle instrumentos amplificadores de

sus capacidades mentales. Los sistemas expertos se construyen siguiendo una concepción modular. Por una parte se construye una *base de conocimientos* o base de hechos y por otra se desarrollan los procedimientos que permiten manipular esa base para obtener una respuesta con datos concretos *(motor de inferencia)*. Los problemas que surgen en relación con estos dos apartados han dado lugar a un área de trabajo que se conoce como *ingeniería del conocimiento*.

En este apartado veremos cómo se utilizan las expresiones lógicas para representar el conocimiento y, en particular y para no complicar mucho la exposición, las sentencias de la lógica proposicional.

Un sistema de producción es un modelo de computación que distingue tres componentes: una base de hechos, una base de conocimientos y un motor de inferencias. La base de comocimientos no es necesariamente una base de datos en el sentido informático habitual: según el sistema puede ser desde una tabla de números hasta una base de datos en sentido propio.

Las reglas de producción se aplican sobre la base de hechos, cambiando su estado y el sistema de control gobierna estos procedimientos y aplicaciones y hace que la computación se detenga cuando el estado de la base de datos satisface alguna condición de terminación predefinida.

La base de hechos contendrá los hechos iniciales y los que se vayan obteniendo como consecuencias en el proceso inferencial. Las *reglas de producción* son pares ordenados (A,B). Según el tipo de sistema se denominan *antecedente* y *consecuente*, *condición* y *acción* o *premisa* y *conclusión*. Su formalización lógica es la de las sentencias condicionales A⇒B. Por ejemplo, en la base de conocimientos del sistema XCON hay unas 2500 reglas. Una de ellas es:

> *Si el contexto actual es el de asignar una fuente de alimentación,*
> *y se ha colocado un módulo SBI en un armario,*
> *y se conoce la posición que ocupa el módulo,*
> *y se dispone de una fuente de alimentación,*
> *entonces colocar la fuente en dicha posición.*

Es claro que, independientemente del significado de esas expresiones en el contexto de dicho sistema, la regla se puede formalizar mediante la sentencia:

$$(p_1 \wedge p_2 \wedge p_3 \wedge p_4) \Rightarrow q.$$

En el contexto de un sistema experto en medicina, podríamos considerar las reglas:

R1: *Si el paciente tiene fiebre,*

y tose,

y tiene dolores musculares,

entonces padece gripe.

R2: *Si el paciente padece gripe o resfriado,*

y no tiene úlcera,

entonces recomendar aspirina y coñac.

La formalización de tales reglas sería:

$$\begin{aligned} R1 &: [f \wedge t \wedge m] \Rightarrow g \\ R2 &: [(g \vee r) \wedge (\neg u)] \Rightarrow (a \wedge c) \end{aligned}$$

Utilizando las equivalencias lógicas vistas, la regla $R2$ es lógicamente equivalente a

$$[(g \wedge (\neg u)) \Rightarrow a] \wedge [(g \wedge (\neg u)) \Rightarrow c] \wedge [(r \wedge (\neg u)) \Rightarrow a] \wedge [(r \wedge (\neg u)) \Rightarrow c]$$

y en consecuencia se podría descomponer en las siguientes:

$$\begin{aligned} R2a &: (g \wedge (\neg u)) \Rightarrow a \\ R2b &: (g \wedge (\neg u)) \Rightarrow c \\ R2c &: (r \wedge (\neg u)) \Rightarrow a \\ R2d &: (r \wedge (\neg u)) \Rightarrow c. \end{aligned}$$

Ante una situación concreta en la que se hace una consulta al sistema, se tiene la evidencia de que un subconjunto de esos hechos son verdaderos y se trata de encontrar qué otros hechos pueden inferirse de esa certidumbre y de las reglas.

Si por ejemplo hiciésemos una consulta al sistema sobre un paciente que satisface los hechos f, t, m y $\neg u$, el sistema trataría de aplicar las reglas para ver qué terapia habría que aplicar. Así pues, sabiendo que tenemos los hechos f, t, m y $\neg u$, podemos hacer las siguientes inferencias, usando la regla de inferencia *modus ponens*:

$$\frac{\begin{array}{c} f \wedge t \wedge m \\ R1 : [f \wedge t \wedge m] \Rightarrow g \end{array}}{g}$$

(en este momento hemos ampliado la base de hechos con el nuevo hecho g)

$$\frac{\begin{array}{c} g \wedge (\neg u) \\ R2a : (g \wedge (\neg u)) \Rightarrow a \end{array}}{a}$$

(en este momento hemos ampliado la base de hechos con el nuevo hecho a)

$$\frac{\begin{array}{c} g \wedge (\neg u) \\ R2b : (g \wedge (\neg u)) \Rightarrow c \end{array}}{c} \ ,$$

es decir, la terapia es aspirina y coñac.

El problema que surge es que en general no tendremos dos, sino muchas reglas (en los sistemas expertos es frecuente que sean del orden de cientos o de miles), y en general hay que establecer en qué orden y de qué manera aplicamos las reglas (es decir, cuál o cuáles se aplican primero) para establecer el procedimiento de inferencia.

Para esto, hay dos estrategias básicas:

1. Ir aplicando cuantas reglas de producción y cuantas reglas de inferencia se puedan para ir sucesivamente ampliando la base de hechos, que es lo que hemos hecho en el ejemplo y que se conoce como *encadenamiento hacia adelante.*

2. Fijarse un hecho como objetivo y tratar de deducirlo, viendo de qué reglas de producción es consecuente. Este es el principio de *encadenamiento hacia atrás.*

En otras palabras, el encadenamiento hacia adelante consistiría en ir recorriendo las reglas desde la primera hasta llegar a una que pueda aplicarse (de acuerdo con la ley de inferencia modus ponens), ampliar la base de hechos con la nueva consecuencia, empezar de nuevo con la primera regla, y así sucesivamente hasta incluir el objetivo en la base de hechos, o hasta que ya no puedan aplicarse reglas.

Cuando no se trata de derivar cuantas conclusiones se puedan, sino de inferir un objetivo, o verificar que un hecho concreto es consecuencia de otros hechos y de las reglas de producción, lo que procede es aplicar el encadenamiento hacia atrás, para lo cual, si por ejemplo intentamos ver si puede

deducirse un hecho x, buscaríamos en primer lugar las reglas en las que figura como consecuente, y a partir de ellas iríamos realizando el estudio hacia atrás.

Un lenguaje gráfico muy adecuado es el de los árboles de decisión que estudiaremos en su momento.

1.7 Ejercicios

Ejercicio 18. Escribe la tabla de verdad de las siguientes proposiciones:

1) $((p \vee \neg q) \wedge m) \Rightarrow m$
2) $(\neg q \vee (p \Rightarrow q)) \Rightarrow \neg p$
3) $(p \Rightarrow (m \vee n)) \Rightarrow m$
4) $(p \Rightarrow (m \wedge n)) \Rightarrow m$

Ejercicio 19. Determina si el siguiente razonamiento es correcto: todo estudiante de esta clase mide menos estrictamente que 1.70 o más de 1.70. Pedro mide más de 1.70 luego es de esta clase.

Ejercicio 20. Sea X el conjunto de países e Y el conjunto de deportes olímpicos. La sentencia $M(x, y)$ significa que el país $x \in X$ ha ganando una medalla en el deporte $y \in Y$. Escribe las siguientes frases utilizando cuantificadores, evitando la aparición de un signo de negación delante de un cuantificador:

Ningún país ha ganado todas las medallas.
Todos los países han ganado alguna medalla.
Algún país ha ganado alguna medalla.
Algún país no ha ganado ninguna medalla.

Ejercicio 21. Sea $\mathbb{N}$ el conjunto de los números naturales y $P(a, b)$ la sentencia a divide a b (esto es, el resto de dividir b entre a es 0). Determina el valor de verdad de las siguientes proposiciones:

$P(2, 3)$
$P(5, 10)$
$P(2, 3) \wedge P(5, 10)$
$P(2, 3) \vee P(5, 10)$
$P(2, 3) \Rightarrow P(5, 10)$
$\forall m \in \mathbb{N} \forall n \in \mathbb{N}\ P(m, n)$
$\exists m \in \mathbb{N} \forall n \in \mathbb{N}\ P(m, n)$

$\exists n \in \mathbb{N} \forall m \in \mathbb{N}\ P(m,n)$
$\forall n \in \mathbb{N}\ P(1,n)$
$\forall m \in \mathbb{N}\ P(m,1)$

Ejercicio 22. Escribe formalmente las siguientes frases y su negación en lenguaje formal y en lenguaje natural:

Si vale menos de 1000 pts, comeré en la cafetería.
Todos los habitantes de Madrid viajan en metro.
Ningún habitante de Móstoles coge el autobús.
Hay personas en todas las ciudades que usan el transporte público.

Ejercicio 23. Demuestra por inducción las siguientes afirmaciones:

$1^2 + 2^2 + ... + n^2 = n(n+1)(2n+1)/6$.
$2^n > n+1$ para cada entero mayor o igual que 2.
$x^n - y^n$ es divisible por $x - y$ para todo natural n.

Ejercicio 24. Poner un ejemplo distinto al de las notas de una sentencia tal que siendo la afirmación $\forall x \exists y P(x,y)$ cierta no lo sea la afirmación $\exists x \forall y P(x,y)$.

Ejercicio 25. Demuestra por casos la propiedad del valor absoluto por la cual:

$$|xy| = |x||y|$$

Ejercicio 26. Demuestra poniendo un contraejemplo que las siguientes afirmaciones no son verdaderas:

Todo entero mayor que 17 es el cuadrado de un número entero.
Todo entero mayor que 6 es múltiplo de 2 ó de 3.
$100n + 1 > n^2$ para todo entero n.

Ejercicio 27. Demuestra por reducción al absurdo las siguientes afirmaciones:

$\sqrt{2}$ no es racional,
si un número real x es racional entonces $\pi + x$ no es racional.

Ejercicio 28. Escribe en lenguaje formal los siguientes razonamientos demostrando si son correctos:

Si salgo a la calle me visto. Salgo a la calle, por tanto voy vestido.

La comida incluye el primer plato y el segundo, por tanto incluye el primer plato.

Venda es una palabra de género femenino, por tanto venda es femenino y singular.

Si vale menos de 1000 pts. tengo dinero suficiente. Si tengo dinero suficiente lo compraré. Vale menos de 1000 por tanto lo compro.

1.8 Ejercicios resueltos

Los ejercicios 1, 2, 3, 4 y 18 se presentan resueltos con Maple.

1.8.1 Lógica con Maple

El programa Maple V permite trabajar algunos aspectos de lógica, que se cargan con el paquete logic.

```
> restart; with(logic);
```

Nos interesan en este momento algunas de estas funciones:

bequal (B1,B2) - comprobación de la equivalencia lógica de las expresiones **B1** y **B2**.

tautology(B) - comprobación si la expresion **B** es una tautología.

convert/toinert - para poder evaluar.

bsimp(A)- simplifica una expresión.

Para obtener más información sobre facilidades lógicas de Maple se puede consultar la ayuda.

```
> ?logic;
```

Conectivos lógicos

Maple sólo tiene los conectivos lógicos **and, or, not.** Son suficientes por las equivalencias lógicas de la implicación y la doble implicación con sentencias lógicas sólo hechas con and, or y not. Para evaluar las funciones hay que poner delante el símbolo & con que se representa la conjunción y.

```
> p and q;
```

$$p \textbf{ and } q$$

```
> p or q;
```

$$p \textbf{ or } q$$

```
> not p;
```

$$\textbf{not } p$$

Usemos la función **bequal** para comprobar que **la negación de (p or q)** es equivalente a **(not p and not q)**.

```
> bequal(&not(p &or q), (&not p) &and (&not
> q));
```

true

Aunque el conectivo implies no está, sí hay una función **implies**. La implicación **p implies q** es equivalente a **not p or q**.

```
> bequal(p &implies q, &not p &or q);
```

true

Podemos hacer un procedimiento para tener el conectivo implicación (**impl**) y la doble implicación (la vamos a llamar **dimpl**). Recordemos que **p dimpl q** es equivalente a **(p impl q) and (q impl p)**.

```
> impl:=proc(a,b)
> local c:
> c:=(not a or b):
> c; end:
> dimpl:=proc(a,b) local c:
> c:=(not a or b)
> and (not b or a):
> c; end:
> impl(p,q); dimpl(p,q);
```

$$\textbf{not } p \textbf{ or } q$$

$$(\textbf{ not } p \textbf{ or } q) \textbf{ and } (\textbf{ not } q \textbf{ or } p)$$

Ejemplos de uso

Veamos algunos ejemplos donde se comprueba que ciertas expresiones son tautologías. La función **convert/toinert** pone los símbolos & para que se pueda evaluar:

Ejercicio 2.
Escribimos las formas proposicionales:

```
> ej2_1:=impl(p and impl(p,q),q):
> ej2_2:=impl(impl(p,q) and
> impl(q,r),impl(p,r)):
> ej2_3:=impl(impl(p,q) and impl(not
> p,q),q):
> ej2_4:=impl((p or q) and (not p),q):
> ej2_5:=impl(p and not p, q):
```

Comprobamos que son tautologías:

```
> tautology(convert(ej2_1,toinert));
```

true

```
> tautology(convert(ej2_2,toinert));
```

true

```
> tautology(convert(ej2_3,toinert));
```

true

```
> tautology(convert(ej2_4,toinert));
```

true

```
> tautology(convert(ej2_5,toinert));
```

true

Ejercicio 3.
1. La equivalencia entre la doble implicación y la conjunción de las dos implicaciones (en realidad esto sólo comprueba que hemos hecho las cosas bien pues así hemos definido la doble implicación al construir dimpl):

```
> A:=impl(p,q) and impl(q,p);
```

$$A := (\ \textbf{not}\ p\ \textbf{or}\ q)\ \textbf{and}\ (\ \textbf{not}\ q\ \textbf{or}\ p)$$

```
> A1:=dimpl(p,q);
```

$$A1 := (\ \textbf{not}\ p\ \textbf{or}\ q)\ \textbf{and}\ (\ \textbf{not}\ q\ \textbf{or}\ p)$$

```
> bequal(convert(A,toinert),convert(A1,toinert));
```

$$true$$

2. La equivalencia entre una sentencia y su contrarrecíproca:

```
> A:=impl(p,q);
```

$$A := \textbf{not}\ p\ \textbf{or}\ q$$

```
> B:=impl(not q, not p);
```

$$B := q\ \textbf{or}\ \ \textbf{not}\ p$$

```
> bequal(convert(A,toinert),convert(B,toinert));
```

$$true$$

3. Otra forma de ver la implicación: **p implica q** es equivalente a que **p and not q implica not p:**

```
> B:=dimpl(impl(p,q),impl(p and not q,not
> p));
```

$$B := \textbf{not}\ ((\ \textbf{not}\ p\ \textbf{or}\ q)\ \textbf{and}\ p\ \textbf{and}\ \ \textbf{not}\ q\ \textbf{and}\ p)\ \textbf{and}\ (p\ \textbf{and}\ \ \textbf{not}\ q\ \textbf{and}\ p\ \textbf{or}\ \ \textbf{not}\ p\ \textbf{or}\ q)$$

```
> B1:=convert(B,toinert):
> tautology(B1); bequal(B1,true);
```

$$true$$

$$true$$

4. Reducción al absurdo:

```
> C:=dimpl(p,impl(not p,p));
```

$$C := true$$

```
> tautology(convert(C,toinert));
```

$$true$$

5. Reducción al absurdo:

```
> E:=dimpl(p,impl(not p,q and not q));
```

$$E := true$$

Ejercicio 4.
Doble negación:

```
> dimpl(not(not q),q);
```

$$true$$

And es asociativo:

```
> Andasociativo:=dimpl((p and q) and r,p
> and (q
> and r)):
> tautology(convert(Andasociativo,toinert));
```

$$true$$

Or es asociativo:

```
> Orasociativo:=dimpl((p or q) or r,p or (q
> or
> r)):
> tautology(convert(Orasociativo,toinert));
```

$$true$$

And es conmutativo:

```
> Andconmuta:=dimpl((p and q),(q and p)):
> tautology(convert(Andconmuta,toinert));
```

$$true$$

Distributivas:

```
> dist1:=dimpl(p and (q or r),(p and q) or
> (p
> and r)):
> dist2:=dimpl(p or (q and r),(p or q) and
> (p
> or r)):
> tautology(convert(dist1,toinert));
```

$$true$$

```
> tautology(convert(dist2,toinert));
```

true

Leyes de de Morgan:

```
> dimpl(not(r and q),not r or not q);
```

true

```
> dimpl(not(r or q),not r and not q);
```

true

Las restantes son obvias.

Ejercicio 1. Para hacer tablas de verdad hay que construir un pequeño procedimiento en Maple. Se construyen para una, dos y tres variables y los nombres de las variables los da el nombre del procedimiento.

```
> tablap:=proc(B)
> local A, L, c, i, j:
> L:=[true,false]:
> for i from 1 to 2 do
> c:=bsimp(subs(p=L[i],q=L[i],B));
> print(p=L[i],q=L[i],A=c);
> od:
> end:
```

```
> tablapq:=proc(B)
> local A, L, c, i, j:
> L:=[true,false]:
> for i from 1 to 2 do
> for j from 1 to 2 do
> c:=bsimp(subs(p=L[i],q=L[j],B));
> print(p=L[i],q=L[j],A=c);
> od:
> od:  end:
```

```
> tablapqr:=proc(B)
> local A, L, c, i, j, k:
> L:=[true,false]:
> for i from 1 to 2 do
> for j from 1 to 2 do
> for k from 1 to 2 do
> c:=bsimp(subs(p=L[i],q=L[j],r=L[k],B));
> print(p=L[i],q=L[j],r=L[k],A=c);
> od:
> od:
> od:  end:
```

Estos procedimientos se pueden usar siempre que haya una, dos o tres variables y éstas se llamen p, q y r. Si no se llaman así las cambiamos el nombre. Si queremos añadir más variables no hay más que introducir otro bucle.

Las cinco proposiciones que se presentan son:

```
> ej1:=not(p or q):
> ej2:=p and (not p):
> ej3:=not(p and (not p)):
> ej4:=impl(p,q and not r):
> ej5:=dimpl(not(p and q),(not p) or (not
> q)):
```

Sus tablas de verdad son:

```
> tablapq(ej1);
```

$$p = true,\ q = true,\ A = false$$

$$p = true,\ q = false,\ A = false$$

$$p = false,\ q = true,\ A = false$$

$$p = false,\ q = false,\ A = true$$

```
> tablapq(ej2);
```

$$p = true,\ q = true,\ A = false$$

$$p = true,\ q = false,\ A = false$$

$$p = false,\ q = true,\ A = false$$

$$p = false,\ q = false,\ A = false$$

```
> tablap(ej3);
```

$$p = true,\ q = true,\ A = true$$

$$p = false,\ q = false,\ A = true$$

```
> tablapqr(ej4);
```

$$p = true,\ q = true,\ r = true,\ A = false$$

$$p = true,\ q = true,\ r = false,\ A = true$$

$$p = true,\ q = false,\ r = true,\ A = false$$

$$p = true,\ q = false,\ r = false,\ A = false$$

$$p = false,\ q = true,\ r = true,\ A = true$$

$$p = false,\ q = true,\ r = false,\ A = true$$

$$p = false,\ q = false,\ r = true,\ A = true$$

$$p = false,\ q = false,\ r = false,\ A = true$$

```
> tablapq(ej5);
```

$$p = true,\ q = true,\ A = true$$

$$p = true,\ q = false,\ A = true$$

$$p = false,\ q = true,\ A = true$$

$$p = false,\ q = false,\ A = true$$

Ejercicio 18.

En el apartado 1 las variables proposicionales se llaman p, q y m. Como en nuestro procedimiento se llaman p,q y r, cambiamos el nombre de m por r.

```
> ej18_1:=impl((p or not q) and m,m);
```

$$ej18\_1 := \textbf{not}\ ((p\ \textbf{or}\ \ \textbf{not}\ q)\ \textbf{and}\ m)\ \textbf{or}\ m$$

```
> ej18_1:=subs(m=r,ej18_1);
```

$$ej18\_1 := \textbf{not}\ ((p\ \textbf{or}\ \ \textbf{not}\ q)\ \textbf{and}\ r)\ \textbf{or}\ r$$

```
> tablapqr(ej18_1);
```

$$p = true,\ q = true,\ r = true,\ A = true$$

$$p = true,\ q = true,\ r = false,\ A = true$$

$$p = true,\ q = false,\ r = true,\ A = true$$

$$p = true,\ q = false,\ r = false,\ A = true$$

$$p = false,\ q = true,\ r = true,\ A = true$$

$$p = false,\ q = true,\ r = false,\ A = true$$

$$p = false,\ q = false,\ r = true,\ A = true$$

$$p = false,\ q = false,\ r = false,\ A = true$$

En el apartado 2 no hay problemas con los nombres de las variables:

```
> ej18_2:=impl(not q and impl(p,q),not p);
```

$$ej18\_2 := \textbf{not}\ (\ \textbf{not}\ q\ \textbf{and}\ (\ \textbf{not}\ p\ \textbf{or}\ q)\ \textbf{and}\ p)$$

```
> tablapq(ej18_2);
```

$$p = true,\ q = true,\ A = true$$

$$p = true,\ q = false,\ A = true$$

$$p = false,\ q = true,\ A = true$$

$$p = false,\ q = false,\ A = true$$

En los apartados 3) y 4) hay que cambiar los nombres de m y n por q y r:

```
> ej18_3:=impl(impl(p,m or n),m):
> ej18_3:=subs(m=q,n=r,ej18_3):
> ej18_4:=impl(impl(p,m and n),m):
> ej18_4:=subs(m=q,n=r,ej18_4):
> tablapqr(ej18_3);
```

$$p = true,\ q = true,\ r = true,\ A = true$$

$$p = true,\ q = true,\ r = false,\ A = true$$

$$p = true,\ q = false,\ r = true,\ A = false$$

$$p = true,\ q = false,\ r = false,\ A = true$$

$$p = false,\ q = true,\ r = true,\ A = true$$

$$p = false,\ q = true,\ r = false,\ A = true$$

$$p = false,\ q = false,\ r = true,\ A = false$$

$$p = false,\ q = false,\ r = false,\ A = false$$

```
> tablapqr(ej18_4);
```

$$p = true,\ q = true,\ r = true,\ A = true$$

$$p = true,\ q = true,\ r = false,\ A = true$$

$$p = true,\ q = false,\ r = true,\ A = true$$

$$p = true,\ q = false,\ r = false,\ A = true$$

$$p = false,\ q = true,\ r = true,\ A = true$$

$$p = false,\ q = true,\ r = false,\ A = true$$

$$p = false,\ q = false,\ r = true,\ A = false$$

$$p = false,\ q = false,\ r = false,\ A = false$$

Ejercicio 5. Sean

$$A_1 := p \Rightarrow (q \vee r) \qquad A_2 := q \Rightarrow \neg r$$

$$A_3 := p \qquad A_4 := q \wedge \neg r.$$

Se trata de demostrar que la proposición siguiente P es una tautología:

$$A_1 \wedge A_2 \wedge A_3 \Rightarrow A_4.$$

Si tomamos $p = V$, $q = F$ y $r = V$ se observa que la proposición P toma el valor F con lo que no es una tautología.

Ejercicio 6.

1) Sea $p :=$ *H es un subgrupo*, $q := (H \neq \emptyset)$, $r :=$*el elemento neutro pertenenece a H*, $s :=$ *H es cerrado*. La primera hipótesis es:

$$\mathcal{A}_1 := (q \wedge r \wedge s) \Rightarrow p.$$

La segunda hipótesis es:

$$\mathcal{A}_2 := (q \wedge s) \Rightarrow r.$$

La conclusión es:

$$B := s \Rightarrow p$$

Se trata de demostrar si la siguiente proposición es una tautología:

$$(\mathcal{A}_1 \wedge \mathcal{A}_2) \Rightarrow B.$$

Y haciendo su tabla de verdad se comprueba que no lo es.

2) Sea p:=*a es perfecto*, q:=*a es par* e i:=*n impar* se comprueba que la proposición siguiente es en efecto una tautología:

$$((p \Rightarrow q \wedge i) \wedge q \wedge \neg i) \Rightarrow \neg p.$$

3) Si h es *hay huelga*, s es *el sindicato miente* y r es *el ministro tiene razón* entonces el razonamiento que se presenta no es correcto porque la siguiente proposición no es una tautología:

$$(\neg h \Rightarrow s \vee r) \wedge (h \vee (\neg h \wedge \neg r)) \Rightarrow \neg s.$$

Ejercicio 7. Sean p := *José ganó la carrera*, q :=*Pedro fue el segundo*, r :=*Ramón fue el segundo* y s :=*Carlos fue el segundo*.

El razonamiento que se presenta es correcto porque la siguiente proposición es una tautología:

$$((p \Rightarrow q \vee r) \wedge (q \Rightarrow \neg p) \wedge (s \Rightarrow \neg r) \wedge p) \Rightarrow \neg s.$$

En efecto, como la última hipótesis es p podemos suponer que p es verdadero. Como p es verdadero la primera hipótesis dice que o bien q es verdadero o lo es r. La segunda hipótesis indica que si q es verdadero entonces p es falso, por tanto no se puede dar q verdadero. Entonces r es verdadero. La sentencia contrarrecíproca de la tercera hipótesis indica que si r es verdadero entonces $\neg s$ es verdadero, como queríamos demostrar.

Ejercicio 8. Si $a > b$ entonces existe un número natural d tal que $a = b + d$. De igual manera si $b > c$ entonces existe un número natural d' tal que $b = c + d'$. De este modo usando ambas igualdades se puede escribir

$$a = b + d = (c + d') + d = c + (d + d')$$

con lo que efectivamente se verifica $a > c$.

Ejercicio 9. La tautología que justifica el método de demostración por casos es:

$$((p \Rightarrow a_1 \vee ... \vee a_n) \wedge (a_1 \Rightarrow q) \wedge ... \wedge (a_n \Rightarrow q)) \Rightarrow (p \Rightarrow q).$$

En el caso particular de $n = 2$ se tiene:

$$((p \Rightarrow a_1 \vee a_2) \wedge (a_1 \Rightarrow q) \wedge (a_2 \Rightarrow q)) \Rightarrow (p \Rightarrow q).$$

Cuya tabla de verdad está efectivamente formada sólo por V.

Ejercicio 10. El ejercicio es consecuencia de las siguientes equivalencias lógicas:

$$p \downarrow q \Leftrightarrow \neg(p \vee q)$$
$$p \downarrow p \Leftrightarrow \neg(p \vee p) \Leftrightarrow \neg p$$
$$\neg p \downarrow \neg q \Leftrightarrow \neg(\neg p \vee \neg q) \Leftrightarrow p \wedge q$$
$$p|q \Leftrightarrow \neg(p \wedge q)$$
$$p|p \Leftrightarrow \neg(p \wedge p) \Leftrightarrow \neg p$$
$$\neg p|\neg q \Leftrightarrow \neg(\neg p \wedge \neg q) \Leftrightarrow p \vee q$$
$$(p|q)|(p|q) \Leftrightarrow \neg(p|q) \Leftrightarrow p \wedge q$$

Ejercicio 11. Como la hipótesis $P(x) \wedge \neg P(x)$ es siempre falsa la implicación es siempre verdadera.

Ejercicio 12. Se deducen directamente de la tautología:

$$\exists x\ P(x) \Leftrightarrow \neg(\forall x \neg P(x))$$

en el primer caso aplicándola a $\neg P(x)$ y en el segundo caso tomando la equivalencia de las negaciones.

Ejercicio 13. Razonamos demostrando la sentencia contrarrecíproca: si $a \neq 0$, como es $a \geq 0$ entonces se tiene $a > 0$. De este modo tomando $\epsilon = a$ se verifica que existe ϵ verificando $0 < \epsilon \leq a$.

Ejercicio 14.

1) Sea el universo U de las personas y los predicados $E(x) =$ *ser español* y $P(x) =$ *ser periodista*:

$$\neg(\forall x \in U(E(x) \Rightarrow P(x)))$$

o equivalentemente

$$\exists x \in U(E(x) \wedge \neg P(x)).$$

Su negación es:

$$\forall x \in U(E(x) \Rightarrow P(x))$$

esto es, *todos los españoles son periodistas.*

2) Sea U el conjunto de los caminantes y el predicado $B(x)$ *x bosteza.* Entonces:

$$(\exists x \in U\ B(x)) \Rightarrow (\forall x \in U\ B(x)).$$

Su negación es:

$$(\exists x \in U\ B(x)) \wedge (\exists x \in U\ \neg B(x)).$$

Esto es, *existen personas que bostezan y personas que no bostezan.*

3) En el universo U de las personas se toman los predicados $M(x)$ *x es modisto* y $P(x, y)$ *x conoce a y*:

$$\forall x \in U\ \exists y \in U\ (P(x, y) \wedge M(y)).$$

Su negación es:

$$\exists x \in U\ \forall y \in U\ (\neg P(x, y) \vee \neg M(y)).$$

Esto es, *existen personas que no conocen a ningún modisto.*

4)

$$\forall p, q \in \mathbb{R}\ p < q\ \exists s \in \mathbb{Q}\ p < s < q$$

Su negación es:

$$\exists p, q \in \mathbb{R}\ p < q\ \forall s \in \mathbb{Q}\ ((s \geq q) \vee (s \leq p)).$$

Esto es, *existen dos números reales p, q sin números racionales entre ellos.*

5) Sea P el conjunto de los números primos:

$$\forall p \in P\ \exists q \in P\ p < q.$$

Su negación es:

$$\exists p \in P\ \forall q \in P\ p \geq q.$$

Esto es, *existe un número primo mayor que el resto de los números primos.*

Ejercicio 15. El $lim(x_n) \neq a$ si

$$\neg(\forall \epsilon \in \mathbb{R}^+(\exists n \in \mathbb{N}(\forall m \in \mathbb{N}(m > n \Rightarrow |x_m - a| < \epsilon))))$$

o equivalentemente, usando la equivalencia lógica entre la implicación $p \Rightarrow q$ y $\neg p \vee q$:

$$\exists \epsilon > 0 \; \forall n \; \exists m > n \; |x_m - a| > \epsilon.$$

Ejercicio 16. Demostraciones por inducción:

i) $1 + ... + n = n(n+1)/2$.
Base de inducción: en efecto se cumple que $1 = (1 \cdot 2)/2$.
Paso de inducción: usando la hipótesis de inducción se tiene:

$$1 + ... + n + (n+1) = (1 + .. + n) + (n+1) = n(n+1)/2 + (n+1).$$

Y es una comprobación verificar la igualdad de polinomios:

$$n(n+1)/2 + (n+1) = (n+1)(n+2)/2.$$

ii)

$$\Sigma_{k=1}^{n}(2k-1) = n^2.$$

Base de inducción: en efecto se cumple que $1 = 1$.
Paso de inducción: usando la hipótesis de inducción se tiene:

$$\Sigma_{k=1}^{n+1}(2k-1) = \Sigma_{k=1}^{n}(2k-1) + (2n+1) = n^2 + 2n + 1.$$

Y en efecto se tiene:

$$n^2 + 2n + 1 = (n+1)^2.$$

iii) $2^n \leq (n+1)!$.
Base de inducción: en efecto se cumple que $2 \leq 2! = 2$.
Paso de inducción: usando la hipótesis de inducción se tiene:

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

Y como $2 \leq n+2$ se tiene que:

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

como queríamos demostrar.

Ejercicio 17. Base de inducción: $((p \Rightarrow a_1) \wedge (a_1 \Rightarrow q)) \Rightarrow (p \Rightarrow q)$ es una tautología como vimos en los apuntes.

Paso de inducción: Es una consecuencia inmediata de la equivalencia lógica:

$$(p \Rightarrow a_1 \vee ... \vee a_n \vee a_{n+1}) \Leftrightarrow ((p \Rightarrow a_1 \vee ... \vee a_n) \vee (p \Rightarrow a_{n+1})).$$

Ejercicio 19. Sea $x \in U$ donde U es el conjunto de estudiantes. Sea $R(x)$: *el estudiante x es de esta clase.* Sea $P(x)$: *El estudiante x mide estrictamente menos que 1.70.* El razonamiento se puede simbolizar en la forma

$$(H_1 \wedge H_2) \Rightarrow T$$

donde

$$H_1 := \forall x \in U \ R(x) \Rightarrow) P(x) \vee \neg P(x)).$$

$$H_2 := \neg P(a)$$

$$T := R(a)$$

y donde a representa a Pedro. Observamos que H_1 es siempre V y que tomando $R(a) := F$ y $P(a) := F$ se tiene que el razonamiento no es válido.

Ejercicio 20.

Ningún país ha ganado todas las medalla:

$$\neg(\exists x \forall y \ M(x,y).$$

O equivalentemente

$$\forall x \exists y \neg M(x,y).$$

Todos los países han ganado alguna medalla: $\forall x \exists y \ M(x,y)$.
Algún país ha ganado alguna medalla: $\exists x \exists y \ M(x,y)$.
Algún país no ha ganado ninguna medalla $\exists x \forall y \ \neg M(x,y)$.

Ejercicio 21. $P(2,3) : F$, $P(5,10) : V$, $P(2,3) \wedge P(5,10) : F$, $P(2,3) \vee P(5,10) : V$, $P(2,3) \Rightarrow P(5,10) : V$, $\forall m \in \mathbb{N}\ \forall m \in \mathbb{N}\ P(m,n) : F$, $\exists m \in \mathbb{N}\ \forall n \in \mathbb{N}\ P(m,n) : V$,$\exists n \in \mathbb{N}\ \forall m \in \mathbb{N}\ P(m,n) : F$, $\forall n \in \mathbb{N}\ P(1,n) : V$, $\forall m \in \mathbb{N}\ P(m,1) : F$.

Ejercicio 22.

Si vale menos de 1000 ptas, comeré en la cafetería.

Sea U el conjunto de posibles precios de la comida. Sea $P(x) : x < 1000$. Sea q : *comeré en la cafetería.* Nuestra frase es, en lenguaje formal, $P(x) \Rightarrow q$. Su negación es $\neg\left(P(x) \Rightarrow q\right)$, que es lógicamente equivalente a $\neg\left(\neg P(x) \vee q\right)$. Por la ley de de Morgan, esto es equivalente a $P(x) \wedge \neg q$. En lenguaje natural:

vale menos de 1000 ptas y no como en la cafetería.

Todos los habitantes de Madrid viajan en metro.

Sea U el conjunto de habitantes de Madrid. Sea $P(x) : x$ *viaja en metro.* La frase es en lenguaje formal $\forall x \in U; P(x)$. Su negación es $\neg\forall x \in U\ P(x)$ que es lógicamente equivalente a $\exists x \in U\ \neg P(x)$. En lenguaje natural:

existe algún habitante de Madrid que no viaja en metro.

Ningún habitante de Móstoles coge el autobús.

Sea U el conjunto de habitantes de Móstoles. Sea $P(x) : x$ *coge el autobús.* La frase es en lenguaje formal $\neg(\exists x \in U\ P(x))$. Su negación es $\exists x \in U\ P(x)$. En lenguaje natural:

existe algún habitante de Móstoles que coge el autobús.

Hay personas en todas las ciudades que usan el transporte público.

Sea U el conjunto de personas y V el conjunto de ciudades. Sea $P(x,y)$: *La persona x usa el transporte público en la ciudad y.* Entonces, en lenguaje formal, nuestra frase es $\forall y \in V\ \exists x \in U\ P(x,y)$. Su negación es $\neg\forall y \in V\ \exists x \in U\ P(x,y)$, que es lógicamente equivalente a $\exists y \in V\ \neg\exists x \in U\ P(x,y)$ equivalente a su vez a $\exists y \in V\ \forall x \in U\ \neg P(x,y)$. En lenguaje natural:

hay ciudades en las que ninguno de sus habitantes usa el transporte público.

Ejercicio 23. a)

$$P(n) : \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

Verifiquemos la base de inducción $P(1)$. Por una parte,

$$\sum_{k=1}^{1} k^2 = 1$$

y por otra

$$\frac{1(1+1)(2\cdot 1+1)}{6} = 1$$

En consecuencia, $P(1)$ es V.

Verifiquemos que bajo la hipótesis de inducción ($P(n)$ es V) es $P(n) \Rightarrow P(n+1)$ también V; es decir, que $P(n+1)$ es V. Para ello calculamos

$$\sum_{k=1}^{n+1} k^2 = \sum_{k=1}^{n} k^2 + (n+1)^2 = \text{(por hipótesis de inducción)}$$

$=$

$$\frac{n(n+1)(2n+1)}{6} + (n+1)^2 = (n+1)\left[\frac{n(2n+1)}{6} + (n+1)\right] =$$

$$= (n+1)\left[\frac{2n^2+7n+6}{6}\right] == (n+1)\frac{((n+1)+2)(2(n+1)+1)}{6}$$

lo que implica que, en efecto, es $P(n+1)$ verdadero. Por inducción, $\forall n \in \mathbb{N}$, $P(n)$ es V.

b)

$$P(n) : 2^n > n+1 \text{ para } n \geq 2.$$

Nuestro argumento comienza en $n = 2$. Por tanto, la base de inducción la podemos tomar como $P(2)$. $P(2)$ es V, ya que $2^2 > 2+1$. La hipótesis de inducción es que $P(n)$ es V. Verifiquemos que bajo esta hipótesis, es $P(n+1)$ verdadera:

$$2^{n+1} = 2^n 2 > \text{(por hipótesis de inducción)}$$

$$> (n+1)\cdot 2 = n+2+n > n+2 = (n+1)+1$$

Luego $\forall n \in \mathbb{N}$, $P(n)$ es V.

c)

$$P(n) : x^n - y^n \text{ es divisible por } x-y$$

La base de inducción es V, ya que $x - y$ es divisible por $x - y$ (obvio). La hipótesis de inducción es que $x^n - y^n$ es divisible por $x - y$. Veamos si $P(n+1)$ es V:

$$x^{n+1} - y^{n+1} = x(x^n - y^n) + (x - y)y^n$$

que es claramente divisible por $(x - y)$ si lo es $(x^n - y^n)$. Por inducción, $\forall n \in \mathbb{N}$, $P(n)$ es V.

Ejercicio 24. Sea $P(x, y)$: *y es el padre de x*. Es claro que todo x tiene un padre, pero no es cierto que exista al menos un x que es hijo de y (que puede no tener hijos en absoluto).

Ejercicio 25. Hay cuatro casos:
a) $x \geq 0, y \geq 0$. Entonces $xy \geq 0$ y se tiene $|xy| = xy = |x|\,|y|$.
b) $x < 0, y \geq 0$. Entonces $xy \leq 0$ y se tiene $|xy| = -xy = (-x)y = |x|\,|y|$.
c) $x \geq 0, y < 0$. Entonces $xy \leq 0$ y se tiene $|xy| = -xy = x(-y) = |x|\,|y|$.
d) $x < 0, y < 0$. Entonces $xy > 0$ y se tiene $|xy| = xy = (-x)(-y) = |x|\,|y|$.

Ejercicio 26. *Todo entero mayor que 17 es el cuadrado de un número entero.* Por ejemplo, el 21 no es el cuadrado de un entero, ya que $4^2 = 16$ es menor que 21 y $5^2 = 25$ es mayor.

Todo entero mayor que 6 es múltiplo de 2 ó de 3. El 23 es primo.

$100n + 1 > n^2$. Basta tomar n suficientemente grande. Por ejemplo, $n = 1000$.

Ejercicio 27. Si $\sqrt{2}$ es racional entonces existen $p, q \in \mathbb{N}$ primos entre sí de modo que

$$\sqrt{2} = p/q.$$

Elevando al cuadrado se obtiene $2q^2 = p^2$ con lo que p^2 es par y por tanto (demostrado en los apuntes) p es par. Si p es par entonces existe $r \in \mathbb{N}$ de modo que $p = 2r$ con lo que $p^2 = 4r^2$. De este modo $2q^2 = p^2 = 4r^2$ de donde se deduce que q^2 es par y por tanto q es par. Entonces p y q son ambos pares en contradicción con el hecho de que los hemos tomado primos entre sí.

Si $\pi + x$ es racional entonces existe $y \in \mathbb{Q}$ de modo que $\pi + x = y$. Despejando obtenemos $\pi = y - x$, con lo que π es la diferencia de dos

números racionales por tanto racional. Como π no es racional tenemos la contradicción.

Ejercicio 28. a) Sea p :=*salgo a la calle* y q :=*me visto*. Entonces, el razonamiento es

$$[(p \Rightarrow q) \wedge p] \Rightarrow q.$$

El razonamiento es válido (modus ponens).

b) Sea p :=*la comida incluye el primer plato* y q :=*la comida incluye el segundo plato*. El razonamiento sería $[p \wedge q] \Rightarrow p$. Esto es una tautología, ya que la implicación solo puede ser falsa si $p \wedge q$ es V y p es F. Pero esto es imposible, porque para que $p \wedge q$ sea V es necesario que p sea V. El razonamiento es correcto.

c) Sea p :=*Venda es una palabra de género femenino*. Sea q :=*Venda es una palabra singular*. El razonamiento sería $p \Rightarrow [p \wedge q]$. Esto no es una tautología, ya que si p es V y q es F se tiene $V \Rightarrow F$ que es F. El razonamiento no es correcto.

d) Sea v :=*vale menos de 100*, s :=*tengo dinero suficiente*, c :=*lo compro*. El razonamiento es correcto puesto que la siguiente sentencia es una tautología:

$$((v \Rightarrow s) \wedge (s \Rightarrow c) \wedge v) \Rightarrow c.$$

Capítulo 2

Algoritmos

En este capítulo abordaremos el tema del diseño y evaluación de algoritmos, es decir, de procedimientos automáticos de resolución de problemas. Se definirá el concepto de algoritmo, se describirá el lenguaje en que vamos a escribir los algoritmos (pseudocódigo) y se introducirá una medida de la bondad de un algoritmo: la complejidad en el peor de los casos, esto es, una estimación del número de operaciones básicas que realiza el algoritmo en el caso más complicado.

Se pretende que al final del capítulo el alumno:

- Conozca los conceptos de modelo computacional, algoritmo y complejidad y sepa discernir si un procedimiento es o no un algoritmo.
- Escriba algoritmos en pseudocódigo.
- Construya algoritmos sencillos y sea capaz de analizarlos, estudiando su complejidad.

2.1 Definiciones y ejemplos

Uno de los puntos principales de relación entre las matemáticas y la informática es la *resolución automática de problemas*, esto es, la construcción de métodos que permitan abordar con la ayuda del ordenador situaciones matemáticamente complicadas o laboriosas. Las calculadoras y, más recientemente, los programas de cálculo simbólico (Derive, Maple, Mathematica, Mathlab...) son ejemplos de la ayuda eficiente que presta la máquina en

tediosas o complejas cuentas u otros procedimientos donde habitualmente el tiempo y la probabilidad de error al realizarse por una persona son muy grandes.

El cliente plantea un problema que se puede formular en términos precisos y el informático hace riguroso el **planteamiento** del problema, estudia si **presenta solución**, diseña un **procedimiento** para buscar dicha solución, **demuestra** que su método resuelve el problema y estudia la **eficiencia** de dicho método.

Problema real
Planteamiento formal
Presenta solución
Algoritmo que da la solución
Demostración
Estudio de la eficiencia

En el tema anterior se introdujo el lenguaje de la lógica como ejemplo de lo que quiere decir ser riguroso y formalizar el enunciado de un problema. En este tema analizaremos las siguientes etapas del proceso: construcción de **algoritmos**, demostración de su solvencia y estudio de su eficiencia, introduciendo el concepto de **complejidad** y su formalización matemática.

Definamos el concepto de algoritmo, poniendo el acento en los elementos que lo conforman.

Definición 2.1.1 *Se define* **algoritmo** *como un procedimiento constructivo para la resolución de un problema que consta de los siguientes elementos: unos* datos de entrada *de naturaleza precisamente definida, una* cantidad finita de instrucciones *ordenadas que aplicadas a los datos de entrada ofrecen, tras una cantidad finita de operaciones, una* solución *precisa del problema de partida.*

Dos comentarios sobre el término algoritmo:

i) proviene del nombre del matemático árabe del siglo IX *Al-khowarizmi*,

ii) tiene un significado más amplio que el meramente aplicado a las matemáticas, como procedimiento definido paso a paso y encaminado a un cierto fin.

Resaltamos que:

i) la naturaleza de los datos de entrada debe ser exactamente descrita (una lista de números enteros, un numero racional...);

ii) las operaciones que el algoritmo realiza son siempre una cantidad finita cuando se aplican a unos datos de entrada de los que admite el algoritmo, esto es, el algoritmo siempre termina, aportando un valor de salida;

iii) se describe una demostración o al menos una evidencia de que el algoritmo resuelve el problema;

iv) se estudia en profundidad y en términos precisos la complejidad del algoritmo, es decir, de alguna manera (que precisaremos), el tiempo (o el espacio) que va a necesitar para obtener la solución de este problema.

v) la respuesta es precisa, lo que significa que se obtiene el resultado esperado. Esto no quiere decir que la respuesta sea necesariamente exacta, sino que cumple las condiciones que se la exigen. A saber, se puede por ejemplo diseñar un algoritmo para obtener el valor numérico aproximado de una cierta integral (o un límite, o una operación...) con un error del orden de las milésimas. Si el algoritmo está bien construido, el resultado no es necesariamente el valor exacto de dicha integral (o límite, u operación...), pero verifica las condiciones pedidas, esto es, aproxima la solución exacta con un error menor que 1 milésima.

De las observaciones i) y ii) del párrafo anterior se deduce que cuando se diseña un algoritmo se debe saber con qué tipo de datos se puede trabajar y cuáles son las operaciones básicas que se pueden considerar; estos dos elementos constituyen lo que se suele denominar **modelo computacional**.

Un modelo computacional habitual es el conocido como **RAM** (*random access machine*) donde los datos están formados por números enteros y las operaciones básicas son las siguientes:

i) **Asignación** a un dato de una dirección de memoria.

ii) Las operaciones de **suma, resta, multiplicación y división** (cálculo de cociente y resto) de números enteros.

iii) La **comparación** de dos números enteros a, b, pudiendo decir si $a < b$, $a = b$ o $a > b$.

Este es el modelo computacional con que nosotros vamos a trabajar.

Observar que un modelo computacional es un ente abstracto. La arquitectura de los ordenadores por un lado y por otro la tecnología de la

programación hacen de un modelo computacional una máquina concreta que admite esos datos (adecuadamente representados) como datos de entrada y realiza con ellos esas operaciones básicas y todas aquellas otras que se pueden realizar como combinación de las operaciones básicas.

En este sentido podemos poner otros ejemplos de modelos computacionales.

Ejemplos 2.1.2 *i) La Geometría con regla y compás. Donde los datos son puntos, rectas y circunferencias y las operaciones básicas son:*

a) apoyar la regla en un punto,
b) apoyar una pata del compás en un punto,
c) apoyar una pata del compás en una recta,
d) producir una recta con la regla,
e) producir una circunferencia con el compás,
f) intersecar rectas, circunferencias y rectas con circunferencias.

ii) El modelo conocido como Real RAM donde los datos de entrada son números reales en lugar de enteros y las operaciones básicas son las mismas que en el RAM.

Y hay una primera interesante cuestión de saber **qué tipo de problemas se pueden resolver** dentro de un modelo computacional. Es conocido (aunque tardó varios siglos en descubrirse) que los problemas de la trisección de un ángulo (dividir un ángulo en tres sectores iguales), la duplicación del cubo (dado un cubo construir otro que tenga exactamente el doble de volumen que el primero) y la cuadratura del círculo (dado un círculo construir un cuadrado con exactamente el mismo área) no se pueden resolver con regla y compás, esto es, no hay un algoritmo cuyas instrucciones estén formadas por combinaciones de las operaciones básicas del modelo computacional denominado *Geometría con regla y compás* que permita realizar esas construcciones. Una exposición sencilla que relaciona estos problemas clásicos de las matemáticas con las ciencias de la computación se puede encontrar en [M].

Una segunda cuestión importante es la de conocer los **problemas que realmente se pueden resolver** en un modelo computacional. Es decir, aquellos problemas para los que la sucesión de operaciones que efectúa el algoritmo que los resuelve precisa de un tiempo razonable (no de miles de

años por ejemplo) para aportar la solución del problema. Haremos otras consideraciones acerca de esta segunda pregunta más adelante cuando hablemos de la complejidad.

Ejemplo 2.1.3 *Problema que plantea el cliente: contabilidad, una lista de ingresos y una lista de gastos. Obtención del balance.*

Etapa 1: Planteamiento en términos formales: se tienen dos listas de números enteros de tamaños n y m correspondientes a los ingresos y a los gastos respectivamente. La suma de los elementos de la primera lista proporciona los ingresos. Con el mismo procedimiento para la segunda lista se obtienen los gastos. El balance es la diferencia entre ingresos y gastos.

Etapa 2: Procedimiento para resolver el problema. Es evidente que el problema tiene solución y que se podría resolver manualmente. Se trata de escribir ese procedimiento manual para que se pueda hacer automáticamente.

Datos de entrada: $a_1, a_2, ..., a_n$ y $b_1, b_2, ..., b_m$, dos listas de números enteros.

Procedimiento: generar un mecanismo que vaya recorriendo los valores de la lista de ingresos desde el primero hasta el enésimo y de los gastos desde el primero hasta el emésimo y dos variables, una donde se van añadiendo los valores de los ingresos y otra para los gastos. Finalmente hacer la diferencia para obtener el balance.

Una vez que sabemos cómo resolver el problema queremos escribir el algoritmo de manera que sea claro en su escritura y fácil de traducir a un lenguaje concreto de programación. El lenguaje de los algoritmos en nuestro contexto, esto es, en el modelo *RAM* se suele conocer como *pseudocódigo* y está formado por los siguientes elementos:

Los **datos de entrada** son variables que se puedan sustituir por constantes para que el algoritmo actúe sobre ellas. Normalmente representaremos las variables por letras.

Para **asignar** a una variable x un valor a escribiremos:

$$x := a.$$

Para **sumar, restar o multiplicar** números enteros utilizaremos los signos habituales:

$$a + b,\ a - b,\ a \cdot b.$$

Para el cociente y el resto de la **división entera**:

$$a/b,\ a\ mod\ b.$$

Donde, por ejemplo, $5/2$ es 2 y $5\ mod\ 2$ es 1. Atención a la diferencia con la notación matemática habitual. En el contexto de nuestro pseudocódigo $5/2$ no es una fracción sino un número entero, justamente la parte entera de dicho número.

Las comparaciones se representan con los signos $\leq$ (menor o igual), $\geq$ (mayor o igual), $<$ (estrictamente menor), $>$ (estrictamente mayor), $=$ (igual) y $\neq$ (distinto).

Esto con respecto a las operaciones básicas.

También introducimos ciertas instrucciones.

- Podemos escribir frases **condicionales** del tipo:

  If condición **then** operación 1 **else** operación 2.

Si ocurre la condición entonces se efectúa la operación 1 y si no ocurre se efectúa la operación 2. Si no se escribe la segunda parte de la condicional (la que empieza con *else*) se interpreta que si la condición no ocurre no se hace nada. Cabe señalar que también podemos tener varias condiciones que separaremos con comas.

Ejemplo 2.1.4 *Por ejemplo:*

Entrada: a, b
If $a > b$ then $c := a$ else $c := b$.
Salida: c (el más grande entre los dos números).

- Repetición de operaciones denominadadas **bucles**.

1. **For** $i = a$ **to** b operación.

Donde la operación se repite desde el valor a del contador i hasta que éste toma el valor b, increméntandose el valor del contador en una unidad cada vez que se realiza la operación.

Ejemplo 2.1.5 *Por ejemplo:*

Entrada: a, b (numeros enteros positivos)
$c := 0$
For $i = 1$ to b
  $c := c + a$
Salida: c (el producto ab).

Observar que las operaciones que se realizan dentro del bucle se escriben un poco más en el interior del párrafo para distinguirlas.

2. **While** condición, operación.

Mientras la condición se cumple se repite la operación.

Ejemplo 2.1.6 *Por ejemplo el bucle anterior se puede escribir como:*

$i := 1$
While $i \leq b$
  $c := c + a$
  $i := i + 1$

Y de nuevo señalar la posibilidad de que se tengan varias condiciones separadas por comas.

Estas normas de escritura nos permiten escribir el algoritmo de la contabilidad de una manera clara y precisa. Esta manera de escribir, según las normas propuestas la llamaremos *pseudocódigo*. Además resulta fácilmente transcribible a un lenguaje de programación con la sintaxis adecuada del lenguaje escogido.

Datos de entrada: $a_1, ..., a_n; b_1, ..., b_m$
$I := a_1$, $i := 1$
While $i \leq n - 1$
  $i := i + 1$
  $I := I + a_i$
$Ingresos := I$
$G := b_1$ $j := 1$
While $j \leq m - 1$
  $j := j + 1$

$G := G + b_j$
$Gastos := G$
Salida: $Balance := Ingresos - Gastos$

Etapa 3: Estudio de la solvencia del algoritmo: comprobemos que el algoritmo termina. En efecto, al introducir los datos de entrada, se establecen los valores de n y m que son dos números enteros positivos. Así se entra en el primer bucle que se repite exactamente $n-1$ veces (desde $i = 1$ hasta $n-1$), cuando i toma el valor n ese proceso termina. Lo mismo pasa en el segundo bucle cambiando el papel de n por el de m. Finalmente hay una asignación del valor correspondiente al balance con lo que el proceso finaliza. El algoritmo da la solución al problema porque copia exactamente el proceso manual que realizaría un contable.

Etapa 4: Estudio de la eficiencia del algoritmo. Para realizar dicho estudio necesitamos algunas definiciones precisas, que se establecerán más adelante, para saber en qué términos se va a medir esa eficiencia. Posponemos este estudio para después de la definición del concepto de *complejidad*.

Ejercicio 29 *Aplicar el agoritmo anterior a dos listas concretas de números. (Es una manera adecuada de observar el funcionamiento de un algoritmo.)*

Ejemplo 2.1.7 *Encontrar el valor máximo en una lista de números enteros.*

Datos de entrada: una lista de enteros $a_1, a_2, ..., a_n$.

Procedimiento: definir una variable a que toma inicialmente el valor de a_1 y que va comparándose con los distintos valores a_2, a_3, ... hasta a_n cambiando su valor (por el de la correspondiente a_i) si en la comparación es estrictamente menor.

Datos de entrada: $a_1, ..., a_n$
$a := a_1$, $i := 1$
While $i \leq n - 1$
  $i := i + 1$
  If $a < a_i$ then $a := a_i$
Salida: a

O equivalentemente usando el bucle *for*:

Datos de entrada: $a_1, ..., a_n$
$a := a_1$
For $i = 2$ to n
  If $a < a_i$ then $a := a_i$
Salida: a

Ejercicio 30 *Aplicar el algoritmo a una lista concreta de números enteros. Mostrar que el algoritmo termina.*

Ejemplo 2.1.8 *El siguiente procedimiento no es un algoritmo porque no termina. Esto quiere decir que realiza una cantidad no finita de operaciones. Si lo consideramos implementado por una máquina, dicha máquina comienza a ejecutar operaciones pero no termina nunca.*

$k := 0$
For $a = 1$ to 10
  $b := a$
  While $b \leq 10$
    $k := k + 1$
Salida:= k

Observamos que el valor inicial de la a es 1, por tanto b toma inicialmente el valor 1, que es menor o igual que 10. Se realiza entonces el bucle interior, donde k cambia de valor pero b no cambia de valor. Así, la condición $b \leq 10$ se verifica siempre y el procedimiento nunca sale de ese bucle.

Ejercicio 31 *Consideremos el siguiente algoritmo:*

Entrada: $n \in \mathbb{N}$
c:=0
For i=1 to n
  For j=1 to i
    c:=c+j
Salida: c

Explicar qué hace el algoritmo y dar una fórmula para la salida.

Terminamos esta sección con la construcción de algunos algoritmos más. Veamos cómo hacer algunas operaciones básicas en listas. Para ello introduciremos algunas instrucciones nuevas. Usaremos una función $longitud(L)$ que indica el número de elementos de una lista L y una función $elementos(L)$ que indica si la lista es vacía (saca un 0) o no (saca un uno). Estas dos funciones no son operaciones elementales de nuestro modelo pero entenderemos que las hacemos, de alguna manera, al introducir la lista. Además llamaremos a un algoritmo denominado $min(L)$ que indica cuál es el mínimo de una lista, propuesto como ejercicio más adelante. Finalmente permitiremos hacer asignaciones del tipo $x := L'$ donde L es una lista.

Problema. Dada una lista $L = a_1, \ldots, a_n$ y un número natural $1 \leq i \leq n$, construir una lista con un elemento menos, resultado de borrar de L el elemento que ocupa el lugar i-ésimo. Lo denotaremos $Borrar(L, i)$.

Ejemplo. Sea $L = 3, 4, 2, 8$ entonces $Borrar(L, 2) = 3, 2, 8$.

Algoritmo $Borrar(L, i)$

Entrada: $L = a_1, \ldots a_n$, i
For $j = i$ to $n - 1$
$a_j := a_{j+1}$
$a_n := null$ (deshace la asignación de a_n)
Salida: L

Problema. Sea m un número entero. Dada una lista $L = a_1, \ldots, a_n$ y un número natural $1 \leq i \leq n + 1$, construir una lista con un elemento más, resultado de añadir a L en el lugar i-ésimo el elemento m, desplazando en una unidad los términos a partir de él. Lo denotaremos $Añadir(L, m, i)$.

Ejemplo. Sea $L = 3, 4, 2, 8$ entonces $Añadir(L, 7, 3) = 3, 4, 7, 2, 8$.

Algoritmo $Añadir(L, m, i)$

Entrada: $L = a_1, \ldots a_n$, i, m
$j := n + 1$
While $j > i$
$a_j := a_{j-1}$
$j := j - 1$

$a_i := m$
Salida: L

Problema. Dadas dos listas $L = a_1, \ldots, a_n$ y $L' = b_1, \ldots, b_m$ construir la lista de longitud $n+m$ resultado de poner los elementos de L' a continuación de los elementos de L, esto es, el resultado será $a_1, \ldots, a_n, b_1, \ldots, b_m$. Lo denotaremos $Concatenar(L, L')$.

Ejemplo. Sea $L = 3, 4, 2, 8$ y $L' = 4, 8$ entonces

$$Concatenar(L, L') = 3, 4, 2, 8, 4, 8.$$

Algoritmo $Concatenar(L, L')$

Entrada: $L = a_1, \ldots a_n$, $L' = b_1, \ldots, b_m$
For $i = 1$ to m
$L :=$*Añadir*$(L, b_i, n + i)$
Salida: L

Problema. Sea m un número entero. Dada una lista $L = a_1, \ldots, a_n$ y un número natural $1 \leq i \leq n + 1$, construir una lista con los mismos elementos resultado de sustituir L_i por m. Lo denotaremos $Sustituir(L, i, m)$.

Ejemplo. Sea $L = 3, 4, 2, 8$, $m = 3$, $i = 4$ entonces $Sustituir(L, 3, 4) = 3, 4, 2, 3$.

Algoritmo $Sustituir(L, i, m)$

Entrada: $L = a_1, \ldots a_n$, i, m
$L_i := m$
Salida: L

Problema. Sean dos listas de números reales $L = a_1, \ldots, a_n$ y $L' = b_1, \ldots, b_m$ ordenadas de menor a mayor: construir un algoritmo para **mezclar** ambas listas, esto es, construir una lista M de longitud $n + m$ con los elementos de

L y L' ordenados de menor a mayor. El funcionamiento del algoritmo debe ser el siguiente:

Ejemplo. Sean $L = 1, 5, 89$, $L' = 3, 6, 8$ entonces $M := mezclar(L_1, L_2) = 1, 3, 5, 6, 8, 89$.

Idea. Comparar los primeros elementos de L y L' y elegir el menor como primer elemento de la lista salida.

Con la idea señalada y estos algoritmos se puede escribir una solución del problema de la siguiente manera.

Algoritmo $mezclar(L, L')$

Entrada: L, L' dos listas ordenadas:
While $elementos(L)elementos(L') = 1$
If $min\{L_1, L'_1\} = L_1$ then $j = 0$ else $j = 1$
$M_i := min\{L_1, L'_1\}$
If $j = 0$ then $L = Borrar(L, 1)$ else $L' = Borrar(L', 1)$
If $longitud(L) > 0$ then $M := concatenar(M, L)$ else
$M = concatenar(M, L')$
Salida: M

Es siempre de utilidad verificar el funcionamiento de un algoritmo sobre una lista particular. Llamaremos a estos algoritmos en secciones posteriores cuando los necesitemos.

2.2 Algoritmos de búsqueda y de ordenación

Dos problemas típicos que sirven para ilustrar distintas maneras de diseñar algoritmos son los problemas de búsqueda (determinar si un dato está o no en una lista de datos) y de ordenación (dada un lista, ordenarla de alguna manera preestablecida). Son problemas importantes dado que el manejo adecuado de la información es fundamental para la tecnología actual. Estudios de mercado, el genoma humano, determinación de patrones de comportamiento... son algunas de las situaciones en las cuales un adecuado tratamiento de los datos es crucial para poder extraer todas las consecuencias de los mismos.

2.2.1 Algoritmos de búsqueda

Se trata de resolver la siguiente cuestión:

Problema: dada una lista de números enteros determinar si un número que se da está o no en la lista.

Datos de entrada: $a_1, a_2, ...a_n$; a. Una lista de números enteros y un entero a.

Algoritmo 1. Búsqueda Secuencial. Va comparando el elemento a con todos los de la lista. Si encuentra uno igual que a la salida es *sí* en caso contrario la salida es *no*.

Entrada: $a_1, ..., a_n$; a
$i := 1$
While $i \leq n$, $a \neq a_i$
  $i := i + 1$
If $i \leq n$ then $r :=$*Sí* else $r :=$*No*
Salida: r

Algoritmo 2. Búsqueda binaria. En este caso la lista de números enteros de la entrada debe estar ordenada de menor a mayor. En este procedimiento se compara el elemento a con el que ocupa el lugar central en la lista. Si es igual, el proceso termina; si es mayor, el proceso se repite con la segunda mitad de la lista; si es menor, con la primera mitad de la lista. En el siguiente algoritmo i y j representan en cada momento los subíndices de los elementos inicial y final de la lista que se trata, m es el subíndice que ocupa el lugar central.

Entrada: $a_1, ..., a_n$; a (con $a_1 \leq a_2 \leq ... \leq a_n$)
$i := 1$ $j := n$
While $i < j$, $a \neq a_{(i+j)/2}$
  $m := (i + j)/2$
  If $a_m < a$ then $i := m + 1$ else $j := m - 1$
If $i < j$ then $r :=$*Sí*
else If $a_{i+j/2} \neq a$ then $r :=$*No*
Salida: r

La búsqueda binaria es un ejemplo de una forma de abordar el diseño de algoritmos conocido como **divide y vencerás**. Este paradigma propone dividir el problema en problemas de tamaño más pequeño (en el ejemplo problemas de tamaño la mitad del de partida) y aplicar sucesivamente esta técnica de división hasta llegar a problemas triviales (en nuestro caso comparar dos números, que es una de nuestras operaciones básicas).

Observación 2.2.1 *Como veremos más adelante no resulta imprescindible ser completamente preciso al contar el número de operaciones que realiza un algoritmo, pues vamos a centrarnos en el estudio de su complejidad. Vamos a entender que ni las asignaciones de la entrada, ni el cómputo del tamaño de una lista guardada necesitan consumir niguna operación.*

Ejercicio 32 *Aplica los dos algoritmos para buscar el elemento 9 en la lista* 1, 3, 5, 7, 9 *y compara las operaciones que se realizan en cada uno de los casos, teniendo en cuenta la observación anterior.*

2.2.2 Algoritmos de ordenación

Se trata de resolver la siguiente cuestión:

Problema: dada una lista de números enteros distintos, $a_1, ..., a_n$, ordenarla de menor a mayor. (Por ejemplo: Entrada 2, 1, 6, 4. Salida: 1, 2, 4, 6.)

Datos de entrada: a_1, a_2, ..., a_n. Una lista de números enteros distintos.

Algoritmo 1. Ordenación burbuja. La idea en este algoritmo es comparar dos elementos sucesivos de la lista y en caso de que no estén ordenados adecuadamente intercambiar sus valores. Comenzamos comparando el primer elemento de la lista con el segundo e intercambiando los valores si es necesario, luego el segundo con el tercero y así hasta acabar la lista. Cuando acabamos la lista, el valor más grande se ha colocado en el último lugar (que es el que debe ocupar en la salida) por lo que debemos repetir el proceso con la nueva lista resultado de los intercambios prescindiendo del último valor.

Veamos un ejemplo:

Tomamos la lista $a_1 = 2$, $a_2 = 1$, $a_3 = 6$, $a_4 = 4$.

Como el primer valor $a_1 = 2$ es mayor que el segundo $a_2 = 1$ entonces se intercambian los valores:

$$a_1 = 1 \qquad a_2 = 2$$

Como el segundo valor es ahora $a_2 = 2$ menor que el tercero, el algoritmo no hace nada:

$$a_2 = 2 \qquad a_3 = 6$$

Como el tercer valor $a_3 = 6$ es mayor que el cuarto entonces se produce un intercambio de valores:

$$a_3 = 4 \qquad a_4 = 6$$

Se acaba la lista y la nueva lista es:

$$a_1 = 1, a_2 = 2, a_3 = 4, a_4 = 6.$$

En este caso la lista ya está ordenada, pero podría no ser así con lo que habría que repetir el proceso con la lista a_1, a_2, a_3.

Visto el ejemplo escribamos el algoritmo:

Entrada: $a_1, ..., a_n$.
For $j = 1$ to $n - 1$
For $i = 1$ to $n - j$
If $a_i > a_{i+1}$ then intercambiar sus valores (ver observación siguiente)
Salida: $a_1, ..., a_n$.

Observación 2.2.2 *Observamos que para intercambiar el valor de las variables a_i y a_{i+1} necesitamos hacer tres operaciones:*

$c := a_i$
$a_i := a_{i+1}$
$a_{i+1} := c$

Porque si sencillamente hiciésemos:

$a_i := a_{i+1}$
$a_{i+1} := a_i$

En la primera línea la variable a_i toma el valor de a_{i+1} y en la segunda al asignar a a_{i+i} el valor de a_i, como ésta ya tiene el valor de a_{i+1}, en realidad no hemos hecho nada. Por esto se necesita la variable auxiliar c.

Ejercicio 33 *Aplicar el algoritmo burbuja de ordenación a la lista 1, 3, 2, 9, 5, 4.*

Algoritmo 2. Algoritmo de selección. La idea de este algoritmo de ordenación es que en el paso j-ésimo mira la lista a_j, a_{j+1}, ..., a_n calcula el mínimo y si no es a_j (como debiera ser si la lista estuviera ordenada) lo intercambia con a_j.

Ejercicio 34 *Diseñar un algoritmo que tome una lista de números enteros $a_1, a_2, ..., a_n$ e indique cuál de ellos es el mínimo.*

Escribamos el algoritmo usando el ejercicio anterior. Esto es, dentro de nuestro algoritmo vamos a llamar a otro algoritmo, digamos *min*, que calcula el mínimo de una lista, la salida es $min(L) = m, i$ donde m es el mínimo e i el lugar que ocupa en la lista L. Llamamos a un algoritmo denominado $Intercambiar(L, i, j)$ que intercambia los valores de a_i y a_j en la lista L.

Entrada: $M := a_1, ..., a_n$.
$L := a_1, ..., a_n$
For $j = 1$ to $n - 1$
  If $min(L)_1 \neq a_j$ then
  $M := Intercambiar(M, j, min(L)_2 + j - 1)$
  $L := Borrar(L, j)$
Salida: M.

Veamos un ejemplo:

Tomamos la lista $a_1 = 2$, $a_2 = 1$, $a_3 = 6$, $a_4 = 4$.

Como el mínimo de la lista es a_2 y $a_2 \neq a_1$ entonces intercambian sus valores, obteniéndose por tanto:

$$a_1 = 1 \qquad a_2 = 2.$$

En la lista $a_2 = 2$, $a_3 = 6$, $a_4 = 4$ el mínimo es a_2 por lo que no se realiza ninguna operación.

Finalmente en la lista $a_3 = 6$, $a_4 = 4$ el mínimo es a_4 y por tanto se intercambian los valores de los dos elementos, ordenándose de este modo toda la lista.

Ejercicio 35 *Aplicar el algoritmo de selección para ordenar la lista 1, 3, 2, 9, 5, 4.*

Este es un ejemplo del paradigma de construcción de algoritmos denominado **algoritmo voraz** porque va construyendo una sublista ordenada cada vez más grande que al concluir contiene a todos los elementos de la lista inicial.

2.3 Complejidad

Como hemos señalado en las secciones anteriores debemos hacer un estudio de la eficiencia de un algoritmo. Esto resulta especialmente interesante cuando conocemos distintos algoritmos para abordar un mismo problema (como en la búsqueda y la ordenación) y debemos decidir cuál de ellos usar, según unos criterios razonables y defendibles.

Evidentemente hay dos parámetros que sería interesante minimizar: el tiempo y el espacio. Con el tiempo nos referimos a la cantidad de operaciones básicas que la máquina debe realizar en un algoritmo y con el espacio a la cantidad de memoria que el algoritmo utiliza en su funcionamiento.

Nos centraremos en el control del número de operaciones que el algoritmo efectúa, esto es, en el control del tiempo.

Definición 2.3.1 *Sea un algoritmo cuyos datos de entrada tienen tamaño n. Definimos la funcion $T(n)$,* **número de operaciones que realiza el algoritmo en el peor de los casos**, *como el valor máximo del número de operaciones que realiza el algoritmo entre todas las aplicaciones del mismo a las posibles entradas de tamaño n.*

El tamaño de la entrada en el modelo computacional (*RAM*) que hemos elegido estará relacionada con la cantidad de números enteros que involucra dicha entrada. Las operaciones a que se refiere la definición de la función denominada $T(n)$ son las operaciones básicas de nuestro modelo.

La función $T(n)$ para el ejemplo del cálculo del máximo

Reproducimos el algoritmo que teníamos para calcular el máximo de una lista:

Datos de entrada: $a_1, ..., a_n$
$a := a_1$ $i := 1$

While $i \leq n-1$
  $i := i+1$
  If $a < a_i$ then $a := a_i$
Salida: a

Definimos el tamaño de la entrada como la longitud de la lista. Es decir el valor de n. Analicemos el número de operaciones para una lista de longitud n. En el propio análisis iremos viendo cuál es el peor de los casos.

El algoritmo comienza con dos asignaciones (a e i) y determinando el valor de $n-1$ por medio de una diferencia (recordamos que en la observación 2.2.1 decíamos que el cómputo de n no precisa de ninguna operación para ser efectuado).
Después entra en un bucle que se repite $n-1$ veces y en cada repetición se hace:

una comparación que determina si entrar o no en el bucle (i con $n-1$),
una asignación a la i y una suma $i+1$,
una comparación de a con a_i y
finalmente una asignación que a veces se hace y a veces no. En el peor de los casos se hará.

Por tanto en el bucle se hacen $5(n-1)$ operaciones.
Después hay una comparación (i valiendo n con $n-1$) que nos saca del bucle y la asignación a la salida.
Resultado total en el peor caso:

$$T(n) = 3 + 5(n-1) + 2 = 5n.$$

Ejercicio 36 *Describir cómo es la lista que da el peor caso en este algoritmo.*

Observación 2.3.2 *Se podrían definir otro tipo de funciones para medir la cantidad de operaciones de un algoritmo, por ejemplo funciones de tiempo medio, $M(n)$, que nos dan el número de operaciones promedio que el algoritmo realiza cuando recibe una entrada de tamaño n. (Se suelen obtener de forma experimental.)*

Sobre la función $T(n)$ conviene hacer dos precisiones:

i) suele ser difícil calcularla exactamente;

ii) si cada operación necesita una fracción muy pequeña de tiempo para realizarse, no es precisa una exactitud total en el cálculo de $T(n)$, por ejemplo un error de una operación en el cálculo de $T(n)$ es totalmente irrelevante.

Estas dos precisiones nos invitan a la definición de la **complejidad del algoritmo** como un concepto que recoja la naturaleza asintótica de la función $T(n)$ y no su valor exacto. Dicha naturaleza que señalamos debe dar información de cómo crece la función cuando aumenta el tamaño n de la entrada, especialmente para valores muy grandes del tamaño n de la entrada. Para dar esta definición precisamos ciertos conceptos del análisis matemático.

Definición 2.3.3 *Sean $T, S : \mathbb{N} \to \mathbb{R}^+$ dos sucesiones reales con valores positivos. Se dice que $T(n)$* **domina a** *$S(n)$ si existen dos números reales n_0 y $k > 0$ de modo que para cada $n \geq n_0$ se verifique:*

$$S(n) \leq kT(n).$$

Sea $O(T(n))$ el conjunto de las sucesiones dominadas por $T(n)$, entonces si $T(n)$ domina a $S(n)$ escribiremos $S(n) \in O(T(n))$.

Si $S(n) \in O(T(n))$ y $T(n) \in O(S(n))$ diremos que $S(n)$ y $T(n)$ son **del mismo orden**.

Definición 2.3.4 *Se llama* **complejidad de un algoritmo** *al orden, $O(T(n))$, de su función $T(n)$ número de operaciones en el peor de los casos.*

La complejidad de un algoritmo es entonces una medida del número de operaciones que éste realiza en el peor de los casos para tamaños muy grandes de la entrada n y establece una jerarquía que permite determinar si un algoritmo es mejor que otro. Si el algoritmo A_1 tiene como función número de operaciones en el peor de los casos T_1 y, respectivamente, el algoritmo A_2 tiene como función T_2 y T_2 domina a T_1, entonces (según esta forma de comparar) el algoritmo A_1 es mejor (o al menos igual) que el algoritmo A_2. El análisis matemático (estudiado en la asignatura de Bases de Matemáticas) es el instrumento que permite establecer esta jerarquía.

Ésta no es la única forma de medir la eficiencia de los algoritmos. En muchas ocasiones los algoritmos no se van a usar para entradas muy grandes ni posiblemente va a interesar lo que ocurre en el peor de los casos.

Ejemplo 2.3.5 *Sean $T(n) = n^2$ y $S(n) = 2n^2$ se verifica que $T(n)$ y $S(n)$ son del mismo orden ya que para cada n natural se tiene que $T(n) \leq S(n)$ y que $S(n) \leq (1/2)T(n)$.*

Ejercicio 37 *Comprobar que todos los polinomios de grado s en las condiciones de la definición de dominancia, digamos,*

$$a_0 + a_1 x + ... + a_s x^s \quad a_s \neq 0$$

son del mismo orden.

Ejercicio 38 *Sean a y b números reales positivos mayores que 1. Demostrar que $log_a(n)$ y $log_b(n)$ son del mismo orden.*

Ejercicio 39 *Sea $l(n)$ el logaritmo neperiano. Demostrar la siguiente cadena de inclusiones:*

$$O(1) \subset O(l(n)) \subset O(n) \subset O(nl(n)) \subset O(n^2) \subset O(e^n)$$

Ejercicio 40 *Sean T y S sucesiones reales con valores positivos. Demostrar que si T domina a S entonces T domina a $T + S$.*

Demostrar, usando el apartado anterior, que si un algoritmo A_1 tiene como función número de operaciones en el peor de los casos T y respectivamente un algoritmo A_2 tiene la función S, y T domina a S, entonces la complejidad del algoritmo resultado de realizar sucesivamente A_1 y A_2 es $O(T + S) = O(T)$.

La terminología más usada es la siguiente:

• si $T(n)$ es del mismo orden que la función $S(n) = n$ se habla de **complejidad lineal**.

• si $T(n)$ es del mismo orden que $S(n) = n^2$ decimos **complejidad cuadrática**;

• así sucesivamente complejidad cúbica, cuártica... y en general **complejidad polinomial** si el orden de $T(n)$ es el de un polinomio;

• si $T(n)$ es del mismo orden que $l(n)$ se habla de **complejidad logarítmica**;

• si $T(n)$ es del mismo orden que e^n se habla de **complejidad exponencial**.

Volvemos a comentar aquí la cuestión de los problemas que son realmente resolubles en un modelo computacional. Estamos entonces preocupados por problemas que precisan de un tiempo razonable para ser resueltos. Por

ejemplo supongamos que un algoritmo A_1 tiene complejidad $O(T_1) = O(n)$, un algoritmo A_2 tiene complejidad $O(T_2) = O(n^2)$ y un algoritmo A_3 tiene complejidad $O(T_3) = O(n^6)$. Supongamos que la operación básica se realiza en una décima de segundo. Si tomamos una entrada de tamaño 10^3, entonces respectivamente:

$$T_1(10^3) = 10^3, \qquad T_2(10^3) = 10^6 \qquad T_3(10^3) = 10^{18}.$$

Mientras el primer algoritmo emplea apenas dos minutos, el segundo emplea algo más de un día y el tercero emplea más de mil millones de años en aportar la salida.

Analicemos la complejidad de los dos algoritmos de búsqueda.

Complejidad de la búsqueda secuencial. El peor de los casos es áquel en que a no está en la lista que se nos ha proporcionado. Como nos interesa la complejidad no nos preocupamos de las asignaciones que se hacen al principio y al final sino de cuántas veces se repite el bucle. Esto se justifica en el hecho de que $O(1 + T) = O(T)$ si T domina a 1. Como a no está en la lista, el bucle se repite n veces, y en cada una de ellas hay dos comparaciones, una asignación y una suma, es decir, hay $4(n - 1)$ operaciones. Por tanto este algoritmo es de **complejidad lineal**.

Complejidad de la búsqueda binaria. El peor de los casos es de nuevo el caso en que a no está en la lista que se nos ha proporcionado. Como señalamos en el ejemplo anterior, para calcular la complejidad nos interesa cuántas veces se repite el bucle. Veamos una idea acerca de este número de repeticiones. Cada vez que se realiza el bucle el tamaño de la lista queda reducido a la mitad. Tras el primer paso por el bucle tenemos una lista de tamaño $n/2$, tras el segundo paso será una lista de tamaño $n/4$ y así sucesivamente en s pasos por el bucle tendremos una lista de tamaño $n/2^s$. Por tanto, como el proceso termina cuando la lista está formada por un único elemento, se tiene:

$$1 = n/2^s,$$

es decir,

$$s = log_2(n).$$

De esta manera, el algoritmo de búsqueda binaria presenta **complejidad logarítmica**.

Propiamente los dos algoritmos no pueden ser comparados, ya que los datos de entrada de uno y otro no son exactamente iguales, en el segundo caso deben estar ordenados. Así pues, aunque a simple vista el segundo algoritmo resulta mejor (recordemos que $O(l(n)) \subset O(n)$), lo que propiamente se debe comparar es el algoritmo de búsqueda secuencial con un algoritmo resultado de realizar sucesivamente un algoritmo de ordenación y la búsqueda binaria. Lo que sí se puede concluir del análisis anterior es que cuando los datos de la lista están ordenados entonces ambos algoritmos se pueden aplicar y resulta más eficiente la búsqueda binaria.

Ejercicio 41 *Estudiar la complejidad de los algoritmos de ordenación presentados.*

Ejercicio 42 *Estudiar la complejidad del algoritmo de búsqueda resultado de concatenar el algoritmo de ordenación burbuja con el algoritmo de búsqueda binaria. Comparar con la complejidad de la búsqueda secuencial.*

Terminamos el capítulo con el interesante concepto de **complejidad inherente a un problema**. Se trata de encontrar el mejor algoritmo que resuelve un problema (siempre desde el punto de vista del concepto de complejidad presentado).

Así diremos que un **problema es de complejidad** $O(T)$ cuando:

i) hay un algoritmo que resuelve dicho problema cuya función número de operaciones en el peor de los casos es del mismo orden que T;

ii) cada algoritmo que resuelve el problema, cuya función número de operaciones en el peor de los casos la denotamos S, verifica que $T \in O(S)$. Esto es, cualquier otro algoritmo que resuelve el problema tiene complejidad igual o mayor que el de función T.

Demostraremos como ejemplo más adelante (cuando hablemos de árboles de decisión) que el problema de ordenación de listas es de complejidad $O(nl(n))$.

2.4 Algoritmos de búsqueda con Maple

Problema: dada una lista de números a y un número b determinar si dicho número es uno de los elementos de la lista.

2.4.1 Búsqueda secuencial

El primer algoritmo que presentamos toma una lista de números a y un número b y devuelve un vector donde cada entrada es: un uno si la entrada de la lista correspondiente es igual a b y un cero en caso contrario.

```
> Vector:=proc(a::list,b)
> local c,j,i:
> c:=[seq(0*j,j=1..nops(a))]:
> for i from 1 to nops(a) do
> if a[i]=b
> then c[i]:=1 fi:
> od:
> c; end:

> Vector([1,2,0,4,5],4);
```

$$[0, 0, 0, 1, 0]$$

Ejemplo de una aplicación a la teoría de números:

Determinar si 356 es el cuadrado de un número entero.

Como 10^2 =100 y 20^2=400 entonces si 356 fuera un cuadrado lo sería de un número entre 11 y 19; tomando la lista de dichos cuadrados podemos ver si 356 es un cuadrado.

```
> Vector([seq(x^2,x=11..19)],356);
```

$$[0, 0, 0, 0, 0, 0, 0, 0, 0]$$

Como no está en la lista (pues la salida del procedimiento *Vector* está formada únicamente por ceros) no puede ser un cuadrado.

Dado este algoritmo se puede responder a distintas preguntas, como por ejemplo:

a) el elemento b está en la lista a si el resultado de aplicar el procedimiento *Vector* a a contiene algún uno;

b) el número de unos de *Vector(a)* dice cuántas veces está b en la lista a;

c) si ya se sabe que se tiene un elemento de la lista, se puede construir un sencillo algoritmo que obtenga todos los lugares donde está el elemento b.

Para responder a las preguntas a) y b).

```
> preguntasayb:=proc(a::list,b)
> local i,l,c:
> l:=add(Vector(a,b)[i],i=1..nops(Vector(a,b))):
> if l=0 then
> c:=[no,0] else c:=[si,l] fi:
> c; end:
```

Lo aplicamos a un ejemplo concreto:

```
> preguntasayb([1,2,1,3,0,1,0],1);
```

$$[si, 3]$$

Para responder a la pregunta c)

```
> preguntac:=proc(a::list,b)
> local c,j,i:
> c:=[seq(0*j,j=1..preguntasayb(a,b)[2])]:  j:=1:
> for i from 1 to
> nops(a) do
> if Vector(a,b)[i]=1 then
> c[j]:=i:  j:=j+1:  fi:  od:
> c; end:
```

Lo aplicamos a un ejemplo concreto:

```
> preguntac([1,2,1,3,0,1,0],1);
```

$$[1, 3, 6]$$

En cualquier caso estos algoritmos tienen **complejidad lineal** ya que, cuando se tiene una lista de tamaño n , realizan esencialmente kn operaciones con k constante. Comprobamos esta última afirmación.

El procedimiento vector:
Vector:=proc(a::list,b)
local c,j,i:
c:=[seq(0*j,j=1..nops(a))]: (se hacen n asignaciones)
for i from 1 to nops(a) do (un bucle que se repite n veces)
if a[i]=b then c[i]:=1 fi: od: (se hace una comparación y una asignación)
c; end:
Total: *3n operaciones*

El procedimiento preguntasayb:
preguntasayb:=proc(a::list,b)
local i,l,c:

l:=add(Vector(a,b)[i],i=1..nops(Vector(a,b))): (se hace el procedimiento Vector por tanto $3n$ operaciones, luego se hacen como máximo n sumas)

if l=0 then c:=[no,0] (una comparación y una asignación)

else c:=[si,l] fi: c;

end:

Total: *4n+2 operaciones*

El procedimiento *preguntac*:

preguntac:=proc(a::list,b)

c:=[seq(0*j,j=1..preguntasayb(a,b)[2])]: (se hace el procedimiento preguntasayb por tanto $4n+2$ operaciones, luego se hacen como máximo n asignaciones)

j:=1: (1 asignación)

for i from 1 to nops(a) do (un bucle que se repite n veces y cada vez que se repite hace 3 operaciones)

if Vector(a,b)[i]=1 then c[j]:=i: j:=j+1:

fi: od: c;

end:

Total: $8n+3$

Este algoritmo tiene complejidad lineal O(n)

2.4.2 Búsqueda binaria

Se trata de un algoritmo que toma una lista ordenada de menor a mayor de números y realiza la búsqueda comparando b con el elemento que está en el medio de la lista. Si es igual, el algoritmo termina; si es menor, trabaja con una nueva lista (la de los elementos menores que el elemento intermedio) e itera el algoritmo; si es mayor, trabaja con una nueva lista (la de los elementos mayores que el elemento intermedio) y hace lo mismo.

```
> Binaria:=proc(a::list,b)
> local c,i,j,k,s:
> i:=1:  j:=nops(a):  c:=0;
> for s from 1 to nops(a) while j-i>-1 do
> k:=floor((i+j)/2):
> if b=a[k] then c:=1:  s:=nops(a)+1:  fi:
> if
> b>a[k] then i:=k+1:  fi:
> if b<a[k] then j:=k-1:  fi:  od:
> if c=1 then
> RETURN(si) else RETURN(no) fi:
> end:
```

Comprobemos el algoritmo en algunos ejemplos:

```
> Binaria([3,4,5,6,7,7.5,8,9],7.25);
```

$$no$$

```
> Binaria([3,4,5,6,7,7.5,8,9],9);
```

$$si$$

```
> Binaria([3,4,5,6,7,7.5,8,9],4);
```

$$si$$

```
> Binaria([seq(x^2,x=10..20)],169);
```

$$si$$

En el último ejemplo la salida es *si* porque $169 = 13^2$.

Para estudiar la complejidad es relevante saber cuántas veces como máximo se repite el bucle, ya que, cada vez que se repite, el número de operaciones que se efectúan es constante. Como la lista se va dividiendo en una lista de tamaño la mitad, hasta que queda una lista con un único elemento, el bucle se repite como máximo *log(n)* donde *log* es el logaritmo en base 2.

Binaria:=proc(a::list,b)
local c,i,j,k,s: i:=1: j:=nops(a): c:=0; (tres asignaciones)
el bucle se repite log(n) veces y cada vez se hace
for s from 1 to nops(a) while j-i>-1 do (una comparación)
k:=floor((i+j)/2): (una asignación)

if b=a[k] then c:=1: s:=nops(a)+1: fi: (una comparación y una asignación, en el peor de los casos el número no está en la lista)
if b>a[k] then i:=k+1: fi:
if b<a[k] then j:=k-1: fi: od:
if c=1 then RETURN(si) else RETURN(no) fi: (una comparación y una asignación)
end:
Total: ***4log(n)+5***
Este algoritmo tiene **complejidad logarítmica O(log(n))**.

2.5 Algoritmos de ordenación con Maple

2.5.1 Ordenación Burbuja

Presentamos el algoritmo burbuja para ordenar de menor a mayor una lista, esto es, los datos de entrada son una lista L de números enteros y la salida es una lista ordenada de menor a mayor con los mismos elementos que L.

Intercambiar dos elementos

Una operación que hace falta en este algoritmo es intercambiar dos elementos, el i-ésimo y el j-ésimo, de una lista que llamamos *lista*. La idea que vamos a utilizar para ello consiste en introducir una nueva variable transitoria, llamada *temp,* y seguir los tres pasos siguientes:

- Primero: almacenaremos el contenido de *lista[i]* en la variable transitoria *temp*.
- Segundo: introducimos el contenido de *lista[j]* en el lugar *lista[i]*. Ahora *lista[i]* y *lista[j]* contienen el mismo dato.
- Por último copiamos el contenido de la variable transitoria *temp* en *lista[j]*.

Podríamos pensar que el código siguiente nos resuelve el problema, pero presenta una dificultad al asignar valores a la lista de entrada:

```
> cambiar:=proc(a,i,j)
> local t:  t:=a[i]:
> a[i]:=a[j]:  a[j]:=t:  a; end:
> cambiar([1,2,3,4,5,6],1,6);

Error, (in cambiar) illegal use of a formal parameter
```

El problema viene de la asignación *a[i]:=a[j]*. Tenemos que escribirlo usando el comando *subsop* y estableciendo una copia de la lista inicial, así:

```
> Intercambiar:=proc(lista,i,j)
> local
> loc_lista,temp;
> loc_lista:=lista;
> temp:=loc_lista[i];
> loc_lista:=subsop(i=loc_lista[j],loc_lista);
> loc_lista:=subsop(j=temp,loc_lista);
> loc_lista;
> end:


> Intercambiar([3,2,3,4,5,10],1,6);
```

$$[10, 2, 3, 4, 5, 3]$$

```
> Intercambiar([1,6,3,4,7,6],2,5);
```

$$[1, 7, 3, 4, 6, 6]$$

Ordenación burbuja

Ahora ya podemos escribir el algoritmo de ordenación burbuja:

```
> Burbuja:=proc(a)
> local i,j,n,aloc;
> n:=nops(a);
> aloc:=a;
> for i from 1 to n-1 do
> for j from 1 to n-i do
> if (aloc[j]>aloc[j+1]) then aloc:=Intercambiar(aloc,j,j+1)
fi;
> od:
> od:  RETURN(aloc);
> end:
```

Probamos el algoritmo en un ejemplo:

```
> Burbuja([5,2,8,2,4,3,7,7,1]);
```

$$[1, 2, 2, 3, 4, 5, 7, 7, 8]$$

2.5.2 Ordenación selección

Necesitamos un procedimiento que calcule el lugar que ocupa el elemento de una lista que da el mínimo de dicha lista:

```
> minimo:=proc(a)
> local c,i,j:
> c:=a[1]:
> for i from 2 to nops(a) do
> if a[i]<c then c:=a[i] fi:
> od:
> j:=1:
> while a[j]<>c do
> j:=j+1
> od:
> j;
> end:

> minimo([1,2,3,4,5,0.5]);
```

$$6$$

```
> minimo([1,4,0.8,67,50]);
```

$$3$$

Ahora podemos llamar a este procedimiento para construir el algoritmo de selección. Hay que construir una lista cada vez más pequeña e ir calculando los mínimos para intercambiar los valores si es necesario. En lugar de construir listas cada vez más pequeñas, lo que vamos a hacer es ir poniendo el valor infinito en el correspondiente término de la lista de modo que, como infinito es mayor que cualquier número, se tendrá que a la hora de calcular mínimos este término no va a intervenir nunca.

```
> seleccion:=proc(a)
> local b,c,i,m:
> b:=a:
> c:=a:
> for i from 1 to nops(a) do
> m:=minimo(b):
> if b[i]<>b[m] then
> b:=Intercambiar(b,i,m):
> c:=Intercambiar(c,i,m):
> fi:
> b[i]:=infinity:
> od:  c;
> end:
> seleccion([2,1,3,6,5]);
```

$$[1, 2, 3, 5, 6]$$

```
> seleccion([2,5,4,3,1,0]);
```

$$[0, 1, 2, 3, 4, 5]$$

2.6 Ejercicios

En todos los ejercicios escribir el algoritmo que se pide, aplicarlo a dos ejemplos y estudiar su complejidad.

Ejercicio 43. Dada una lista de números enteros, construir un algoritmo que calcule el máximo de sus valores absolutos.

Ejercicio 44. Supongamos que quitamos de nuestro modelo computacional las operaciones básicas de producto y división de números enteros. Construir un algoritmo para, dados dos números enteros, calcular su producto.

Ejercicio 45. Construir un algoritmo para, dados dos enteros a y n construir la potencia a^n.

Ejercicio 46. Escribir un algoritmo para calcular la media aritmética de los elementos de una lista de números enteros.

Ejercicio 47. Escribir un algoritmo que dada una lista de números naturales y otro número natural indique si hay algún múltiplo de dicho número entre

los elementos de la lista, señale cuántos de estos múltiplos hay y en qué posiciones de la lista están.

Ejercicio 48. Distancia de Hamming. Dadas dos palabras de la misma longitud construidas con ceros y unos la distancia de Hamming entre ellas se define como el número de letras diferentes entre una y otra palabra (letras diferentes situadas en el mismo lugar): por ejemplo la distancia de Hamming entre la palabra 01 y la palabra 00 es uno porque la primera letra es igual y la segunda distinta. La distancia de Hamming entre 111 y 010 es dos. Construir un algoritmo que dadas dos palabras de ceros y unos nos devuelva su distancia de Hamming.

Ejercicio 49. Dada una lista ordenada de menor a mayor de números enteros y otro número entero construir un algoritmo que inserte dicho número en el lugar correspondiente de la lista inicial.

Ejercicio 50. Dadas dos listas $a_1, .., a_n$ y $b_1, ..., b_m$ de números enteros, construir un algoritmo para construir el producto cartesiano de las dos listas, esto es, el conjunto completo de los pares ordenados de la forma (a_i, b_j).

2.7 Ejercicios resueltos

Ejercicio 29. Lista de Ingresos: 5, 4, 4. Lista de Gastos: 4, 3.

Entrada: 5, 4, 4; 4, 3.
$I := 5$, $i := 1$
como $1 \leq 2$ entonces $i := 2$, $I := 5 + 4$
como $2 \leq 2$ entonces $i := 3$, $I := 13$
como $3 > 2$ sale del bucle.
$Ingresos := 13$

$G := 4$, $j := 1$
como $1 \leq 1$ entonces $j := 2$, $G := 4 + 3$
como $2 > 1$ sale del bucle.
$Gastos := 7$

$Balance := 13 - 7 = 6$

Ejercicio 30. El algoritmo termina porque el bucle se repite exactamente $n - 1$ veces.

Entrada: 1, 2, 0.

$a := 1$, $i := 1$
como $1 \leq 2$ entonces $i := 2$ y al ser $1 < 2$ entonces $a := 2$
como $2 \leq 2$ entonces $i := 3$ y al ser $2 < 0$ entonces $a := 2$
como $3 > 2$ sale del bucle.
Salida: $a = 2$.

Ejercicio 31. El algoritmo realiza la siguiente suma:

$$\Sigma_{i=1}^{n}\Sigma_{j=1}^{i} j.$$

Ejercicio 32. Lista: 1, 3, 5, 7, 9.

Búsqueda secuencial:
$i := 1$ (1 operación)
Como $1 \leq 5$ y $1 \neq 9$ entonces $i := 2$. (4 operaciones)
Como $2 \leq 5$ y $3 \neq 9$ entonces $i := 3$. (4 operaciones)
Como $3 \leq 5$ y $5 \neq 9$ entonces $i := 4$. (4 operaciones)
Como $4 \leq 5$ y $7 \neq 9$ entonces $i := 5$. (4 operaciones)
Como $5 \leq 5$ pero $9 = 9$ entonces sale del bucle. (2 operaciones)
Como $5 \leq 5$ entonces la salida es *Sí.* (2 operaciones)

Total de operaciones: 21.

Búsqueda binaria:
$i := 1$, $j := 5$ (2 operaciones)
Como $1 < 5$ y $1 \neq 9$ entonces $m := 3$. (4 operaciones)
Como $5 < 9$ entonces $i := 4$, $j = 5$. (4 operaciones)
Como $4 < 5$ entonces $m := 4$. (3 operaciones)
Como $7 < 9$ entonces $i := 5$, $j = 5$. (4 operaciones)
Como $5 = 5$ entonces sale del bucle. (1 operación)
Como $a_5 = 9$ entonces la salida es *Sí.* (2 operaciones)

Total de operaciones: 20.

Ejercicio 33. Lista: 1, 3, 2, 9, 5, 4.

Como $a_1 < a_2$ entonces no se hace ninguna operación.
Como $a_2 > a_3$ entonces $a_2 = 2$ y $a_3 = 3$.
Como $a_3 < a_4$ entonces no se hace ninguna operación.
Como $a_4 > a_5$ entonces $a_4 = 5$ y $a_5 = 9$.

Como $a_5 > a_6$ entonces $a_5 = 4$ y $a_6 = 9$.
De este modo la lista es ahora: 1, 2, 3, 5, 4, 9.
El algoritmo recomienza con la lista: 1, 2, 3, 5, 4.
Al recorrer esta nueva lista, el máximo, que es el 5, se coloca en el último lugar y la lista queda ordenada.

Ejercicio 34. Para aprovechar el algoritmo del máximo observamos que el mínimo de una lista a_1, ..., a_n es el máximo (cambiado de signo) de los valores de la lista $-a_1$, ..., $-a_n$. Por tanto si llamamos M al algoritmo que calcula el máximo tenemos que:

Entrada: a_1, ..., a_n.
For $i = 1$ to n
$\quad b_i := -a_i$
$m := -M(b_1, ...b_n)$
Salida: m.

Ejercicio 35. Sea la lista L:= 1, 3, 2, 9, 5, 4.
Como el mínimo de la lista L es $a_1 = 1$ entonces no se hace ninguna operación.
L :=3, 2, 9, 5, 4.
Como el mínimo de la lista L es a_3 entonces $a_2 = 2$ y $a_3 = 3$.
L :=3, 9, 5, 4.
Como el mínimo de la lista L es a_3 entonces no se hace ninguna operación.
L :=9, 5, 4.
Como el mínimo de la lista L es a_6 entonces $a_6 = 9$ y $a_4 = 4$.
L :=5, 9.
Como el mínimo de la lista L es a_5 entonces no se hace ninguna operación y el algoritmo ha terminado.
Salida: L :=1, 2, 3, 4, 5, 9.

Ejercicio 36. El peor de los casos es áquel en que la asignación $a := a_i$ se hace siempre, esto es, cuando la lista está ordenada de menor a mayor.

$$a_1 \leq a_2 \leq ... \leq a_n.$$

Ejercicio 37. Demostramos que $P(x) = a_0 + a_1x + ... + a_sx^s$ es del mismo orden que x^s. La observación fundamental es que un polinomio de grado

s tiende a $+\infty$ (cuando x tiende a $+\infty$) cuando su coeficiente principal (el coeficiente del monomio de mayor grado) es de signo positivo. Por tanto en tal caso su signo es positivo para todo x mayor o igual que un cierto n_0 (para el que no se aporta una expresión explícita).

El polinomio x^s domina a $P(x)$ ya que si tomamos $k > a_s$ se tiene que $kx^s - P(x) = (k - a_s)x^s + ... + (-a_0)$, que según la observación anterior es positivo para todo $x \geq n_0$.

$P(x)$ domina a x^s ya que si tomamos $k > 1/a_s$ entonces $kP(x) - x^s = (ka_s - 1)x^s + ... + ka_0$.

Ejercicio 38. Es una consecuencia directa de la fórmula del cambio de base:

$$log_a(n) = log_a(b)log_b(n).$$

Ejercicio 39. Adjuntamos las gráficas representadas con Maple, donde se observan las distintas relaciones de dominancia.

La función $l(n)$ domina a la función 1 porque $l(n) > 1$ para cada $n \geq 3$. En efecto, la función $g(n) = l(n) - 1$ es positiva para cada $n \geq 3$ puesto que $g(3) > 0$ y es creciente (su derivada $g'(n) = 1/n > 0$).

Se verifica que $n > l(n)$ para cada $n \in \mathbb{N}$. En efecto, sea la función $h(n) = n - l(n)$, se verifica que $h(1) > 0$ y que es siempre creciente ya que $h'(n) = 1 - 1/n > 0$.

Las demás dominancias son consecuencia de que:

$$r(n) = nl(n) - n \qquad r(3) > 0 \qquad r'(n) = l(n) > 0$$

$$s(n) = n^2 - nl(n) \qquad s(1) > 0 \qquad s'(n) = 2n - l(n) - 1 > 0$$

$$t(n) = e^n - n^2 \qquad t(1) > 0 \qquad t'(n) = e^n - 2n > 0.$$

Ejercicio 40. Si T domina a S entonces existen $n_0 \in \mathbb{N}$ y $k \in \mathbb{R}^+$ de modo que para cada $n \geq n_0$ se verifica

$$S(n) \leq kT(n).$$

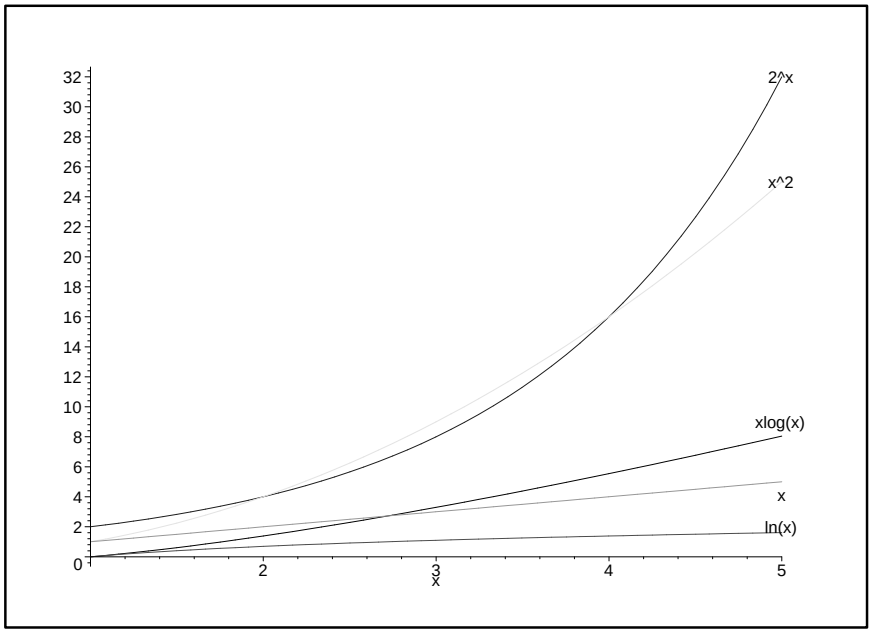


Figura 2.1: Gráficas con Maple

De este modo para cada $n \geq n_0$ se verifica

$$S(n) + T(n) \leq (k+1)T(n).$$

Lo que muestra que T domina a $S + T$.

La segunda cuestión es sólo observar que la función número de operaciones en el peor de los casos del algoritmo que concatena A_1 y A_2 es $T + S$.

Ejercicio 41. El algoritmo de ordenación burbuja es un algoritmo de complejidad cuadrática $O(n^2)$. En efecto, cada vez que se ejecuta el bucle en j el bucle en i hace una cantidad constante de operaciones y las hace exactamente $n - j$ veces, como

$$1 + 2 + 3 + .. + n = n(n+1)/2,$$

se tiene el resultado.

De igual manera el algoritmo de selección es cuadrático pues el bucle j se repite $n - 1$ veces y cada una de ellas llama a un algoritmo llamado min que es lineal.

Ejercicio 42. El algoritmo de ordenación burbuja es cuadrático y el algoritmo de búsqueda binaria es logarítmico, por tanto la concatenación de ambos es cuadrática.

Mientras, la búsqueda secuencial es lineal.

En los siguientes ejercicios, dejamos al estudiante la parte de aplicar los algoritmos a listas concretas.

Ejercicio 43. Definimos primero un algoritmo que calcula el valor absoluto de un entero.

Entrada: a
If $a < 0$ then $|a| := -a$ else $|a| := a$.
Salida: $|a|$

Observar que $-a = a * (-1)$.

Entonces aplicamos el algoritmo M que calcula el máximo a la lista de los valores absolutos.

Entrada: a_1, ..., a_n.
For $i = 1$ to n
$a_i := |a_i|$
Salida: $M(a_1, ..., a_n)$

El algoritmo del máximo es de complejidad lineal, el cálculo del valor absoluto es de complejidad constante ($T(n) = 3$). La reasignación de la lista tiene complejidad lineal ($T(n) = 5n + 3$). Por tanto el algoritmo es de complejidad lineal.

Ejercicio 44. El producto de dos enteros positivos no es más que sumar uno de ellos tantas veces como indica el otro. Como el producto es conmutativo y la suma una operación básica, elegimos el mínimo de los dos para decir cuántas veces hay que sumar. Multiplicamos entonces los valores absolutos y después atendemos al signo:

Entrada: a, b.
a_1 =mínimo$\{|a|, |b|\}$, a_2 :=máximo$\{|a|, |b|\}$.
$p := 0$, $i := 1$
While $i \leq a_1$
$p := p + a_2$
$i := i + 1$
If $a > 0, b < 0$ then $p := -p$ else If $a < 0, b > 0$ then $p := -p$
Salida: p

El tamaño de la entrada es aquí el valor absoluto del mínimo de ambos números (esto es $n = a_1$). El cálculo del mínimo y el máximo son algoritmos de complejidad lineal, en este caso se aplican a listas de longitud 2, luego realizan una cantidad constante de operaciones. En el algoritmo que estamos estudiando hay primero dos algoritmos constantes, después un bucle que se repite $n = a_1$ veces y que realiza cada vez que se ejecuta una cantidad constante de operaciones. Finalmente una asignación de signo, que se hace mediante una cantidad constante de operaciones. Así el algoritmo es de complejidad lineal.

Ejercicio 45.
Entrada a, n.
If $a = n = 0$ then p :=indeterminación
else
If $n \geq 0$ then

$p := 1$, $i := 1$
While $i \leq n$
$p := p * a$
$i := i + 1$
else
$p := 1$, $i := 1$
While $i \leq |n|$
$p := p * (1/a)$
$i := i + 1$
Salida:=p

El tamaño de la entrada es $|n|$ y la complejidad del algoritmo es lineal ya que el bucle se repite exactamente tantas veces como el tamaño de la entrada.

Ejercicio 46. Sumaremos todos los elementos de la lista y la suma la dividimos por el n úmero de elementos. El problema esencial en el diseño de este algoritmo es que no tenemos una operación básica que calcule divisiones y exprese el resultado como número decimal. Llamemos a esa operación $div(a, b)$ y el resultado es $\frac{a}{b}$ con, digamos, 2 decimales. Entonces, nuestro algoritmo es muy simple:

Datos de entrada: $a_1, a_2, ..., a_n$
$c := 0$
For $i := 1$ to n
$c := c + a_i$
$media := div(c, n)$

Implementemos ahora la operación $div(c, n)$ de la siguiente manera: $c1 := |c|/n$, $r1 := |c| \ mod \ n$
$c2 := (10 \cdot r1)/n$, $r2 := 10 \cdot r1 \ mod \ n$
$c3 := (10 \cdot r2)/n$, $r3 := 10 \cdot r2 \ mod \ n$

El algoritmo completo, para calcular la media con dos decimales, sería

Datos de entrada: $a_1, a_2, ..., a_n$
$c := 0$
For $i := 1$ to n
$c := c + a_i$
$c1 := |c|/n$, $r1 := |c| \ mod \ n$

$c2 := (10 \cdot r1)/n,\ r2 := 10 \cdot r1\ mod\ n$
$c3 := (10 \cdot r2)/n,\ r3 := 10 \cdot r2\ mod\ n$
If $c > 0$ then media:$=c1'c2c3$ else media:$=-c1'c2c3$

Complejidad. Hay:
1) Una asignación,
2) Un bucle que se repite n veces y en cada paso se hace una suma y una asignación. Total: $2n$ operaciones.
3) El cálculo de un cociente y un resto, dos multiplicaciones y dos divisiones, el cálculo de dos restos y 6 asignaciones, admás de una comparación y posiblemente un cambio de signo. Total: 16 operaciones.
Entonces, $T(n) := 2n + 12$. Complejidad lineal.
Si calculáramos la media con más decimales, el número de operaciones sería mayor pero la complejidad seguiría siendo lineal.

Ejercicio 47.
Datos de entrada: $a_1, a_2, ..., a_n, a$
$i := 0\ ,\ j := 0$
While $i \leq n - 1$
$i := i + 1$
$b := a_i\ mod\ a$
If $b := 0$ then $j := j + 1$, $d_j := a_i$ y $e_j = i$
Salida: Hay j múltiplos de a. Están en las posiciones $e_1, ..., e_j$ y sus valores son $d_1, ..., d_j$.

Calculemos la complejidad. Hay:
1) dos asignaciones y el cálculo de $n - 1$. Total: 3 operaciones.
2) un bucle que se repite n veces. Dentro del bucle hay: dos asignaciones, el cálculo de un resto y una suma, una condicional con una condición y, en el peor de los casos, tres asignaciones y una suma. Total: $9n$.
3) Del bucle se sale cuando la comparación es falsa. Total: Una operación.
$T(n) = 3 + 9n + 1 = 9n + 4$.
La complejidad es lineal.

Ejercicio 48. Sean $a_1, a_2, ..., a_n$ las cifras en bits de uno de los números y $b_1, b_2, ..., b_n$ las del otro.

Datos de entrada: $a_1, a_2, a_3, ..., a_n\ ;\ b_1, b_2, ..., b_n$
$c := 0$
For $i = 1$ to n

If $a_i \neq b_i$ then $c := c + 1$
Salida: c

Calculamos la complejidad. Hay:
1) Una asignación.
2) Un bucle que se repite n veces. Dentro de él, y en el peor de los casos, hay una comparación, una suma y una asignación. Total: $3n$ operaciones.
Entonces $T(2n) = 3n + 1$ operaciones (notemos que hay $2n$ datos de entrada). Por tanto, $T(n) = \frac{3}{2}n + 1$.

La complejidad es lineal.

Ejercicio 49. La idea de este algoritmo es leer la lista hasta que lleguemos al punto en el que debemos insertar el número dado a. Almacenar en valor de la lista en ese punto en una variable auxiliar (para no perderlo), introducir el valor a, y a partir de allí colocar los demás elementos de la lista. Estos últimos elementos se pueden rellenar desde el final de la lista hasta el punto en el que está a (probar otras alternativas para comprobar que es esta la que mejor funciona).

Datos de entrada: $a_1, a_2, a_3, ..., a_n; a$
$i := 0$, $p := 0$
While $i \leq n - 1$ and $p \geq 0$
$i := i + 1$
$p := a - a_i$
$c := a_i$, $a_i := a$
For $j = 0$ to $n - i - 1$
$a_{n+1-j} := a_{n-j}$
$a_{i+1} := c$
Salida: $a_1, a_{2,.....,}a_{n+1}$

Complejidad. Hay:
1) Dos asignaciones y el cálculo de $n - 1$. Total: 3 operaciones.
2) Un bucle que se repite, en el peor de los casos, n veces. En cada pasada del bucle tendremos dos comparaciones, una asignación y una suma, otra asignación y una diferencia. Total: $6n$.
3) Del bucle se sale cuando una de las comparaciones sea falsa, lo que da lugar a otra operación.
4) Dos asignaciones. Total: 2 operaciones.
5) El cálculo de $n - i - 1$. 2 operaciones

6) Un bucle que, en el peor de los casos, se repite n veces y en el que hay una asignación y el cálculo de los subíndices. Total: $4n$ operaciones.

7) Una asignación y el cálculo del subíndice.

Notemos que es imposible que se ejecuten todas las operaciones del bucle de 2) y del bucle de 6) ya que, en el valor del índice en el que termina uno termina el otro. En el peor de los casos, es el bucle 2) el que se ejecuta completamente.

En consecuencia la complejidad es lineal.

Ejercicio 50.

Datos de entrada: $a_1, a_2, ..., a_n; b_1, b_2, ..., b_n$

For $i := 1$ to n

    For $j := 1$ to n

        $c_{ij} := (a_i, b_j)$

Salida: $c_{11}, c_{12}, ..., c_{21}, c_{22}, ..., c_{n1}, ..., c_{nn}$

Complejidad. Hay dos bucles anidados. El primero se repite n veces y el segundo otras n. En este segundo bucle hay una asignación. Por tanto, $T(2n) = n^2$. Por tanto, $T(n) = \frac{1}{4}n^2$. Complejidad cuadrática.

Capítulo 3

Aritmética modular

En este capítulo se estudiarán los números enteros, sus propiedades y operaciones, introduciendo también las nociones básicas de la aritmética modular. Se pretende que al final del capítulo el alumno:

- Conozca las propiedades básicas de los números enteros (operaciones, factorización).
- Pueda calcular el mcd de dos números enteros por el algoritmo de Euclides.
- Realice con soltura operaciones en la aritmética modular.
- Resuelva congruencias lineales y sistemas de congruencias.

3.1 Los números naturales y los números enteros

Los números naturales y los números enteros son objetos conocidos. El lector está familiarizado con dichos objetos y con sus operaciones:

$$\mathbb{N} = \{1, 2, 3, 4....\}$$

$$\mathbb{Z} = \{0, 1, -1, 2, -2, ...\}.$$

Son, como decimos, objetos con los que estamos habituados a trabajar y de hecho en el capítulo 1 hemos usado una de sus propiedades fundamentales

para presentar una técnica potente de demostración que es el principio de inducción.

Nuestro primer objetivo es introducir de forma rigurosa estos objetos. Existen varias maneras de hacerlo, una de ellas (la que vamos a elegir) es la axiomática. Consiste en caracterizar el conjunto $\mathbb{N}$ de los números naturales por algunas de sus propiedades que se imponen como axiomas, de manera que cualquier otra propiedad se deduce (usando las reglas de la lógica) de estos axiomas.

Podemos usar por ejemplo la caracterización de $\mathbb{N}$ como cualquier conjunto que verifique los llamados **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. Se denota $s(n)$. Es un elemento de $\mathbb{N}$ y verifica que $s(n) \neq 1$ para cada $n \in \mathbb{N}$.
- Si $s(n) = s(m)$ entonces $n = m$.
- **Principio de inducción**: si un subconjunto $A \subset \mathbb{N}$ verifica que $1 \in A$ y que $n \in A$ implica $s(n) \in A$, entonces se tiene que $A = \mathbb{N}$. (Esta propiedad es la que se usó en el capítulo 1.)

Observación 3.1.1 *Los axiomas de Peano segundo y tercero son equivalentes a la existencia de una aplicación inyectiva $s : \mathbb{N} \to \mathbb{N}$ de modo que el 1 no está en la imagen de s.*

En este conjunto se puede definir una primera operación denominada **suma**, de manera que dados dos naturales cualesquiera $a, b \in \mathbb{N}$ se puede construir el número natural $a + b \in \mathbb{N}$ que es la suma de los dos. Esta operación se define por las reglas:

- Para cada $a \in \mathbb{N}$ se define $a + 1 = s(a)$.
- Para cada $a, b \in \mathbb{N}$ se define $a + s(b) = s(a + b)$.

Es consecuencia del principio de inducción que así hemos definido cualquier suma $a + b$ con $a, b \in \mathbb{N}$. En efecto, sea $a \in \mathbb{N}$ un número natural. Sea A el conjunto de los números naturales $b \in \mathbb{N}$ para los que $a+b$ está definido.

Vamos a demostrar por inducción que $A = \mathbb{N}$ con lo que las reglas anteriores permiten definir $a + b$ para cada par de números naturales $a, b \in \mathbb{N}$.

Base de inducción. Se tiene que $1 \in A$ ya que la primera regla indica que $a + 1 = s(a)$.

Paso de inducción. Debemos demostrar que si $b \in A$ entonces $s(b) \in A$. Como por hipótesis de inducción $b \in A$ entonces $a + b$ está definido. Usando la regla segunda sabemos que $a + s(b) = s(a + b)$ con lo que $a + s(b)$ está definido y por tanto $s(b) \in A$ como queríamos demostrar.

Estudiemos la estructura algebraica del par $(\mathbb{N}, +)$, es decir las propiedades que verifican los números naturales con esta operación de suma. Se tienen las siguientes:

i) propiedad **asociativa**: $(a + b) + c = a + (b + c)$ para cualesquiera $a, b, c \in \mathbb{N}$,

ii) propiedad **conmutativa**: para cualquier pareja $a, b \in \mathbb{N}$ se tiene que $a + b = b + a$.

Ejercicio 51 *Demostrar (por inducción sobre c) la propiedad asociativa, usando las reglas de la definición de la suma.*

También se puede definir una segunda operación en $\mathbb{N}$ que es el **producto**, asignando a cada par $a, b \in \mathbb{N}$ el producto de ambos $a \cdot b$ (ó $a \times b$ ó simplemente ab, usaremos las tres notaciones indistintamente). De la misma manera que en el caso de la suma, es una consecuencia del principio de inducción que las reglas que se presentan a continuación son suficientes para definir el producto de dos números naturales $a, b \in \mathbb{N}$.

- Para cada $a \in \mathbb{N}$ se tiene que $a \cdot 1 = a$;
- Para cada $a, b \in \mathbb{N}$ se tiene que $a \cdot s(b) = a \cdot b + a$.

Ejercicio 52 *Demostrar, usando el principio de inducción, que las reglas anteriores permiten definir el producto de cualquier par de números naturales.*

Los números naturales con esta operación de producto, $(\mathbb{N}, \times)$, tienen las siguientes propiedades, que se pueden demostrar usando el principio de inducción y las reglas de definición del producto:

1) propiedad **asociativa**: $(a \times b) \times c = a \times (b \times c)$ para cualesquiera $a, b, c \in \mathbb{N}$,

2) existencia de **elemento neutro**: existe $1 \in \mathbb{N}$ tal que para cualquier número natural $a \in \mathbb{N}$ se tiene que $a \times 1 = 1 \times a = a$,

3) **propiedad conmutativa**: para cualquier pareja $a, b \in \mathbb{N}$ se tiene que $a \times b = b \times a$.

Las dos operaciones, la suma y el producto, quedan relacionadas por la propiedad distributiva:

4) **propiedad distributiva**: para cualesquiera $a, b, c \in \mathbb{N}$ se tiene que $a \times (b + c) = a \times b + a \times c$.

Una vez definidos los números naturales, los números enteros se pueden construir a partir de los naturales de la siguiente manera. Tomamos dos copias del conjunto de los números naturales, marcando los elementos de una de ellas con un signo menos. Llamamos $\mathbb{N}^+ = \{1, 2, 3, 4, ...\}$ a la primera copia y $\mathbb{N}^- = \{-1, -2, -3, -4, ...\}$ a la segunda. Los números enteros son la unión de $\mathbb{N}^+$, $\mathbb{N}^-$ y un conjunto formado por un elemento distinguido, que escribimos como $\{0\}$:

$$\mathbb{Z} = \{1, 2, 3, ...\} \cup \{0\} \cup \{-1, -2, -3, ...\} = \mathbb{N}^+ \cup \{0\} \cup \mathbb{N}^-.$$

Tanto $\mathbb{N}^+$ como $\mathbb{N}^-$ vienen dotados de su respectivo concepto de *siguiente*, que denotamos respectivamente s^+ y s^-. Así se tiene, por ejemplo, que $s^+(5) = 6$ y que $s^-(-5) = -6$. Estas dos nociones permiten definir en el conjunto de los números enteros un concepto de *siguiente* denotado como s y un concepto de *anterior*, denotado como a, de modo que:

- Para cada $n \in \mathbb{N}^+$ se tiene que $s(n) = s^+(n)$,
- $s(0) = 1$,
- $s(-1) = 0$,
- para cada $n \in \mathbb{N}^-$, $n \neq -1$, existe $m \in \mathbb{N}^-$ de modo que $n = s^-(m)$. Entonces $s(n) = m$.
- Para cada $n \in \mathbb{N}^-$ se tiene que $a(n) = s^-(n)$,
- $a(0) = -1$,
- $a(1) = 0$,

- para cada $n \in \mathbb{N}^+$, $n \neq 1$ existe $m \in \mathbb{N}^+$ de modo que $n = s^+(m)$. Entonces $a(n) = m$.

Notación Usaremos el símbolo $\emptyset$ para representar el conjunto vacío.

Y como ya señalamos cuando hablamos de la inducción estructural se verifica que: *si $A \subset \mathbb{Z}$ verifica que $A \neq \emptyset$ y para cada $n \in A$ se tiene que $s(n) \in A$ y $a(n) \in A$ entonces $A = \mathbb{Z}$.*

La operación **suma de números enteros** se construye apoyándose en la suma de los números naturales, de modo que las siguientes reglas son suficientes:

- Para cada $n \in \mathbb{Z}$ se define $n + 1 = s(n)$.
- Para cada $n \in \mathbb{Z}$ se define $n + (-1) = a(n)$.
- Para cada $n, m \in \mathbb{Z}$ se define $n + s(m) = s(n + m)$.
- Para cada $n, m \in \mathbb{Z}$ se define $n + a(m) = a(n + m)$.

Ejercicio 53 *Demostrar por inducción estructural que con las reglas anteriores se define $n + m$ para cada $n, m \in \mathbb{Z}$.*

Y los números enteros con la operación de suma así construida, $(\mathbb{Z}, +)$, verifican las siguientes propiedades, que se pueden demostrar usando las reglas de definición de la suma y el principio de inducción estructural:

i) propiedad **asociativa**: $(a + b) + c = a + (b + c)$ para cualesquiera $a, b, c \in \mathbb{Z}$,

ii) propiedad **conmutativa**: para cualquier pareja $a, b \in \mathbb{Z}$ se tiene que $a + b = b + a$.

iii) existencia de **elemento neutro**: existe un entero $0 \in \mathbb{Z}$ tal que para cualquier número entero $a \in \mathbb{Z}$ se tiene que $a + 0 = 0 + a = a$,

iv) existencia de **elemento inverso**: para cualquier entero $a \in \mathbb{Z}$ existe otro entero denominado $-a \in \mathbb{Z}$ tal que $a + (-a) = (-a) + a = 0$.

Notación. En la propiedad iv) para cada $a \in \mathbb{Z}$ denotamos con $-a$ el inverso de a. Se puede demostrar que: si $a = 0$ entonces $-a = 0$; que si $a \in \mathbb{N}^+$ entonces $-a$ es su correspondiente pareja marcada en $\mathbb{N}^-$ y que si $a \in \mathbb{N}^-$ entonces $-a$ es su correspondiente pareja en $\mathbb{N}^+$ quitándole la marca. Por

ejemplo $-(-5) = 5$, $-0 = 0$, $-(5) = -5$. Escribiremos $a - b$ para simplificar la expresión $a + (-b)$.

Decimos entonces que $(\mathbb{Z}, +)$ es un **grupo conmutativo** (cualquier conjunto con una operación verificando las propiedades i) a iv) es un grupo conmutativo). Observamos que al construir los números enteros hemos ampliado el conjunto de los números naturales para que la operación resta tenga siempre sentido. Por ejemplo $5 - 7$ es una diferencia de números naturales cuyo resultado no es un número natural pero sí un número entero, $-2 \in \mathbb{Z}$.

De la misma manera que con la operación de la suma, atendiendo a la casuística del signo, se puede definir otra operación sobre los números enteros que es el producto: dados dos números enteros $a, b \in \mathbb{Z}$ se puede construir el producto de ambos $a \times b \in \mathbb{Z}$.

Ejercicio 54 *Describir un conjunto de reglas suficientes para definir el producto ab de cualquier par de números enteros $a, b \in \mathbb{Z}$, atendiendo a las distintas posibilidades para los signos de a y b.*

En esta operación se tienen las mismas propiedades que en $\mathbb{N}$:

1) propiedad **asociativa**: $(a \times b) \times c = a \times (b \times c)$ para cualesquiera $a, b, c \in \mathbb{Z}$,
2) existencia de **elemento neutro**: existe un entero $1 \in \mathbb{Z}$ tal que para cualquier número entero $a \times 1 = 1 \times a = a$,
3) **propiedad conmutativa**: para cualquier pareja de enteros $a, b \in \mathbb{Z}$ se tiene que $a \times b = b \times a$.

Pero es claro que, en general, no existe elemento inverso, por ejemplo, el inverso del número entero 2 debería ser el número $1/2$ que no es un número entero. (La sucesiva ampliación de $\mathbb{Z}$ para que la división sea una operación que tenga sentido es construir el conjunto de los números racionales $\mathbb{Q}$.)

Las dos operaciones, la suma y el producto, quedan, igual que en $\mathbb{N}$, relacionadas por la propiedad distributiva:

4) **propiedad distributiva**: para cualesquiera $a, b, c \in \mathbb{Z}$ se tiene que $a \times (b + c) = a \times b + a \times c$.

Entonces decimos que $(\mathbb{Z}, +, \times)$ es un **anillo conmutativo** (de nuevo cualquier conjunto con dos operaciones verificando las propiedades i) a iv) y 1) a 4) es un anillo conmutativo).

El estudio de las estructuras algebraicas es importante y abre todo el campo de las matemáticas conocido como **álgebra**. Capta la estructura profunda de los conjuntos dotados de operaciones y permite describir sus propiedades y estudiar características generales. El álgebra es entonces el estudio de las *reglas del juego*, como en esos juegos de mesa que se presentan con distinto aspecto pero al leer sus reglas descubrimos un juego ya conocido.

Ejercicio 55 *Comprobar que las matrices de tamaño* 2×2 *con coeficientes enteros son un anillo no conmutativo (esto es, el producto no es conmutativo) con la suma y el producto habituales de matrices.*

3.2 Teorema de la división

Aunque en el anillo de los números enteros no se puede dividir, porque los cocientes no son enteros, sí se puede hacer una división entera, obteniéndose un cociente y un resto. Esta división nos va a permitir por un lado construir un algoritmo para el cálculo del máximo común divisor de dos números sin necesidad de factorizarlos y por otro establecer una relación de equivalencia, la **congruencia módulo un entero**, con interesantes utilidades para trabajar en anillos parecidos a los enteros pero con una cantidad finita de elementos.

Antes de enunciar y demostrar el teorema del resto debemos detenernos un momento a reflexionar sobre el orden que presentan los números naturales y los números enteros. Aunque volveremos sobre ellas en el capítulo 6, definamos lo que es una relación de orden.

Definición 3.2.1 *Sea* A *un conjunto, se define una* **relación** *en el conjunto* A *como un subconjunto* R *del producto cartesiano* $A \times A = \{(a, b) : a, b \in A\}$*, de modo que dos elementos* $a, b \in A$ *están relacionados si y solamente si la pareja* (a, b) *está en* R*.*

Ejemplos 3.2.2 *Podemos definir una relación en el conjunto finito* $A = \{1, 2, 3, 4\}$ *como*

$$R = \{(1, 1), (2, 1), (1, 2)\},$$

es decir, el 1 se relaciona con el 1 (a veces escrito $1R1$*) y con el 2 (*$1R2$*) y el 2 se relaciona con el 1 (2R1).*

Una relación habitual es la que define los números racionales. Tomemos el conjunto de las fracciones $F = \{p/q : p, q \in \mathbb{Z}, q \neq 0\}$ *y definimos la relación* R *de modo que* p/q *se relaciona con* r/s *si y solamente si* $ps = qr$ *(producto de medios es igual a producto de extremos).*

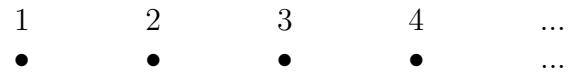
Definición 3.2.3 *Sea* A *un conjunto, una* **relación** R *en* A *se dice que es* **de orden** *si verifica las propiedades:*

i) **Reflexiva***: a se relaciona con a para cualquier a en A, esto es* $(a, a) \in R$ *para cada a de A;*

ii) **Antisimétrica***: si* $(a, b) \in R$ *y* $(b, a) \in R$ *entonces* $a = b$*;*

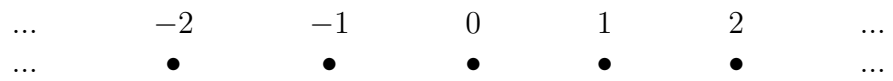
iii) **Transitiva***: si a se relaciona con b y b se relaciona con c entonces a se relaciona con c, es decir si* $(a, b) \in R$ *y* $(b, c) \in R$ *entonces* $(a, c) \in R$*.*

En el conjunto de los números naturales hay una relación de orden denotada con el signo $\geq$, por la que, por ejemplo $5 \geq 3$ y $100 \geq 34$ y que se puede definir de la siguiente manera: dados $a, b \in \mathbb{N}$ se dice que $b \geq a$ si o bien $b = a$ o bien existe un número natural c tal que $b = a + c$. Este ordenamiento de los números naturales se suele representar sobre una semirrecta (hacia la derecha los valores mayores):



Consideremos $\mathbb{N}^+ \subset \mathbb{Z}$ (a veces diremos sencillamente $\mathbb{N} \subset \mathbb{Z}$) así ordenado según el orden anteriormente descrito. Vamos a extender dicho orden a los números enteros, de manera que se respete el orden que tenemos en $\mathbb{N}^+$. La relación de orden se definirá de la siguiente manera: sean $a, b \in \mathbb{Z}$ entonces $a \geq b$ si y solamente si $a - b \in \mathbb{N}^+ \cup \{0\}$.

Este orden sobre el conjunto de los números enteros se suele representar en una recta (hacia la derecha los valores mayores):



Un teorema fundamental sobre el conjunto $\mathbb{N}$ es el siguiente:

Teorema 3.2.4 *Cada subconjunto no vacío* A *de* $\mathbb{N}$ *tiene un elemento mínimo* $m \in A$ *de manera que para cada elemento* $n \in A$ *se tiene que* $n \geq m$*. Dicho elemento* m *es además único.*

Demostración. Supongamos que A no tiene mínimo, entonces demostramos por inducción completa que $\mathbb{N} - A = \mathbb{N}$, lo que quiere decir que A es el conjunto vacío.

Como A no tiene mínimo entonces $1 \notin A$, esto es, $1 \in \mathbb{N} - A$. Se tiene la base de inducción.

Ahora bien si $1 \notin A$, $2 \notin A$, ..., $n \notin A$ entonces $s(n) \notin A$ pues sería mínimo. De este modo se tiene el paso de inducción y por tanto $A = \emptyset$.

Hemos demostrado la existencia de un mínimo, veamos su unicidad. Si m y m' son mínimos entonces se tiene que $m \leq m'$ y $m' \leq m$ por lo que $m = m'$. Por tanto el mínimo es único.

Notación. Si $a \geq b$ y $a \neq b$ escribiremos $a > b$. La expresión $b \leq a$ es otra manera de escribir $a \geq b$.

Observación 3.2.5 *El* 1 *y el* -1 *son los únicos números enteros que tienen inverso para el producto,* $1 \times 1 = (-1) \times (-1) = 1$. *Estos dos números se denominan* **unidades** *del anillo de los enteros. A partir de ahora trabajaremos siempre con números positivos. Para cambiar el signo a un número basta multiplicar por -1, un proceso fácilmente reversible. Esto indica que trabajar con números positivos para la división no es una grave restricción.*

Ya estamos en condiciones de enunciar y demostrar el teorema del resto.

Teorema 3.2.6 *Sean* $a, b \in \mathbb{Z}$ *con* $a \geq 0$ *y* $b > 0$, *entonces existen dos enteros* q *(cociente) y* r *(*resto*) únicos tales que* $q \geq 0$, $0 \leq r < b$ *y se tiene la expresión*

$$a = bq + r$$

esto es, dividendo es igual a divisor por cociente más resto.

Demostración. Comenzamos demostrando la existencia de q y r.

Si $a < b$ entonces $q = 0$ y $r = a$ verifican las hipótesis.

Si $a = b$ entonces $q = 1$ y $r = 0$ verifican las hipótesis.

Si $a > b$ entonces tomamos el conjunto $A = \{a - nb : \ n \in \mathbb{N}\} \cap (\mathbb{N}^+ \cup \{0\})$ que es un subconjunto no vacío de $\mathbb{N}^+ \cup \{0\}$. Por tanto, o $0 \in A$ y es un elemento mínimo, o $0 \notin A$ y el teorema anterior garantiza que A tiene un elemento mínimo. Denominamos $r \in A$ al mínimo. De este modo existe $q \in \mathbb{N}$ tal que

$$a - qb = r$$

o lo que es lo mismo

$$a = qb + r.$$

Para demostrar que $r < b$ se razona por reducción al absurdo. Si $r \geq b$ entonces $r = b + l$ con $l \in \mathbb{N}^+ \cup \{0\}$, de modo que

$$a = (q + 1)b + l$$

entonces l es un elemento de A estrictamente menor que r en contradicción con que r es el mínimo de A.

La unicidad es consecuencia en los primeros casos de la desigualdad

$$a \leq b$$

y en el último de la unicidad del mínimo y del hecho de que dos elementos de A difieren en un múltiplo de b.

3.3 Divisibilidad, mcd y factorización

Como ya hemos señalado, la no existencia de inverso para el producto de números enteros hace que la operación de división no se pueda realizar siempre (igual que no se puede efectuar siempre la resta en los números naturales). Aunque se puede dividir 4 entre 2 y obtener un número entero, sin embargo, la división 5 entre 3 no se puede efectuar (sólo podemos obtener un cociente y un resto según el teorema antes demostrado). Esto da lugar a una noción interesante que es la de **divisibilidad**.

3.3.1 Definiciones básicas

Definición 3.3.1 *Sean a, b dos números enteros, se dice que a* **divide a** *b y se escribe $a|b$ si existe un número entero c tal que $b = a \times c$. También decimos que b es* **múltiplo** *de a o que a es un* **factor** *de b.*

Es decir, en el caso divisible, se puede efectuar la división b entre a y se obtiene un número entero, esto es, al hacer la división entera entre b y a el resto es 0.

Ejemplo 3.3.2 *El número entero* 12 *divide a* 60 *porque* $60 = 5 \times 12$ *pero* 12 *no divide a 34.*

Observación 3.3.3 *Todo número entero $a \in \mathbb{Z}$ es divisible por a, por $-a$, por 1 y por -1.*

Observación 3.3.4 *Cada divisor d de un número entero no nulo $a \in \mathbb{Z}$ verifica $|d| \leq |a|$, donde $|x|$ es el valor absoluto, esto es, para cada $x \in \mathbb{Z}$ se tiene $|x| = x$ si $x \geq 0$ y $|x| = -x$ si $x < 0$.*

Definición 3.3.5 *Un entero positivo $a \neq 1$ se dice que es un* **número primo** *si sus únicos factores positivos son a y 1. Si a tiene un factor positivo distinto de a ó de 1 se dice que es* **compuesto**.

Ejemplo 3.3.6 *Los enteros 5, 13 y 19 son números primos, 125 y 258 son compuestos.*

3.3.2 Algoritmo de Euclides para calcular el mcd

Como se deduce de las dos observaciones del párrafo anterior el conjunto de divisores positivos de un número natural a es un conjunto no vacío (ya que al menos a y 1 son divisores de a) y finito (ya que está contenido en el conjunto $\{-a, -a+1, ..., a-1, a\}$) por lo que tiene sentido la siguiente definición.

Definición 3.3.7 *Dados dos números enteros positivos a y b se define el máximo común divisor de a y b y se escribe $mcd(a, b)$ como el mayor de los divisores comunes de a y b.*

Vamos a desarrollar un algoritmo para calcular el mcd de dos números. Es un algoritmo clásico conocido como algoritmo de Euclides. Necesitaremos para esta construcción el siguiente lema.

Lema 3.3.8 *Dados dos números enteros positivos a y b, $a > b$ entonces $mcd(a, b) = mcd(b, r)$ donde r es el resto de dividir a entre b.*

Demostración. Vamos a demostrar que el conjunto de divisores comunes a a y b es igual que el conjunto de divisores comunes a b y a r. Esto implica que $mcd(a, b) = mcd(b, r)$.

En efecto, si p divide a a y a b entonces existen números enteros $s, t \in \mathbb{Z}$ tales que

$$a = ps \qquad b = pt,$$

de modo que la igualdad que da el teorema del resto $a = qb + r$ se puede escribir como

$$r = a - qb = ps - qpt = p(s - qt)$$

lo que demuestra que p divide a a r.

Recíprocamente si p divide a b y a r entonces

$$b = pt \qquad r = pu,$$

con lo cual

$$a = qb + r = qpt + pu = p(qt + u),$$

lo que demuestra que p divide a a. Esto finaliza la demostración del lema.

Por tanto si dividimos a entre b el problema de calcular el $mcd(a, b)$ se reduce al cálculo del $mcd(b, r)$ que son ahora números más pequeños por el teorema de la división. Nuevamente podemos dividir b entre r y obtener un resto r_1, por lo que el problema es calcular el $mcd(r, r_1)$. Así sucesivamente se van calculando r_2, r_3... que son números enteros positivos que van decreciendo ($r_1 > r_2 > r_3$...). Llegará entonces un momento en el que $r_n = 0$, esto implica que r_{n-1} divide a r_{n-2} (el resto de la división es $r_n = 0$) y entonces el $mcd(a, b) = mcd(r_{n-2}, r_{n-1}) = r_{n-1}$.

Ejemplo 3.3.9 *Calcular el máximo común divisor de 125 y 8872.*

Dividimos 8872 entre 125 para obtener:

$$8872 = 125 \times 70 + 122$$

Lo que indica que:

$$mcd(8872, 125) = mcd(125, 122).$$

Dividimos 125 entre 122:

$$125 = 122 \times 1 + 3.$$

Por tanto:

$$mcd(125, 122) = mcd(122, 3)$$

De nuevo:

$$122 = 3 \times 40 + 2,$$

y por tanto:

$$mcd(122, 3) = mcd(3, 2).$$

Otra vez:

$$3 = 2 \times 1 + 1,$$

lo que significa:

$$mcd(3, 2) = mcd(2, 1).$$

El resto de dividir por 1 es siempre 0 por lo que el $mcd(8872, 125) = 1$. Esto se suele indicar como que 8872 y 125 son *primos entre sí* (o *relativamente primos*).

Definición 3.3.10 *Si dos números enteros positivos $a, b \in \mathbb{Z}$ verifican que $mcd(a, b) = 1$ se dicen* **primos entre sí** *(o* **relativamente primos***).*

Ejercicio 56 *Calcula por el algoritmo de Euclides el $mcd(247, 9981)$.*

Notación. Al resto de dividir a entre b lo denotaremos $a \ mod \ b$ (como señalamos en el capítulo anterior).

Ejercicio 57 *Escribe en pseudocódigo el algoritmo de Euclides.*

Lema 3.3.11 *(de Bezout) Sean $a, b \in \mathbb{Z}$ números enteros positivos, existen dos enteros s y t tales que*

$$mcd(a, b) = sa + tb.$$

Demostración. Es una consecuencia del algoritmo de Euclides, recordemos que, suponiendo que $a \geq b$, teníamos una expresión de la forma:

$$\begin{aligned} a &= q_0 b + r_0 \\ b &= q_1 r_0 + r_1 \\ r_0 &= q_2 r_1 + r_2 \\ &\dots \end{aligned}$$

$$r_{n-3} = q_{n-1}r_{n-2} + r_{n-1}$$
$$r_{n-2} = q_n r_{n-1}$$

donde $r_{n-1} = mcd(a, b)$.

Vamos a demostrar que para cada r_i $(i \geq 0)$ existen enteros s_i y t_i de modo que:

$$r_i = s_i a + t_i b.$$

Esto concluye el lema porque $r_{n-1} = mcd(a, b)$.

La demostración que vamos a presentar es, de alguna manera, constructiva. Si despejamos r_{n-1} en la penúltima igualdad, expresamos el $mcd(a, b)$ como combinación lineal entera de r_{n-3} y r_{n-2} (esto quiere decir que existen dos números enteros A y B de modo que $mcd(a, b) = Ar_{n-3} + Br_{n-2}$). Si vamos despejando r_{n-2} en la siguiente igualdad, r_{n-3} en la anterior... y recurrentemente vamos sustituyendo en las expresiones anteriores, al final expresamos $mcd(a, b)$ como combinación lineal entera de a y b que es justamente lo que queremos.

Formalizamos el razonamiento por inducción completa en i.

El caso $i = 0$ es cierto puesto que:

$$r_0 = a - q_0 b.$$

Por tanto $s_0 = 1$ y $t_0 = -q_0$.

El caso $i = 1$ es también cierto puesto que $r_1 = b - q_1 r_0$. De este modo, sustituyendo la igualdad anterior tenemos:

$$r_1 = b - q_1(a - q_0 b) = (1 + q_1 q_0)b - q_1 a.$$

Por tanto $s_1 = -q_1$ y $t_1 = 1 + q_1 q_0$.

Veamos ahora que si el lema es cierto para cualquier valor estrictamente menor que i también lo es para i.

En efecto, tenemos:

$$r_i = r_{i-2} - q_i r_{i-1}.$$

Usando la hipótesis de inducción para r_{i-1} y r_{i-2} se obtiene:

$$r_i = (s_{i-2}a + t_{i-2}b) - q_i(s_{i-1}a + t_{i-1}b).$$

Por tanto

$$a_i = s_{i-2} - q_i s_{i-1}$$

y

$$b_i = t_{i-2} - q_i t_{i-1}$$

hacen que se verifique el lema.

Ejemplo 3.3.12 *Expresar el* $mcd(35, 20)$ *como combinación lineal de 35 y 20.*

Las expresiones del algoritmo de Euclides:

$$35 = 20 \times 1 + 15$$
$$20 = 15 \times 1 + 5$$

Entonces, despejando el 5 en la segunda igualdad obtenemos $5 = 20 - 15 \times 1$. Despejando el 15 en la primera igualdad $15 = 35 - 20 \times 1$. Sustituyendo el segundo valor en la primera igualdad $5 = 20 - (35 - 20 \times 1)$, es decir:

$$5 = 2 \times 20 + (-1) \times 35.$$

Corolario 3.3.13 *Sean* $a, b \in \mathbb{Z}$ *dos números no negativos. Si* $1 < p < ab$ *es un entero positivo que divide al producto* ab *y* $mcd(a, p) = 1$ *entonces* p *divide a* b*.*

Demostración. Como $mcd(a, p) = 1$, del lema de Bezout, se deduce que existen enteros $s, t \in \mathbb{Z}$ de modo que:

$$1 = sa + tp.$$

Multiplicando por b se obtiene que:

$$b = sab + tpb.$$

Como p divide a ab se tiene que existe $c \in \mathbb{Z}$ tal que $ab = pc$, de este modo:

$$b = spc + tpb = p(sc + tb).$$

Y por tanto p divide a b.

Corolario 3.3.14 *Sean* p, p_1, ..., p_n *números primos. Si* p *divide al producto* $p_1 ... p_n$ *entonces existe* i*, tal que* $1 \leq i \leq n$ *y* $p = p_i$*.*

Demostración. Razonamos por inducción en n, que es el número de factores primos.

Si $n = 1$ el corolario es trivial, por definición de número primo.

Si p divide a $p_1...p_np_{n+1}$ entonces, llamando a al producto $p_1...p_n$, tenemos que p divide al producto ap_{n+1}. Si $mcd(a, p) = 1$ entonces, por el corolario anterior, p es un número primo que divide a p_{n+1}, que también es primo. Por tanto concluimos que $p = p_{n+1}$. Podemos entonces suponer $r = mcd(a, p) \neq 1$. En particular r divide a p y, como $r \neq 1$ y p es primo, entonces $r = p$. De este modo $r = p$ divide a $a = p_1...p_n$ y se concluye por hipótesis de inducción.

3.3.3 Factorización

Finalizamos esta sección con una propiedad fundamental de los números enteros, la de la factorización única como producto de números primos. Esto es, cada número entero se puede escribir de forma única como producto de números primos.

Teorema 3.3.15 *Todo número entero $a \geq 2$ puede escribirse de forma única (salvo posibles reordenamientos) como producto de números primos,*

$$a = p_1^{a_1} \times p_2^{a_2} \times ...p_n^{a_n}$$

donde los números a_1, ..., a_n son números naturales no nulos y p_1, p_2, ..., p_n son números primos distintos. A la expresión de a de arriba se le llama **factorización prima de** a *o* **descomposición en factores primos de** a.

Demostración. Construyamos un algoritmo para factorizar el número entero $a \geq 2$. Este algoritmo recoge el procedimiento clásico de ir probando ordenadamente si los números menores que a (y mayores que 1) lo dividen, deteniéndonos en el momento en que encontramos un divisor de a. Si hemos llegado hasta a entonces a no tiene divisores y por tanto es un número primo. Si encontramos un divisor $p < a$ al ser el primero que encontramos, necesariamente es primo. En efecto, como es el primero que encontramos, p no tiene divisores menores que él, pues serían divisores de a. En esta segunda posibilidad dividimos a/p y empezamos de nuevo el proceso con el cociente a/p:

Entrada: a $(a > 2)$
$i := 2$

while $a \ mod \ i \neq 0$
  $i := i + 1$
If $i < a$ then $r :=$*i es factor de a* else $r :=$*a es primo*
Salida: r

En caso de que a no fuera primo repetimos el proceso con el cociente a/i.

Ejemplo 3.3.16 *Calculemos la factorización de 135.*

Como 135 no es par, no es divisible por 2.

Si dividimos entre 3 obtenemos que $135 = 3 \times 45$. Por tanto 3 es un factor primo de 135.

Reiniciamos el algoritmo con 45. (No hace falta probar con los primos que no dividían al número de partida, porque no pueden dividir al cociente, ver ejercicio siguiente).

De nuevo es divisible entre 3, esto es $45 = 3 \times 15$.

El cociente de nuevo es divisible entre 3, $15 = 3 \times 5$.

Y 5 es un número primo: $135 = 3^3 \times 5$.

Ejercicio 58 *Sean a, b, c enteros positivos. Supongamos que b divide a c. Demostrar que si a no divide a c entonces a no divide al cociente de c entre b.*

El algoritmo presentado muestra la **existencia** de la factorización, debemos comprobar ahora la **unicidad** de la misma. Razonamos por reducción al absurdo.

Supongamos que la factorización no es única entonces:

$$a = p_1^{a_1} \times p_2^{a_2} \times ... \times p_n^{a_n} = q_1^{b_1} \times ... \times q_m^{b_m}.$$

Por tanto:

o bien existe q_i distinto a cualquiera de los p_j, lo que es una contradicción porque entonces por un lado q_i divide a a y por el otro q_i no divide a a (aquí se está usando el corolario 3.3.14);

o bien $n = m$ y, reordenando si es necesario, $p_1 = q_1$, ..., $p_n = q_n$ y existe $a_i \neq b_i$. Supongamos que $a_i > b_i$ (si no fuera así el razonamiento se haría cambiando los papeles de la a y la b) entonces tomando el cociente $c := a/p_i^{b_i}$ se tiene que por un lado p_i divide a c y por el otro p_i no divide a c, ya que c es

producto de primos distintos de p_i (aquí de nuevo se está usando el corolario 3.3.14) lo que es una contradicción.

Por tanto concluimos la demostración del teorema, al haber demostrado tanto la existencia como la unicidad de la factorización de a.

La siguiente proposición aumenta notablemente la eficiencia del algoritmo de factorización.

Proposición 3.3.17 *Todo número entero compuesto a tiene un factor primo $p \neq 1$ menor o igual que $\sqrt{a}$.*

Demostración. Razonamos por reducción al absurdo. Supongamos que todos los factores primos de a son estrictamente mayores que $\sqrt{a}$. Como a es compuesto, tenemos

$$a = p_1^{a_1} \times ... \times p_n^{a_n}$$

donde o bien $a_1 > 1$ o bien $n > 1$ (o ambas cosas). En cualquiera de los casos, como hemos supuesto que $p_i > \sqrt{a}$ para cada $i = 1...n$, se tiene que $a > (\sqrt{a})^2 > a$ lo que es una contradicción.

Veamos un ejemplo donde se muestra la utilidad del lema.

Ejemplo 3.3.18 *Factoricemos 8872.*

Como es un número par $8872 = 2 \times 4436$.

El cociente es de nuevo par $8872 = 2^2 \times 2218$ y de nuevo $8872 = 2^3 \times 1109$.

Como $\sqrt{1109} = 33.3001$, debemos probar con los primos menores o iguales que 31, que son 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 y 31.

Es una comprobación ver que ninguno de ellos lo divide, por tanto 1109 es primo y la descomposición es

$$8872 = 2^3 \times 1109.$$

Observar la mejora que ha introducido el hecho de sólo tener que probar con los primos hasta la raíz de 1109.

Ejercicio 59 *Factoriza 1653.*

La relación de divisibilidad y sobre todo la primalidad de números enteros son conceptos profundamente ligados a la **seguridad informática** en lo que se denomina **criptografía**, referente a la transmisión de mensajes codificados para no poder ser descifrados por un ajeno al sistema. La idea fundamental es que los algoritmos que se conocen para saber si un número es o no primo precisan demasiado tiempo. Esto parece indicar que si se envía un número primo (o compuesto) suficientemente grande, un posible receptor indeseado del mensaje no va a ser capaz de saber si el número es primo o extraer sus factores.

El problema de la distribución de los primos en el conjunto de los enteros, es decir, el de conocer el primo enésimo, permanece misterioso a pesar de ser una cuestión que ha fascinado a matemáticos de todas las épocas. En la página web de la asignatura se puede leer un artículo divulgativo de M. A. Abánades sobre algunos aspectos interesantes de la teoría de los números primos.

La división entera nos abre el campo de la aritmética modular, que es el resultado de establecer unas relaciones de equivalencia en los enteros para construir otros anillos con un número finito de elementos y propiedades muy interesantes.

3.4 Relaciones de congruencia

Aunque volveremos sobre ellas en profundidad en el último capítulo, definamos lo que es una **relación de equivalencia**.

Definición 3.4.1 *Sea A un conjunto, una* **relación** *R en A se dice que es* **de equivalencia** *si verifica las propiedades:*

i) **Reflexiva***: a se relaciona con a para cualquier a en A, esto es $(a, a) \in R$ para cada a de A;*

ii) **Simétrica***: si a se relaciona con b entonces b se relaciona con a, esto es, si $(a, b) \in R$ entonces $(b, a) \in R$;*

iii) **Transitiva***: si a se relaciona con b y b se relaciona con c entonces a se relaciona con c, es decir si $(a, b) \in R$ y $(b, c) \in R$ entonces $(a, c) \in R$.*

Definición 3.4.2 *Sea $\mathbb{Z}$ el conjunto de los números enteros y $p \in \mathbb{Z}$ un número entero $p > 1$. Definimos la relación de* **congruencia módulo** *p,*

que denotamos $a \equiv b \ mod \ p$, de la siguiente manera: a es congruente con b módulo p si y solamente si $a - b$ es múltiplo de p.

Observación 3.4.3 *Si a y b son positivos entonces $a \equiv b \ mod \ p$ si y solamente si $a \ mod \ p = b \ mod \ p$.*

Ejercicio 60 *Demostrar la veracidad de la observación anterior.*

Proposición 3.4.4 *La relación de congruencia antes definida es una relación de equivalencia.*

Demostración. Debemos comprobar las tres propiedades:

Reflexiva. Se tiene la propiedad reflexiva ya que para cada $a \in \mathbb{Z}$ se verifica $a - a = 0 = 0 \times p$.

Simétrica. Si $a \equiv b \ mod \ p$ entonces existe un entero c de modo que $a - b = pc$, de este modo $b - a = p(-c)$ con lo que $b \equiv a \ mod \ p$.

Transitiva. Si $a \equiv b \ mod \ p$ y $b \equiv c \ mod \ p$ entonces existen $q, r \in \mathbb{Z}$ tales que

$$a - b = pq$$

$$b - c = pr.$$

Sumando ambas relaciones se obtiene:

$$a - c = (a - b) + (b - c) = pq + pr = p(q + r).$$

Con lo que $a \equiv c \ mod \ p$ como queríamos demostrar.

Ejercicio 61 *Comprobar las siguientes relaciones de congruencia:*

$$2 \equiv 4 \ mod \ 2 \quad 13 \equiv -2 \ mod \ 5 \quad 15 \equiv 3 \ mod \ 3.$$

La siguiente proposición observa que las operaciones de la aritmética entera, esto es, suma y producto, respetan esta relación de equivalencia. Por tanto se puede hablar de aritmética modular. Es decir, se pueden hacer sumas y productos módulo un cierto entero.

Proposición 3.4.5 *Sean $a, b, c, d, p \in \mathbb{Z}$ números enteros con $p > 1$, tales que $a \equiv c \ mod \ p$ y $b \equiv d \ mod \ p$. Se verifica que:*

i) $a + b \equiv c + d \ mod \ p$;
ii) $ab \equiv cd \ mod \ p$.

Ejercicio 62 *Usar la definición de congruencia para demostrar la proposición anterior.*

Ejemplo 3.4.6 *Tomemos la relación de congruencia módulo 5.*

Empecemos mirando los números positivos. Como $a \equiv b \ mod\ 5$ *si y solamente si el resto de dividir a o b entre 5 es el mismo (por la observación 3.4.3), nos interesa saber cuántas posibilidades existen para dicho resto. Por el teorema de la división entera el valor del resto de dividir por 5 es un número comprendido entre 0 y 4. De esta manera aparecen, al trabajar módulo 5, los números que dan resto 0 al dividir por 5 (es decir los múltiplos de 5), los que dan resto 1, los que dan resto 2, 3 ó 4.*

En este sentido, al tomar relación módulo 5, se tienen 5 elementos, los que se relacionan con el 0, con el 1, ..., con el 4:

- $0 \equiv 5 \equiv 10 \equiv 15 \equiv 20...\ mod\ 5$
- $1 \equiv 6 \equiv 11 \equiv 16 \equiv 21...\ mod\ 5$
- $2 \equiv 7 \equiv 12 \equiv 17 \equiv 22...\ mod\ 5$
- $3 \equiv 8 \equiv 13 \equiv 18 \equiv 23...\ mod\ 5$
- $4 \equiv 9 \equiv 14 \equiv 19 \equiv 24...\ mod\ 5$

Los números negativos también son de alguno de esos tipos:

- $4 \equiv -1 \equiv -6 \equiv -11 \equiv -16...\ mod\ 5$
- $3 \equiv -2 \equiv -7 \equiv -12 \equiv -17...\ mod\ 5$
- $2 \equiv -3 \equiv -8 \equiv -13 \equiv -18...\ mod\ 5$
- $1 \equiv -4 \equiv -9 \equiv -14 \equiv -19...\ mod\ 5$
- $0 \equiv -5 \equiv -10 \equiv -15 \equiv -20...\ mod\ 5$

Esto indica que para hacer operaciones con un entero a módulo un cierto número p podemos utilizar el resto de dividir a por p. Esta elección simplifica las operaciones.

Por ejemplo, si queremos multiplicar módulo 5 los números 7421 y 124590, en lugar de hacer la multiplicación directamente, tomamos los restos módulo 5 y así

$$7421 \times 124592 \equiv 1 \times 2 \equiv 2 \ mod\ 5.$$

Ejercicio 63 *Opera módulo 7, obteniendo un resultado entre 0 y 6:*

$$2345 + 214 \times 432, \quad 2419 + 987.$$

La formalización de estos conceptos, sobre la que profundizaremos en el capítulo 6 se encuentra en las siguientes definiciones:

Definición 3.4.7 *Sea A un conjunto y $\sim$ una relación de equivalencia en A. Para cada elemento $a \in A$ se define la* **clase de equivalencia de** *a, y se denota por $C(a)$ o por $\overline{a}$, como el conjunto de todos aquellos elementos de A que se relacionan con a:*

$$\overline{a} = \{b \in A : b \sim a\}.$$

Definición 3.4.8 *Sean A un conjunto y $\sim$ una relación de equivalencia en A. Se define el* **conjunto cociente** *de A por la relación de equivalencia $\sim$ como el conjunto de las clases de equivalencia:*

$$A/_{\sim} = \{\overline{a} : a \in A\}.$$

Entonces la aritmética módulo un entero positivo p son las operaciones que se realizan en el conjunto cociente de $\mathbb{Z}$ por la relación de congruencia módulo p. A este conjunto cociente lo denotaremos $\mathbb{Z}_p$ y atendiendo a las consideraciones anteriores:

$$Z_p = \{\overline{0}, \overline{1}, ..., \overline{p-1}\}.$$

Ejemplo 3.4.9 *Estudiemos las operaciones de suma y producto en $\mathbb{Z}_7$.*

Está formado por 7 clases de equivalencia:

$$Z_7 = \{\overline{0}, \overline{1}, ..., \overline{6}\}.$$

En la clase de equivalencia del 0 están los múltiplos de 7.

En la clase de equivalencia del 1 están todos los números positivos que dan resto 1 al dividir por 7, por ejemplo, 8, 15, 22. Además el -6, -13, -20...

Y así en cada clase de equivalencia.

Para sumar basta saber lo que ocurre al hacer las sumas módulo 7 de los números del 0 al 6 y lo mismo para el producto. Lo representamos en dos tablas:

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Las tablas de las sumas y los productos de los números

$$\overline{0}, \overline{1}, ..., \overline{p-1}$$

contienen todos los productos y sumas de la aritmética modular (para un cierto entero p).

Hemos abierto entonces el campo de la aritmética modular. Podemos efectuar operaciones (hacer sumas y productos) módulo un cierto número p. Tiene entonces sentido plantear ecuaciones, hacer otras operaciones, tratar de resolver sistemas de ecuaciones...

3.5 Sistemas de ecuaciones módulo enteros

Como decíamos al finalizar la sección anterior, las operaciones de la aritmética modular nos permiten plantear ecuaciones o sistemas de ecuaciones. Así podemos hablar de una **congruencia lineal** como una ecuación lineal de la forma

$$ax + b \equiv c \; mod \; p$$

donde a, b, c, p son enteros fijados, $p > 1$ y x es una indeterminada.

Ejemplo 3.5.1 *Sea por ejemplo la congruencia* $3x + 3 \equiv 4 \; mod \; 5$. *Se trata de buscar todos aquellos números enteros* x *tales que al multiplicarlos por 3 y sumarles 3 son congruentes con 4 módulo 5.*

Si razonamos como si fueran ecuaciones de primer grado tradicionales, lo que hacemos primero es pasar el 3 restando al otro miembro para escribir $3x \equiv 1 \; mod \; 5$. Esto es, sumar -3 (ó 2 ya que $-3 \equiv 2 \; mod \; 5$) a ambos lados de la equivalencia. Esta operación no supone ningún problema porque en el anillo de los números enteros podemos restar (existe el inverso para la suma) y se respetan las relaciones de congruencia.

Ahora lo que querríamos es pasar el 3 dividiendo, lo que en los números enteros no podemos hacer porque el inverso de 3 es la fracción $1/3$ que no es un número entero. La cuestión natural es saber si al trabajar módulo 5 (o módulo otros números enteros) el 3 tiene o no inverso para el producto módulo p. Para saber cuándo ocurre esto necesitamos el lema de Bezout, demostrado anteriormente.

Teorema 3.5.2 *Sean* $a, p \in \mathbb{Z}$ *números enteros positivos no nulos. Si* $mcd(a, p) = 1$ *entonces existe el inverso de* a *para el producto, es decir un entero* b *tal que* $a \times b \equiv 1 \; mod \; p$.

Demostración. Como $mcd(a, p) = 1$ el lema de Bezout garantiza la existencia de dos números enteros $s, t \in \mathbb{Z}$ de modo que:

$$1 = sa + tp$$

y por tanto al trabajar módulo p tenemos:

$$1 \equiv sa \; mod \; p,$$

Por tanto s es el inverso de a para el producto módulo p.

Observación 3.5.3 *Sean* a *y* p *enteros, con* $p > 1$. *Si existe* b *entero tal que* $ba \equiv 1 \; mod \; p$ *entonces* b *es único módulo* p.

Demostración. Supongamos $ab' \equiv 1 \; mod \, p$. Entonces $ab - ab' \equiv 0 \; mod \; p$, esto es, p divide a $a(b-b')$. Si $mcd(a, p) = 1$ podemos usar el Corolario 3.3.13 y así p divide a $b - b'$ lo que concluye la demostración. Si $r = mcd(a, p) \neq 1$, entonces existen enteros s y t tales que $a = rs$ y $p = rt$ $(1 < t < p)$. De este modo, $abt \equiv 0 \; mod \; p$. Como $ab \equiv 1 \; mod \; p$ entonces t es un múltiplo de p lo que es una contradicción, ya que $1 < t < p$.

Observación 3.5.4 *En las mismas condiciones del teorema, si* $mcd(a,p) = r \neq 1$ *no existe* $b \in \mathbb{Z}$ *tal que* $ab \equiv 1\ mod\ p$.

Demostración. Como $mcd(a,p) = r \neq 1$, entonces existen enteros $s,t \in \mathbb{Z}$ tales que $a = rs$ y $p = tr$ con $1 < t < p$. De este modo $at \equiv 0\ mod\ p$. Supongamos que existe $b \in \mathbb{Z}$ tal que $ab \equiv 1\ mod\ p$. Entonces:

$$bat \equiv t\ mod\ p$$

y de este modo

$$t \equiv 0\ mod\ p$$

lo que es una contradicción pues $1 < t < p$.

De esta manera el teorema anterior y las observaciones subsiguientes caracterizan cuando existe el inverso para el producto módulo p y además el teorema indica como calcularlo. Hay que usar el algoritmo de Euclides para llegar a la identidad del lema de Bezout.

Ejemplo 3.5.5 *Sean los números 5 y 11. Vamos a calcular el inverso de* 5 *módulo* 11. *Como son primos entre sí entonces existe* $a \in \mathbb{Z}$ *tal que* $5a \equiv 1\ mod\ 11$. *Calculamos el valor de a. El algoritmo de Euclides da la identidad de Bezout:*

$$1 = 11 + 5 \times (-2).$$

Por tanto, módulo 11:

$$1 \equiv 5 \times (-2)\ mod\ 11.$$

Por lo que $a = 9$ *es el inverso de 5 módulo 11. (Obsérvese que* $-2 \equiv 9\ mod\ 11$.*)*

Esto permite resolver la congruencia que poníamos en un ejemplo anterior:

$$3x + 3 \equiv 4\ mod\ 5.$$

Señalamos que dicha congruencia era equivalente, pasando el 3 restando, a

$$3x \equiv 1\ mod\ 5.$$

Por el teorema anterior, como 3 y 5 son primos entre sí existe el inverso de 3 módulo 5. Al trabajar módulo 5 todo número es congruente con uno del conjunto $\{0, 1, 2, 3, 4\}$. Podemos entonces buscar el inverso probando:

$$\begin{array}{l} 3 \times 0 \equiv 0 \; mod \; 5 \\ 3 \times 1 \equiv 3 \; mod \; 5 \\ 3 \times 2 \equiv 1 \; mod \; 5 \\ 3 \times 3 \equiv 4 \; mod \; 5 \\ 3 \times 4 \equiv 2 \; mod \; 5 \end{array}$$

Es decir el inverso del 3 es el 2, ya que $3 \times 2 \equiv 1 \; mod \; 5$.

Multiplicamos la congruencia por 2 y tenemos

$$3x \times 2 \equiv 1 \times 2 \; mod \; 5,$$

es decir,

$$x \equiv 2 \; mod \; 5.$$

Por tanto todos los números enteros x que son solución de la congruencia son de la forma

$$x = 5K + 2 \qquad K \in \mathbb{Z}.$$

Observar que hemos propuesto dos maneras de calcular el inverso de un número a módulo p. Por un lado usar la identidad de Bezout, por otro lado ir probando con los distintos productos $a \cdot 1$, $a \cdot 2$, ..., $a \cdot (p-1)$ hasta encontrar el producto que valga 1.

Ejemplo 3.5.6 *Como*

$$\begin{array}{lll} 3 \cdot 1 \equiv 3 \; mod \; 6 & 3 \cdot 2 \equiv 0 \; mod \; 6 & 3 \cdot 3 \equiv 3 \; mod \; 6 \\ 3 \cdot 4 \equiv 0 \; mod \; 6 & 3 \cdot 5 \equiv 3 \; mod \; 6 & 3 \cdot 0 \equiv 0 \; mod \; 6 \end{array}$$

entonces 3 no tiene inverso módulo 6. Obsérvese que no se tienen las hipótesis del teorema, ya que $mcd(3, 6) = 3$.

Ejercicio 64 *Demostrar que la congruencia lineal*

$$3x + 4 \equiv 5 \; mod \; 6$$

no tiene solución.

Ejercicio 65 *Hallar todas las soluciones enteras de la congruencia*

$$5x + 2 \equiv 5 \; mod \; 7.$$

Si se pueden plantear ecuaciones lineales, es natural ahora plantear el problema de resolver **sistemas de congruencias lineales**, es decir, expresiones del tipo

$$x \equiv a_1 \; mod \; p_1$$
$$x \equiv a_2 \; mod \; p_2$$
$$...$$
$$x \equiv a_n \; mod \; p_n$$

El **Teorema chino de los restos** es el instrumento adecuado para saber si un sistema de este tipo tiene solución y permite calcularla.

Teorema 3.5.7 *Sean $a_1, a_2, ..., a_n$ números enteros y $p_1, p_2, ..., p_n$ enteros positivos verificando:*
i) $p_i > 1$ para cada $i = 1, ..., n$;
ii) $mcd(p_i, p_j) = 1$ para cada $i, j = 1, ..., n$ con $i \neq j$.
Entonces el sistema

$$x \equiv a_1 \; mod \; p_1$$
$$x \equiv a_2 \; mod \; p_2$$
$$...$$
$$x \equiv a_n \; mod \; p_n$$

tiene solución única módulo el producto $P = p_1 p_2 ... p_n$.

Demostración. Sean $P = p_1 ... p_n$ y $P_i = P/p_i$ con i un entero entre 1 y n. Por hipótesis los p_i son primos entre sí con lo cual $mcd(P_i, p_i) = 1$, $i = 1, ..., n$. Como hemos visto antes (Teorema 3.5.2) para cada i existirá q_i tal que

$$q_i P_i \equiv 1 \; mod \; p_i.$$

Sea entonces

$$x_0 = a_1P_1q_1 + a_2P_2q_2 + ... + a_nP_nq_n$$

de modo que al tomar congruencia módulo p_i se obtiene:

si $j \neq i$ entonces P_j es divisible por p_i, por tanto $P_j \equiv 0 \; mod \; p_i$
si $j = i$ entonces $P_iq_i \equiv 1 \; mod \; p_i$,

por tanto $x_0 \equiv a_i \; mod \; p_i$ $(i = 1, ..., n)$ como queríamos demostrar. Así todos los números de la forma

$$x = x_0 + KP$$

con $K \in \mathbb{Z}$ son solución del sistema.

Demostramos ahora la unicidad de la solución módulo P. Esto prueba además que toda solución del sistema es de la forma $x = x_0 + KP$ con $K \in \mathbb{Z}$.

Tomemos α y β dos soluciones del sistema, entonces $\alpha - \beta$ es solución del sistema de congruencias

$$x \equiv 0 \; mod \; p_1$$

$$x \equiv 0 \; mod \; p_2$$

$$...$$

$$x \equiv 0 \; mod \; p_n$$

es decir, $\alpha \equiv \beta \; mod \; P$, (ya que $\alpha - \beta$ es múltiplo de p_i para cada i) lo que demuestra la unicidad de la solución módulo P.

Ejemplo 3.5.8 *Resolver el sistema:*

$$x \equiv 2 \; mod \; 3$$
$$x \equiv 3 \; mod \; 5$$
$$x \equiv 2 \; mod \; 7$$

Como 3, 5 y 7 son primos entre sí, el teorema chino de los restos permite calcular la única solución módulo $3 \times 5 \times 7 = 105$.

Siguiendo la notación del teorema

$$P_1 = 105/3 = 35$$
$$P_2 = 105/5 = 21$$
$$P_3 = 105/7 = 15$$

Ahora q_1 es el inverso de 35 módulo 3. Como $35 \equiv 2 \ mod \ 3$ entonces entonces $q_1 = 2$, ya que $2 \times 2 \equiv 1 \ mod \ 3$.

De la misma manera q_2 es el inverso de 21 módulo 5, es decir, el inverso de 1 módulo 5, por tanto $q_2 = 1$.

Finalmente $q_3 = 1$.

Por tanto

$$x = 2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1 = 233$$

es la única solución módulo 105, de modo que todas las soluciones del sistema son:

$$x = 23 + 105K \quad K \in \mathbb{Z}.$$

Una aplicación que podríamos presentar del teorema chino de los restos es la observación de que los restos módulo un conjunto de números primos entre sí permiten determinar unívocamente un número. De esta manera números de gran tamaño se podrían representar por sus restos módulo unos ciertos primos y así reducir sustancialmente su tamaño. Con estos restos se pueden hacer operaciones y la recuperación del número en cuestión pasa por la resolución de un sistema de congruencias. Esto resolvería parcialmente, como veremos en el ejemplo siguiente, la limitación de dígitos de una máquina.

Ejemplo 3.5.9 *Supongamos que tenemos una calculadora de 8 dígitos y queremos hacer la suma de los números 5888851358 y 259632147. Tomamos, por ejemplo, los números primos entre sí 100, 99 y 97 cuyo producto P es 960.300, menor de 8 cifras.*

El primer número verifica:

$$5888851358 \equiv 58 \ mod \ 100$$
$$5888851358 \equiv 72 \ mod \ 99$$
$$5888851358 \equiv 78 \ mod \ 97.$$

El segundo verifica:

$$259632147 \equiv 47 \ mod \ 100$$
$$259632147 \equiv 93 \ mod \ 99$$
$$259632147 \equiv 7 \ \ mod \ 97.$$

Y para hacer la suma basta hacer la suma de los restos, y después resolver el sistema

$$x \equiv 47 + 58 \ mod\ 100$$
$$x \equiv 72 + 93 \ mod\ 99$$
$$x \equiv 78 + 7 \ \ mod\ 97$$

En este último paso, para recuperar el número, hay que hacer operaciones con enteros de más de 8 dígitos, por lo que, propiamente, con 8 dígitos sólo podemos hacer aritmética con las representaciones por sus restos módulo 100, 99 y 97.

También puede usarse este teorema para hacer recuentos con números grandes.

Ejemplo 3.5.10 *Supongamos que tenemos una manifestación que sabemos que ha congregado a menos de un millón de personas y queremos, desde la organización, saber el número exacto de congregados. Les pedimos que se agrupen de 100 en 100 y apuntamos el resto. Pongamos que son 30. Esto indica que el número x de participantes verifica*

$$x \equiv 30 \ mod\ 100.$$

Después les pedimos que se agrupen en grupos de 99, y sobran 25 y en grupos de 97 y sobran 13. Es decir, se tiene el siguiente sistema de congruencias:

$$x \equiv 30 \ mod\ 100$$
$$x \equiv 25 \ mod\ 99$$
$$x \equiv 13 \ mod\ 97$$

Como 100, 99 y 97 son primos entre sí, el sistema tiene solución. Como $100 \cdot 99 \cdot 97 = 960.300$ *entonces la única solución positiva y menor que ese número es el número de participantes en la manifestación.*

Ejercicio 66 *Determinar el número de participantes en la anterior manifestación.*

3.6 Sistemas de numeración

Terminamos este tema con esta sección dedicada a los sistemas de numeración, es decir a las formas de representar los números enteros. La base

escogida para la numeración es arbitraria y está universalmente admitida la base 10 (sistema de numeración decimal) quizás debida a nuestra propia anatomía que presenta 10 dedos para poder contar con ellos. Pero otras bases de numeración son también importantes: la base dos (sistema de numeración binaria) es fundamental en la informática y las telecomunicaciones, donde la información está compuesta de ceros y unos; la base 60 es también importante en algunos campos (p.ej. medida de ángulos) debido al gran número de divisores del 60 (1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60); también la base 12 es relevante en medidas temporales, por razones similares a la base 60. Vamos a ver que todos estos sistemas permiten representar, mediante notación posicional, cualquier número entero y hacer operaciones con esta notación.

Cuando escribimos el número 12345 en base 10 estamos usando lo que se llama **notación posicional** y cada dígito tiene un significado por el lugar que ocupa. Así, empezando por la derecha, la primera cifra es la de las unidades, la siguiente la de las decenas..., es decir,

$$12345 = 5 \times 10^0 + 4 \times 10^1 + 3 \times 10^2 + 2 \times 10^3 + 1 \times 10^4.$$

Y esto que se hace con la base 10, expresar el número como combinación lineal de las potencias de la base, se puede hacer para un valor arbitrario de la base.

Teorema 3.6.1 *Sean a y b números enteros positivos, con $b > 1$. Existen unos valores únicos $a_0, a_1, ...a_n$, con $0 \leq a_i < b$ y con $a_n \neq 0$, de modo que*

$$a = a_0 \times b^0 + a_1 \times b^1 + a_2 \times b^2 + ... + a_n \times b^n.$$

Demostración. Dividimos a entre b para obtener

$$a = q_0 b + r_0.$$

Esto muestra que $a_0 = r_0$ porque $0 \leq r_0 < b$. Además muestra la unicidad de a_0 ya que el teorema del resto garantiza que el resto es único.

Si dividimos ahora q_0 entre b se obtiene

$$q_0 = bq_1 + r_1$$

y entonces

$$a = b^2 q_1 + b r_1 + a_0,$$

con lo cual $a_1 = r_1$ (de nuevo único porque es un resto), y así sucesivamente hasta que el cociente q_n sean menor que b.

Ejercicio 67 *Diseñar un algoritmo para escribir un número escrito en base 10 en base a.*

Diseñar un algoritmo para escribir un número escrito en base a en base 10.

Diseñar un algoritmo para escribir un número escrito en base a en base b.

Ejercicio 68 *Escribir en base 10 los siguientes números, escritos en base 2, 3 y 5 respectivamente:*

$$101001001 \qquad 12101 \qquad 342104$$

Ejercicio 69 *Escribir el número 1465 en base 2, 5 y 7.*

Y finalmente se pueden escribir algoritmos para hacer operaciones de suma y producto de números escritos en cualquier base de numeración, sin más que tener en cuenta que lo que tradicionalmente se expresa como *me llevo una*, que en base 10 se hace cuando se llega a 10, en una base de numeración cualquiera se hace cuando se llega al valor de la base.

Ejemplo 3.6.2 *Sumamos en base 2 los números 1101 y 1001.*

$$\begin{array}{r} 1101 \\ 1001 \\ \hline 10110 \end{array}$$

En la primera cifra hemos sumado 1+1 que da 2 que alcanza a la base, por tanto da 0 y me llevo 1.

Ejemplo 3.6.3 *Multiplicamos en base 2 los números 1101 y 11.*

$$\begin{array}{r} 1101 \\ 11 \\ \hline 1101 \\ 1101 \\ \hline 100111 \end{array}$$

Ejercicio 70 *Escribir algoritmos para sumar en base 2 y para multiplicar en base 2.*

3.7 Ejercicios

Ejercicio 71. Poner dos ejemplos de:
i) números congruentes con 2 módulo 3,
ii) números congruentes con 5 módulo 7,
iii) números congruentes con 11 módulo 5,
iv) números no nulos módulo 12 tal que su producto módulo 12 sea nulo,
v) números distintos de 1 módulo 15 tal que su producto sea 1.

Ejercicio 72. Calcular usando el algoritmo de Euclides:
i) $mcd(10223, 33341)$,
ii) $mcd(385, 1729)$.

Ejercicio 73. Demostrar la regla de divisibilidad por 9: *n es divisible por 9 si la suma de sus cifras es divisible por 9.*

Ejercicio 74. Realiza las siguientes operaciones:
i) 3+5, 4+8, 9×8, 84^{1234}, 245+1321 módulo 5
ii) 4+5, 1044+8, 9×8, 84^{1234}, 245+1321 módulo 2
iii) 3+75, 234+458, 9×8, 84^{1234}, 245+1321 módulo 7.

Ejercicio 75. Demostrar que el siguiente sistema de congruencias no tiene solución:

$$x \equiv 2 \; mod \; 6$$
$$x \equiv 3 \; mod \; 9$$

Ejercicio 76. Demostrar que los siguientes sistemas tienen solución y resolverlos:

i)
$$x \equiv 3 \; mod \; 5$$
$$x \equiv 5 \; mod \; 7$$

ii)
$$y \equiv 5 \; mod \; 11$$
$$y \equiv 7 \; mod \; 13$$

3.8 Ejercicios Resueltos

Ejercicio 51. Comenzamos con $c = 1$. Entonces usando las reglas de la suma se tiene

$$(a + b) + 1 = s(a + b) = a + s(b) = a + (b + 1).$$

Ahora supongamos que $(a + b) + c = a + (b + c)$. De nuevo de las reglas de la suma se tiene

$$(a+b)+s(c) = s((a+b)+c) = s(a+(b+c)) = a+s(b+c) = a+(b+s(c)).$$

Ejercicio 52. Fijamos $n \in \mathbb{N}$. Tomamos el conjunto $A = \{m \in \mathbb{N} : \; nm$ está definido$\}$. La primera regla del producto garantiza que $1 \in A$. La segunda regla dice que si $m \in A$ entonces $s(m) \in A$. De este modo $A = \mathbb{N}$ por el principio de inducción.

Ejercicio 53. Fijamos $n \in \mathbb{Z}$. Tomamos el conjunto $A = \{m \in \mathbb{Z} : \; n + m$ está definido$\}$. Como $1 \in A$ entonces $A \neq \emptyset$. Si $m \in A$ las reglas tercera y cuarta garantizan que $s(m) \in A$ y que $a(m) \in A$. Por tanto $A = \mathbb{Z}$ por inducción estructural.

Ejercicio 54. Basta tener en cuenta el signo y usar el producto que tenemos en los números enteros no negativos, $\mathbb{N}^+$.

Definimos como es habitual el valor absoluto de $n \in \mathbb{Z}$ como $|n| = n$ si $n \in \mathbb{N}^+$, $|0| = 0$ y $|n| = -n$ si $n \in \mathbb{N}^-$:

- $a \times 0 = 0 \times a = 0$
- $a \times b = |a| \times |b|$ si o bien $a > 0$ y $b > 0$ o bien $a < 0$ y $b < 0$.
- $a \times b = -|a| \times |b|$ en el resto de los casos.

Ejercicio 55. Dadas dos matrices, se define la suma como la suma de cada una de sus entradas. Como cada entrada está formada por un número entero, la asociatividad de la suma de matrices es consecuencia de la asociatividad de la suma de los números enteros. Idem la conmutatividad.

El elemento neutro es la matriz nula, cuyas entradas son nulas.

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

El elemento inverso para la suma de una matriz A es la matriz $-A$ resultado de cambiar de signo cada entrada de A, denominada con la letra a y su correspondiente subíndice.

$$-A = \begin{pmatrix} -a_{11} & -a_{12} \\ -a_{21} & -a_{22} \end{pmatrix}$$

Sean dos matrices A y B cuyas entradas son denominadas respectivamente con las letras a y b (y sus correspondientes subíndices). El producto de matrices se define:

$$AB = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{11} + a_{22}b_{21} \end{pmatrix}$$

Comprobamos que es asociativo, esto es, $(AB)C = A(BC)$, donde las entradas de C están denominadas con la letra c. Lo hacemos para la entrada de subíndice 11 del producto, de igual manera se haría para las otras tres entradas.

En el producto $(AB)C$ se tiene que dicha entrada es:

$$(a_{11}b_{11} + a_{12}b_{21})c_{11} + (a_{11}b_{12} + a_{12}b_{22})c_{21}.$$

En el producto $A(BC)$ se tiene que dicha entrada es:

$$a_{11}(b_{11}c_{11} + b_{12}c_{21}) + a_{12}(b_{21}c_{11} + b_{22}c_{21}).$$

Y es una comprobación verificar que ambas expresiones son iguales.

El elemento neutro para el producto es la matriz identidad, con la diagonal formada por unos y el resto ceros.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Tomando por ejemplo las matrices

$$M = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

$$N = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$$

se puede comprobar que $MN \neq NM$ con lo que el producto de matrices no es conmutativo.

Ejercicio 56. El resto de dividir 9981 entre 247 es 101. El resto de dividir 247 entre 101 es 45. El resto de dividir 101 entre 45 es 11. El resto de dividir 45 entre 11 es 1. Por tanto:

$mcd(9981, 247) = mcd(247, 101) = mcd(101, 45) = mcd(45, 11) = mcd(11, 1) = 1.$

Ejercicio 57.

Entrada: $a, b \in \mathbb{N}$, $a \geq b$
$i := a$, $j := b$
while $j \neq 0$
    $r := i \ mod \ j$
    $i := j$
    $j := r$
Salida: i

Ejercicio 58. Supongamos que a divide al cociente $q = c/b \in \mathbb{Z}$ entonces existe un entero s de modo que $as = q$. Esto implica $c = abs$ en contradicción con el hecho de que a no divide a c.

Ejercicio 59. El número 1653 no es par pero sí es múltiplo de 3. Entonces dividiendo entre 3 se obtiene $1653 = 3 \times 551$. Ahora bien $23^2 = 529$ y $24^2 = 576$. Por tanto 551 ha de tener un factor menor o igual que 23. El 19 divide a 551. Se tiene que $551 = 19 \times 29$. Como 29 es primo se tiene:

$$1653 = 3 \times 19 \times 29.$$

Ejercicio 60. Si $r = a \ mod \ p = b \ mod \ p$, entonces se tiene que existen enteros no negativos q y q' tales que $a = pq + r$ y $b = pq' + r$. Se sigue que $a - b = p(q - q')$, con lo que $a \equiv b \ mod \ p$.

Sin perdida de generalidad supongamos $a \geq b$. Por el teorema del resto $q, q' \in \mathbb{N} \cup \{0\}$ y enteros r y r' tales que $0 \leq r < p$, $0 \leq r' < p$ y $a = pq + r$ y $b = pq' + r'$. Entonces $a - b = p(q - q') + (r - r')$. De este modo $r - r' = (a - b) - p(q - q')$ es un múltiplo de p. Como $-p < r - r' < p$ entonces necesariamente $r = r'$, como queríamos demostrar.

Ejercicio 61. $2 \equiv 4 \ mod \ 2$ porque $2 - 4 = -2$.
$13 \equiv -2 \ mod \ 5$ porque $13 + 2 = 5 \times 3$.
$15 \equiv 3 \ mod \ 3$ porque $15 - 3 = 4 \times 3$.

Ejercicio 62. Si $a \equiv c \ mod \ p$ entonces existe un entero s tal que:

$$a - c = sp.$$

Si $b \equiv d \ mod \ p$ entonces existe un entero t tal que:

$$b - d = tp.$$

Sumando ambas igualdades se tiene:

$$(a + b) - (c + d) = p(t + s).$$

De modo que $a + b \equiv c + d \ mod \ p$.

Por otro lado multiplicando por b la primera igualdad se tiene:

$$ab - bc = spb.$$

Multiplicando por c la segunda igualdad se tiene:

$$bc - dc = tpc.$$

Sumando las dos igualdades se tiene que $ab - dc = p(ab + tc)$ como queríamos demostrar.

Ejercicio 63. $2345 + 214 \times 432 \equiv 6 \; mod \; 7$.
$2419 + 987 \equiv 4 \; mod \; 7$.

Ejercicio 64. Si existe $x \in \mathbb{Z}$ tal que $3x + 4 \equiv 5 \; mod \; 6$, entonces $3x \equiv 1 \; mod \; 6$, de modo que existe un entero k tal que

$$3x = 6k + 1$$

lo que da una contradicción pues 1 no es múltiplo de 3.

Ejercicio 65. Como $5x+2 \equiv 5 \; mod \; 7$ pasamos el dos restando para obtener $5x \equiv 3 \; mod \; 7$. Como el $mcd(5, 7) = 1$ entonces existe el inverso de 5 para el producto módulo 7. En efecto, dicho inverso es 3 ya que $5 \times 3 \equiv 1 \; mod \; 7$. De este modo multiplicando por 3 obtenemos $x \equiv 2 \; mod \; 7$. Entonces las soluciones son de la siguiente forma donde $k \in \mathbb{Z}$:

$$x = 7k + 2.$$

Ejercicio 66. Las congruencias lineales planteadas tienen solución por el teorema chino de los restos. Dicha solución es 316.330.

Los ejercicios 67), 68), 69) y 70) Se presentan hechos con Maple.

3.8.1 Bases de numeración con Maple

Cambios de base de sistema de numeración

Veamos cómo se pueden implementar en Maple distintos algoritmos de cambio de base.

Ejercicio 67.

De base 10 a base a

La entrada es el número entero n y la base a. El algoritmo debe ir calculando las cifras del número en base a haciendo divisiones sucesivas por a y tomando los restos. La salida será una lista a_0, a_1, ..., a_m de modo que n se escribe como $a_m...a_1a_0$ en base a.

```
> cambio10a:=proc(n,a)
> local b,i,j,l,s:
> i:=n:  s:=1:
> while i<>0 do
> b[s]:=i mod a:
> s:=s+1:
> i:=floor(i/a);
> od:
> seq(b[j],j=1..s-1);
> end:
```

Veamos algunos ejemplos

```
> cambio10a(8,2);
```

$$0, 0, 0, 1$$

```
> cambio10a(724,6);
```

$$4, 0, 2, 3$$

Comprobemos que en efecto éste es el resultado:

```
> 4*6^0+0*6^1+2*6^2+3*6^3;
```

$$724$$

Otro ejemplo y su comprobación:

```
> cambio10a(1234,8);
```

$$2, 2, 3, 2$$

```
> 2*(8^0)+2*(8^1)+3*(8^2)+2*(8^3);
```

$$1234$$

Ejercicio 69.

```
> cambio10a(1465,2);
```

$$1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1$$

```
> cambio10a(1465,5);
```

$$0, 3, 3, 1, 2$$

```
> cambio10a(1465,7);
```

$$2, 6, 1, 4$$

Por tanto 1465 en base 2, en base 5 y en base 7 es, respectivamente, 10110111001, 21330 y 4162.

De base a a base 10

El procedimiento es el siguiente:

```
> cambioa10:=proc(l,a)
> local s,i:
> s:=l[1]:
> for i from 2 to nops(l)
> do
> s:=s+l[i]*a^(i-1):
> od:
> s;
> end:
```

Veamos ejemplos:

```
> cambio10a(567,3);
```

$$0, 0, 0, 0, 1, 2$$

```
> cambioa10([cambio10a(567,3)],3);
```

$$567$$

```
> cambioa10([1,1,1,0,1],2);
```

$$23$$

Ejercicio 68.

```
> cambioa10([1,0,0,1,0,0,1,0,1],2);
```

$$329$$

```
> cambioa10([1,0,1,2,1],3);
```

$$145$$

```
> cambioa10([4,0,1,2,4,3],7);
```

$$60764$$

De base a a base b

Lo hacemos usando los procedimientos antes definidos, pasando por la base 10:

```
> cambioab:=proc(l,a,b)
> local temp:
> temp:=cambioa10(l,a):
> cambio10a(temp,b);
> end:
```

Ejemplos:

```
> cambioab([1,1,0,1],2,3);
```

$$2, 0, 1$$

```
> cambioa10([1,1,0,1],2);
> cambioa10([2,0,1],3);
```

$$11$$

$$11$$

Sumas y productos en base 2

En el procedimiento siguiente el número más largo va en primer lugar.

```
> suma2:=proc(l,m) local t,i,s,r,j:
> t:=[seq(m[i],i=1..nops(m)),seq(0,i=1..nops(l)-nops(m))]:
> r[0]:=0:
> i:=1
> while i<nops(l)+1 do
> s[i]:=(l[i]+t[i]+r[i-1]) mod 2:
> r[i]:=floor((l[i]+t[i]+r[i-1])/2):
> i:=i+1:  od:
> s[i]:=r[i-1]:
> [seq(s[j],j=1..nops(l)+1)];
> end:
> suma2([1],[1]);
```

[0, 1]

```
> suma2([0,1],[1]);
```

[1, 1, 0]

Damos sólo la idea para construir el algoritmo de multiplicación. Se debe hacer lo siguiente: vamos a multiplicar la lista l con *nops(l)* elementos y la lista m con *nops(m)* elementos. Hay que construir un bucle que recorre la lista m. Si la entrada primera es un 0 no hace nada, si no toma la lista l. Si la entrada segunda es 0 no hace nada, si no toma la lista l y le añade un 0 al final. Suma la primera y la segunda lista y así procede sucesivamente.

Ejercicio 71.
i) el 2+3=5 y 2+2×3=8.
ii) el 5+7=12 y el 5+7×2=19.
iii) el 11+5=16 y el 11+2×5=21
iv) 3 y 4 ($3 \times 4 = 12 \equiv 0 \ mod\ 12$), 2 y 6 ($2 \times 6 = 12 \equiv 0 \ mod\ 12$).
v) 2 y 8 ($2 \times 8 = 16 \equiv 1 \ mod\ 15$), 4 y 4 ($4 \times 4 = 16 \equiv 1 \ mod\ 15$).

Ejercicio 72.
i)

$$mcd(10223, 33341) = mcd(33341, 10223)$$

Como $33341 \ mod\ 10223 = 2672$,

$$mcd(33341, 10223) = mcd(10223, 2672)$$

Como $10223 \ mod\ 2672 = 2207$,

$$mcd(10223, 2672) = mcd(2672, 2207)$$

Como $2672 \ mod\ 2207 = 465$,

$$mcd(2672, 2207) = mcd(2207, 465)$$

Como $2207 \ mod\ 465 = 347$,

$$mcd(2207, 465) = mcd(465, 347)$$

Como $465 \ mod\ 347 = 118$,

$$mcd(465, 347) = mcd(347, 118)$$

Como $347 \; mod \; 118 = 111$,

$$mcd(347, 118) = mcd(118, 111)$$

Como $118 \; mod \; 111 = 7$,

$$mcd(118, 111) = mcd(111, 7)$$

Como $111 \; mod \; 7 = 6$,

$$mcd(111, 7) = mcd(7, 6)$$

Finalmente, como $7 \; mod \; 6 = 1$,

$$mcd(7, 6) = mcd(6, 1) = 1$$

Los dos números dados son primos entre sí.

ii)

$$mcd(385, 1729) = mcd(1729, 385)$$

Como $1729 \; mod \; 385 = 189$,

$$mcd(1729, 385) = mcd(385, 189)$$

Como $385 \; mod \; 189 = 7$,

$$mcd(385, 189) = mcd(189, 7)$$

Como $189 \; mod \; 7 = 0$,

$$mcd(189, 7) = mcd(7, 0)$$

y nuestro máximo común divisor es el 7.

Ejercicio 73. Un número n de, pongamos, m cifras se puede escribir como

$$n = a_0 + a_1 10 + a_2 10^2 + a_3 10^3 + ... + a_m 10^m$$

siendo a_i la cifra $i-$ésima del número.

Como $10 \equiv 1 \; mod \; 9$ entonces $10^n \equiv 1 \; mod \; 9$ para cada $n \in \mathbb{N}$. De este modo al trabajar módulo 9 se tiene:

$$n \equiv a_0 + a_1 + ... + a_n \; mod \; 9.$$

Lo que significa que n es múltiplo de 9 si y solamente si lo es $a_0 + a_1 + ... + a_n$.

Ejercicio 74.

i) $3 + 5 = 8$. $8 \; mod \; 5 = 3 \; mod \; 5$

$4 + 8 = 12 = 2 \; mod \; 5$

$9 \times 8 \; mod \; 5 = (-1) \times (-2) \; mod \; 5 = 2 \; mod \; 5$

$84^{1234} \; mod \; 5 = (-1)^{1234} \; mod \; 5 = 1 \; mod \; 5$ (hemos usado que $84 \; mod \; 5 = (-1) \; mod \; 5$)

$(245 + 1321) \; mod \; 5 = (0 + 1) \; mod \; 5 = 1 \; mod \; 5$

ii) $(4 + 5) \; mod \; 2 = (0 + 1) \; mod \; 2 = 1 \; mod \; 2$

$(1044 + 8) \; mod \; 2 = (0 + 0) \; mod \; 2 = 0 \; mod \; 2$

$(9 \times 8) \; mod \; 2 = (1 \times 0) \; mod \; 2 = 0 \; mod \; 2$

$84^{1234} \; mod \; 2 = 0^{1234} \; mod \; 2 = 0 \; mod \; 2$

$(245 + 1321) \; mod \; 2 = (1 + 1) \; mod \; 2 = 0 \; mod \; 2$

iii) $(3 + 75) \; mod \; 7 = (3 + (-2)) \; mod \; 7 = 1 \; mod \; 7$

$(234 + 458) \; mod \; 7 = (3 + 3) \; mod \; 7 = 6 \; mod \; 7$

$(9 \times 8) \; mod \; 7 = (2 \times 1) \; mod \; 1 = 2 \; mod \; 1$

$84^{1234} \; mod \; 7 = 0^{1233} \; mod \; 7 = 0 \; mod \; 7$

$(245 + 1321) \; mod \; 7 = (0 + 5) \; mod \; 7 = 5 \; mod \; 7$.

Ejercicio 75. En principio, no tiene por qué haber solución, ya que 6 y 9 no son primos entre sí, y el teorema chino de los restos no se aplica. Veremos que, en efecto, este es un ejemplo de no existencia de solución.

Si $x \equiv 2 \; mod \; 6$ entonces x se puede escribir de la siguiente forma donde $K_1 \in \mathbb{Z}$

$$x = 2 + 6K_1$$

y si $x \equiv 3 \; mod \; 9$ entonces entonces x se puede escribir de la siguiente forma donde $K_2 \in \mathbb{Z}$

$$x = 3 + 9K_2$$

Tenemos pues

$$2 + 6K_1 = 3 + 9K_2$$

que implica

$$6K_1 - 9K_2 = 1$$

Pero $6K_1 - 9K_2 = 3(2K_1 - 3K_2)$ lo que implica

$$3(2K_1 - 3K_2) = 1$$

es decir, 1 es divisible por 3, lo cual no es cierto.

Ejercicio 76. En ambos casos se aplica el teorema chino de los restos (5 y 7 son primos entre sí, 11 y 13 son primos entre sí).

i) $p_1 = 5$, $p_2 = 7$. $P = 35$. $P_1 = 7, P_2 = 5$. Buscamos q_1 y q_2 tales que

$$q_1 7 \equiv 1 \ mod \ 5$$
$$q_2 5 \equiv 1 \ mod \ 7$$

Probamos los diversos valores posibles de q_1 y vemos que $q_1 = 3$ satisface la ecuación. Hacemos lo mismo con q_2 y comprobamos que $q_2 = 3$ satisface la ecuación.

Entonces, el número buscado es

$$x = (3 \times 3 \times 7 + 5 \times 3 \times 5) \ mod \ 35 = 138 \ mod \ 35 = 33 \ mod \ 35$$

De este modo todas las soluciones son de la forma siguiente con $k \in \mathbb{Z}$:

$$x = 33 + 35k$$

ii) $p_1 = 11$, $p_2 = 13$. $P = 143$. $P_1 = 13, P_2 = 11$. Buscamos q_1 y q_2 tales que

$$q_1 13 \equiv 1 \ mod \ 11$$
$$q_2 11 \equiv 1 \ mod \ 13$$

Probamos los diversos valores posibles de q_1 y vemos que $q_1 = 6$ satisface la ecuación. Hacemos lo mismo con q_2 y comprobamos que $q_2 = 6$ satisface la ecuación.

Entonces, el número buscado es

$$x = (5 \times 6 \times 13 + 7 \times 6 \times 11) \ mod \ 143 = 852 \ mod \ 143 = 137 \ mod \ 143$$

De este modo todas las soluciones son de la forma siguiente con $k \in \mathbb{Z}$:

$$x = 137 + 143k.$$

Capítulo 4

Combinatoria

La combinatoria es el arte de contar, es decir, de calcular inteligentemente cardinales de conjuntos, y de enumerar, esto es, determinar los elementos de un conjunto descrito por alguna propiedad. Es una disciplina clásica que cobra nuevo auge con la aparición de los ordenadores por dos razones: por un lado por la posibilidad de cálculo que éstos aportan, y por otro porque en el estudio de algoritmos o en el análisis de programas los problemas del tipo cálculo del número de operaciones, unidades de memoria que se precisan para realizar una cierta operación, estudio de la complejidad... son problemas de naturaleza combinatoria.

Se pretende que el alumno al finalizar el capítulo:

- Domine las reglas fundamentales del cálculo combinatorio.
- Entienda y utilice los conceptos de variaciones, combinaciones y permutaciones y los pueda aplicar para calcular cardinales de conjuntos.
- Pueda aplicar estas nociones para estudiar la probabilidad de sucesos de experimentos con una cantidad finita de resultados posibles.

4.1 Introducción

Como hemos señalado en el párrafo introductorio, el primer problema que tratamos de abordar es el de **contar**, es decir, computar el número de elementos de un conjunto. Y además hacerlo de manera inteligente. Comencemos con un ejemplo.

Ejemplo 4.1.1 *Sea una competición ajedrecística con 64 participantes que se juega por el sistema de eliminatoria, es decir, en cada partida el ganador pasa a la siguiente fase y el perdedor queda eliminado. Determinar el número de partidas que se han de jugar para obtener un campeón.*

Para una persona acostumbrada al sistema de eliminatorias el problema se resuelve fácilmente:

Empezamos con las 32 partidas de los 32-avos de final.
Proseguimos con las 16 de los dieciseisavos de final.
Después los 8 octavos de final.
Los cuatro cuartos de final.
Las dos semifinales.
Y la final.

$$Total : 32 + 16 + 8 + 4 + 2 + 1 = 63.$$

El problema también se puede resolver de una manera más inteligente y más fácil de generalizar a un número cualquiera de participantes. La idea clave es la siguiente: en cada partida se elimina un jugador. Para que se tenga un ganador hay que eliminar a todos los jugadores salvo a uno (el ganador), por tanto el número de partidas es:

$$Total : 64 - 1 = 63.$$

Y mientras la primera fórmula aplicada por ejemplo a 512 jugadores es:

$$256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 511.$$

La segunda fórmula es sencillamente:

$$512 - 1 = 511.$$

Con este ejemplo hemos querido mostrar lo que se pretende al desarrollar la *combinatoria*, no sólo contar todos los elementos que conforman un conjunto finito, sino hacerlo de forma inteligente para que se puedan obtener fórmulas generalizables a situaciones afines.

Por otro lado también la combinatoria presenta una faceta distinta, la de **enumerar** los elementos de un conjunto finito. Es decir, dado un conjunto definido por una cierta propiedad, diseñar algoritmos que presenten todos los elementos del conjunto.

Ejemplo 4.1.2 *El consejo de ministros (pongamos que son 15 personas) debe elegir a dos de sus miembros para enviarlos a una misión de paz en el África subsahariana acompañando al ministro de exteriores. Determinar cuántas posibles parejas de acompañantes hay.*

Si el consejo de ministros está formado por 15 personas, y hay que elegir 2 para acompañar al ministro de exteriores, entonces hay 14 candidatos elegibles que podemos numerar con las cifras del 1 al 14, es decir se puede establecer una función biyectiva (recordaremos su definición en la próxima sección) entre el conjunto de posibles ministros acompañantes y el conjunto

$$\mathbb{N}_{14} = \{1, 2, 3, 4, ..., 13, 14\}.$$

El problema es decir cuántos y cuáles son los subconjuntos de dos elementos de $\mathbb{N}_{14}$. Lo hacemos ordenadamente:

Los subconjuntos que contienen al elemento 1 son los 13 subconjuntos formados por el 1 y otro ministro cualquiera, esto es,

$$\{1, 2\}, \{1, 3\}, ..., \{1, 14\}.$$

Para el elemento 2 basta elegir entre 12 acompañantes (del 3 al 14) porque ya hemos tenido en cuenta la pareja $\{1, 2\}$. Así son 12:

$$\{2, 3\}, \{2, 4\}, ..., \{2, 14\}.$$

Este proceso se puede ir repitiendo y escribir la lista completa, además sirve para demostrar que las parejas posibles son:

$$13 + 12 + 11 + 10 + 9 + ... + 1.$$

Y por la fórmula de la suma de los 13 primeros números naturales:

$$(14 \times 13)/2.$$

Por tanto principalmente **contar** pero también **enumerar** son los objetivos del capítulo.

4.2 Técnicas de recuento

Vamos a precisar las definiciones de los conceptos que hemos presentado.

Definición 4.2.1 *Sean A y B conjuntos, se define una* **función de A en B** *y se denota $f : A \to B$ a un subconjunto del producto cartesiano $A \times B$ de la forma $\{(a, f(a)) : a \in A\} \subset A \times B$.*

Definición 4.2.2 *Sean A y B dos conjuntos y $f : A \to B$ una función de A en B:*

Se dice que f es **inyectiva** *si para cada par de elementos $a, b \in A$ que verifican $f(a) = f(b)$ se tiene que $a = b$.*

Se dice que f es **sobreyectiva** *si para cada $b \in B$ existe un elemento $a \in A$ de modo que $b = f(a)$.*

Se dice que f es **biyectiva** *si es inyectiva y sobreyectiva.*

Ejemplos 4.2.3 *Sea $A = B = \mathbb{R}$ y $f : \mathbb{R} \to \mathbb{R}$ que asigna a cada $x \in \mathbb{R}$ el valor de su cuadrado $f(x) = x^2$. Se tiene que f no es inyectiva porque dos números reales distintos, por ejemplo el 2 y el -2, verifican $f(2) = f(-2) = 4$.*

Tampoco es sobreyectiva porque hay números reales (todos los negativos) que no se pueden escribir como el cuadrado de un número real.

Como no es inyectiva (ni tampoco sobreyectiva) no es biyectiva.

Definición 4.2.4 *Sean A y B dos conjuntos, se dice que* **el cardinal de A es igual que el cardinal de B,** *y se escribe $|A| = |B|$, si existe una función biyectiva (o biyección) entre A y B.*

Definiciones 4.2.5 *Un conjunto A se dice* **finito** *si, o bien es el conjunto vacío, o bien existe $n \in \mathbb{N}$ y una biyección $f : A \to \mathbb{N}_n$ entre A y $\mathbb{N}_n$, siendo $\mathbb{N}_n$ el conjunto de los números naturales menores o iguales que n, $\mathbb{N}_n = \{1, 2, ..., n\}$.*

Si A es el conjunto vacío $A = \emptyset$ definimos el **cardinal** *de A como 0, esto se denota:*

$$|\emptyset| = 0.$$

Si A es un conjunto finito no vacío entonces para un cierto $n \in \mathbb{N}$ existe una biyección $f : A \to \mathbb{N}_n$ y definimos el **cardinal** *de A como n:*

$$|A| = n.$$

Observamos que la noción de cardinal está bien definida porque si existe una biyección entre $\mathbb{N}_n$ y $\mathbb{N}_m$ entonces $n = m$.

Ejemplos 4.2.6 *i) Sea A el conjunto de las letras del alfabeto español. La ordenación alfabética determina una biyección entre A y $\mathbb{N}_{29}$. Por tanto $|A| = 29$.*

ii) El cardinal del conjunto de los números naturales, $\mathbb{N}$, no puede ser finito porque no hay ninguna aplicación biyectiva entre $\mathbb{N}$ y un subconjunto suyo de la forma $\mathbb{N}_n$.

Vamos a analizar algunas de las propiedades de los cardinales de conjuntos finitos. Sean en cada caso A y B dos conjuntos de cardinal finito.

Propiedad 1. Si $A \cap B = \emptyset$ entonces $|A \cup B| = |A| + |B|$.

Propiedad 2. Dado el subconjunto $B \subset A$ definimos el complementario de B en A, escrito $\overline{B}$, como $A - B = \{x \in A : x \notin B\}$ entonces

$$|\overline{B}| = |A| - |B|.$$

Propiedad 3. Si $A \subset B$ entonces $|A| \leq |B|$.

Propiedad 4. $|A \cup B| = |A| + |B| - |A \cap B|$.

Propiedad 5. $|A \times B| = |A| \times |B|$.

Esta propiedad 5 se suele denominar *regla del producto.* Establece lo siguiente: si el número de posibilidades para una situación (pongamos la primera letra de una palabra) es n y el número de posibilidades para otra situación (pongamos la segunda letra de una palabra) es m, entonces las posibilidades de las dos situaciones conjuntamente (digamos, palabras de dos letras) es nm.

Propiedad 6. Principio del palomar. Si $|A| > |B|$ entonces no hay una función inyectiva de A en B.

Ejercicio 77 *Demostrar que las propiedades 3 y 6 son equivalentes. Demostrar las propiedades 1 a 5.*

Observaciones 4.2.7 *i) La propiedad 2 aplicada a tres conjuntos finitos se puede generalizar:*

$$|A \cup B \cup C| =$$

$$= |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|.$$

Y se pueden escribir fórmulas análogas para cada unión finita de conjuntos finitos.

ii) El Principio del Palomar se puede generalizar de la siguiente manera: Sean m, n números naturales con $m > n$. Si queremos colocar m objetos en n cajas, en alguna de ella habrá necesariamente una cantidad de objetos mayor o igual que la parte entera del cociente m/n.

iii) También la propiedad del producto cartesiano se generaliza:

$$|A_1 \times A_2 \times ... \times A_n| = |A_1| \times |A_2| \times ... \times |A_n|.$$

Ejemplos 4.2.8 *de aplicación de las propiedades.*

i) Sea A el conjunto de alumnos matriculados en alguna asignatura de primero, B el conjunto de alumnos matriculados en alguna asignatura de segundo. El conjunto A tiene cardinal 250 y el conjunto B tiene cardinal 220. Hay exactamente 50 alumnos que tienen alguna asignatura de primero y alguna de segundo. Entonces el número total de alumnos de primero y segundo sería, aplicando la Propiedad 4:

$$|A \cup B| = 250 + 220 - 50 = 420.$$

ii) Sea P el conjunto de palabras de cuatro letras. Si llamamos A al alfabeto español se tiene:

$$P = A \times A \times A \times A.$$

Con lo cual, usando reiteradamente la Propiedad 5:

$$|P| = 29^4.$$

iii) Si a la segunda letra le pedimos que sea vocal, entonces, el conjunto R de las palabras de cuatro letras cuya segunda letra es una vocal es el producto cartesiano

$$R = A \times V \times A \times A,$$

donde V es el conjunto de las vocales, por tanto

$$|R| = 29^3 \times 5.$$

iv) En cualquier conjunto de 368 personas hay dos cuyos cumpleaños son el mismo día. En efecto, como el conjunto de personas P tiene cardinal mayor que el de días del año D ($368 > 365 = |D|$) entonces, por el principio del palomar, cualquier función de P en D es no inyectiva, es decir hay dos personas que cumplen años el mismo día.

4.3 Variaciones

Partimos del siguiente problema:

Problema. Dada una carrera con 154 participantes, determinar cuántos posibles resultados pueden producirse para el podium (oro, plata y bronce).

Formalmente podemos escribir el problema de una forma equivalente: sea A el conjunto de participantes, que forma un conjunto de 154 elementos y por tanto admite una biyección con $\mathbb{N}_{154}$ (a cada corredor un dorsal numerado). Sea B el conjunto de las tres medallas, que admite una biyección con $\mathbb{N}_3$ (por ejemplo el oro al 1, la plata al 2 y el bronce al 3). Se trata de determinar el número de funciones inyectivas de $\mathbb{N}_3$ en $\mathbb{N}_{154}$.

Problema equivalente. Calcular el número de funciones inyectivas de $\mathbb{N}_3$ en $\mathbb{N}_{154}$.

En efecto, ambos problemas son equivalentes porque una función inyectiva f de $\mathbb{N}_3$ en $\mathbb{N}_{154}$ consiste en dar $f(1) = a \in \mathbb{N}_{154}$, $f(2) = b \in \mathbb{N}_{154}$ y $f(3) = c \in \mathbb{N}_{154}$ que son tres números de dorsal distintos (ya que f inyectiva), justamente los que suben al podium (la persona con el dorsal a gana el oro, la de dorsal b la plata y la de c el bronce).

Observamos que para el ganador del oro hay 154 posibilidades (cualquier corredor); una vez que se ha establecido quién ha ganado el oro, para el ganador de la medalla de plata hay 153 posibilidades; y sabiendo el ganador del oro y la plata tenemos 152 resultados posibles para el bronce. De este modo hay $154 \times 153 \times 152$ posibles resultados de la carrera o equivalentemente hay $154 \times 153 \times 152$ funciones inyectivas de $\mathbb{N}_3$ en $\mathbb{N}_{154}$.

Esto nos permite plantear el problema general que atañe al concepto de **variaciones**.

Problema. Determinar el número $V_{m,n}$ de funciones inyectivas del conjunto $\mathbb{N}_n$ en el conjunto $\mathbb{N}_m$.

Observación 4.3.1 *El Principio del palomar señala que si $n > m$ entonces no hay tales funciones inyectivas, esto es, $V_{m,n} = 0$.*

Una generalización sencilla de los razonamientos del ejemplo anterior indica que si $m \geq n$:

$$V_{m,n} = m \times (m-1) \times ... \times (m-n+1) = m!/(m-n)!.$$

Ejemplo 4.3.2 *El número de funciones inyectivas que se pueden establecer entre* $\mathbb{N}_5$ *y* $\mathbb{N}_{10}$ *es:*

$$V_{10,5} = 10 \times 9 \times 8 \times 7 \times 6.$$

Definición 4.3.3 *Al número* $V_{m,n}$ *de funciones inyectivas entre un conjunto de cardinal* n *y otro de cardinal* m *con* $m \geq n$ *se le llama* **número de variaciones de** m **elementos tomados de** n **en** n *y se tiene*

$$V_{m,n} = m \times (m-1) \times ... \times (m-n+1).$$

Ejemplos 4.3.4 *1. Determinar de cuántas maneras se pueden elegir el presidente y el vicepresidente de una asociación de 300 miembros (funciones inyectivas de* $\mathbb{N}_2$ *en* $\mathbb{N}_{300}$*).*

$$V_{300,2} = 300 \times 299$$

2. De cuántas maneras se puede elegir un equipo de cuatro relevistas para correr en la carrera de 4×100 *entre los 10 seleccionados (entendiendo que hay que indicar el orden en el que se corre).*

$$V_{10,4} = 10 \times 9 \times 8 \times 7$$

3. Si el corredor más rápido del ejemplo anterior tiene que correr necesariamente en primer lugar, entonces el problema es calcular los equipos de tres relevistas entre los 9 seleccionados que quedan.

$$V_{9,3} = 9 \times 8 \times 7$$

De esta manera, de las definiciones y de la observación de los ejemplos se deduce que cuando calculamos $V_{m,n}$ estamos contando exactamente las **selecciones ordenadas de** n **elementos en un conjunto de** m **elementos.**

Resumiendo, es lo mismo:

- El número de aplicaciones inyectivas de $\mathbb{N}_n$ en $\mathbb{N}_m$.

- El número de selecciones ordenadas de n elementos en un conjunto de m elementos.
- El número de variaciones de m elementos tomadas de n en n, esto es,

$$V_{m,n} = 0 \quad si \quad m < n$$

$$V_{m,n} = m(m-1)...(m-n+1) \quad si \quad m \geq n.$$

Modifiquemos un poco las definiciones.

Definición 4.3.5 *Se llama* **número de variaciones con repetición de** m **elementos tomados de** n **en** n *al número* $VR_{m,n}$ *de funciones de* $\mathbb{N}_n$ *en* $\mathbb{N}_m$ *y se tiene*

$$VR_{m,n} = m^n.$$

De modo que la hipótesis que ha variado con respecto a la definición anterior (de $V_{m,n}$) es la inyectividad. Veamos un ejemplo donde se aplica este concepto.

Ejemplo 4.3.6 *Determinar cuántas palabras formadas por 4 bytes se pueden construir. O equivalentemente cuántos números enteros se pueden almacenar con 4 bytes.*

El conjunto de las palabras formadas por cuatro bytes, es decir, por cuatro elementos del conjunto $B = \{0, 1\}$ *es*

$$B \times B \times B \times B$$

y por tanto, usando la propiedad del cardinal de un producto cartesiano, son $2^4 = 16$ *palabras.*

Mostremos que, en efecto, una palabra de 4 bytes es una función de $\mathbb{N}_4$ en $\mathbb{N}_2$. Tomemos una biyección b entre B y $\mathbb{N}_2$, por ejemplo, $b(0) = 1$ y $b(1) = 2$. Entonces una función f de $\mathbb{N}_4$ en $\mathbb{N}_2$ asigna valores a $f(1)$, $f(2)$, $f(3)$ y $f(4)$ en $\mathbb{N}_2$. De esta manera $f(1)$ indica la primera letra de la palabra y, respectivamente, $f(2)$ la segunda, $f(3)$ la tercera y $f(4)$ la cuarta. Por ejemplo, sea la función:

$$f : \mathbb{N}_4 \to \mathbb{N}_2$$

$$f(0) = 2$$
$$f(1) = 2$$
$$f(2) = 1$$
$$f(3) = 1$$

Corresponde a la palabra 1100. Como en este caso se permiten palabras con letras repetidas, las funciones que estamos contando no tienen por qué ser inyectivas. Por ejemplo la palabra 0000 responde a la función constante $g : \mathbb{N}_4 \to \mathbb{N}_2$ definida como $g(a) = 1$ para cada $a \in \mathbb{N}_4$.

Es decir, ahora lo que estamos contando son **selecciones ordenadas de n elementos (posiblemente repetidos) de un conjunto de m elementos**.

Ejercicio 78 *Con el alfabeto español de 29 letras:*

i) Determinar el número de palabras de 5 letras.

ii) Determinar el número de palabras de 5 letras sin ninguna repetida.

iii) Del conjunto de palabras de i) determinar cuántas de ellas contienen a la letra b.

iv) Idem que iii) con el conjunto de palabras de ii).

v) Determinar cuántas palabras del conjunto i) empiezan por vocal. Idem para las del conjunto ii).

vi) Determinar el número de palabras de i) que empiezan por vocal y terminan por z. Idem con ii).

vii) Determinar el número de palabras de i) cuyas tres primeras letras son vocales. Idem con ii).

Ejercicio 79 *i) Determinar la cantidad de números de 4 cifras que existen.*

ii) Determinar cuántos de ellos son pares.

iii) Cuántos son impares.

iv) Cuántos son múltiplos de 5.

v) Cuántos contienen la cifra 2.

vi) Cuántos contienen la cifra 2 o la cifra 3.

Resumiendo, es lo mismo:

- El número de aplicaciones de $\mathbb{N}_n$ en $\mathbb{N}_m$.

- El número de selecciones ordenadas de n elementos (admitiendo repeticiones) en un conjunto de m elementos.
- El número de variaciones con repetición de m elementos tomadas de n en n, esto es,
$$VR_{m,n} = m^n.$$

4.4 Permutaciones

Un caso particular de las variaciones resulta cuando $m = n$, es decir, cuando tratamos de contar el número de funciones inyectivas de $\mathbb{N}_n$ en $\mathbb{N}_n$. La observación siguiente muestra que entonces debemos contar las funciones biyectivas (o biyecciones) de $\mathbb{N}_n$ en $\mathbb{N}_n$.

Observación 4.4.1 *Cada aplicación inyectiva $f : \mathbb{N}_n \to \mathbb{N}_n$ es una biyección. En efecto, si $f : \mathbb{N}_n \to \mathbb{N}_n$ es una aplicación inyectiva, entonces el conjunto imagen de f, esto es, los elementos que se pueden escribir como $f(a)$ para $a \in N_n$, es un subconjunto de $\mathbb{N}_n$ de cardinal n, por tanto es todo $\mathbb{N}_n$.*

Ejemplo 4.4.2 *De cuántas maneras se pueden sentar en una mesa presidencial con cinco sillas los cinco miembros de la comisión.*

En efecto, es un ejemplo de la situación anterior porque si las sillas están numeradas del 1 al 5, entonces el conjunto de sillas se puede ver como $\mathbb{N}_5$ y si los comensales están numerados del 1 al 5 también el conjunto de comensales se puede interpretar como $\mathbb{N}_5$. Se trata entonces de establecer cuántas funciones inyectivas (a cada comensal su silla y no a dos comensales la misma) de $\mathbb{N}_5$ en $\mathbb{N}_5$ se pueden construir. A la postre si a cada comensal se le asigna una silla distinta, la aplicación será biyectiva (es sobreyectiva porque no sobran sillas). Entonces:

$$V_{5,5} = 5 \times 4 \times 3 \times 2 \times 1.$$

Ejemplo 4.4.3 *Sea el conjunto de los tres candidatos $A = \{Alicia, Ernesto, Sara\}$ a los tres cargos que son $B = \{presidente, secretario, vocal\}$. Calcular de cuántas maneras se pueden dar estos tres cargos a estas tres personas.*

Cada biyección entre A y B asigna a cada persona un cargo distinto. Por ejemplo, denotando los nombres con su inicial e igualmente los cargos, tenemos las 6 *posibles biyecciones:*

$$f_1(A) = p, f_1(E) = s, f_1(S) = v$$
$$f_2(A) = p, f_2(E) = v, f_2(S) = s$$
$$f_3(A) = s, f_3(E) = p, f_3(S) = v$$
$$f_4(A) = s, f_4(E) = v, f_4(S) = p$$
$$f_5(A) = v, f_5(E) = s, f_5(S) = p$$
$$f_6(A) = v, f_6(E) = p, f_6(S) = s,$$

que interpretadas como las listas ordenadas de las imágenes nos dan las 6 posibles ordenaciones distintas del conjunto B

$$\{p, s, v\}$$
$$\{p, v, s\}$$
$$\{s, p, v\}$$
$$\{s, v, p\}$$
$$\{v, s, p\}$$
$$\{v, p, s\}.$$

Definición 4.4.4 *El número P_n cardinal del conjunto de biyecciones del conjunto $\mathbb{N}_n$ en el conjunto $\mathbb{N}_n$ se denomina* **número de permutaciones de n elementos** *y se tiene*

$$P_n = V_{n,n} = n! = n(n-1)...1.$$

Y justamente lo que estamos contando son las distintas **formas de ordenar un conjunto de n elementos**. Cada posible ordenación de un conjunto A es una permutación de A.

Observación 4.4.5 *Por definición* $0! = 1$.

Ejercicio 80 *De cuántas maneras se pueden sentar 6 personas en una mesa de 6 sillas. Estudiar lo que ocurre si la mesa es redonda y lo único que queremos tener en cuenta es la posición relativa, es decir, quién está a la derecha y quién a la izquierda, independientemente de la silla que se ocupa.*

Resumiendo, es lo mismo:

- El número de aplicaciones biyectivas de $\mathbb{N}_n$ en $\mathbb{N}_n$.
- El número de ordenaciones distintas de un conjunto de n elementos.
- El número de permutaciones de un conjunto de n elementos

$$P_n = n!.$$

- El número de variaciones de n elementos tomadas de n en n.

Podemos introducir ahora una pequeña variante en el problema del cálculo de las distintas ordenaciones.

Problema. Estudiar cuántos números distintos se pueden construir reordenando las cifras del 121.

Es claro que el problema no se resuelve calculando el número de permutaciones de 3 elementos, ya que $P_3 = 6$ y, sin embargo, los números obtenidos como resultado de reordenar las cifras del 121 son:

$$121, \quad 211, \quad 112$$

esto es, son solamente 3.

La razón por la que no son 6 las permutaciones es porque el número 1 está repetido 2 veces y, por tanto, cuando permutamos los unos entre sí, el número no varía. Es decir, si en el número 121 intercambiamos la primera y la tercera cifra obtenemos el mismo número, que debe sólo contarse una vez.

De esta manera, como las maneras de reordenar los unos son exactamente dos, se debe dividir por 2 para contarlas sólo una vez.

$$Total := P_3/2 = 3.$$

Y generalizando a una lista de n elementos donde hay s elementos distintos que se van repitiendo n_1 veces el primero, ..., n_s veces el último, tenemos:

Definición 4.4.6 *En las condiciones del párrafo anterior definimos el* **número de permutaciones con repetición de** n **elementos, donde hay** s **elementos que se repiten** $n_1 > 1$**,** $n_2 > 1$**, ...,** $n_s > 1$ **veces respectivamente**, *como el número* $PR_n^{n_1,n_2\ldots,n_s}$ *de distintas ordenaciones de esa lista con elementos repetidos. Se calcula mediante la fórmula:*

$$PR_n^{n_1,\ldots,n_s} = \frac{n!}{(n_1)!\ldots(n_s)!}.$$

Ejemplo 4.4.7 *Determinar las palabras de 9 letras que se pueden construir como resultado de ordenar las letras de la palabra cocodrilo. El resultado es:*

$$PR_9^{2,3} = \frac{9!}{2!3!}$$

donde el 2 proviene de las dos ces y el tres de las tres oes.

Ejercicio 81 *El nuevo sistema de matriculación de automóviles establece una matrícula formada por tres consonantes seguidas de 4 cifras.*

i) Determinar el número de matrículas diferentes que existen.

ii) Determinar cuántas matrículas hay con las mismas letras y números que la matrícula:

$$BBK \quad 1224$$

iii) Determinar cuántas matrículas empiezan por C *y terminan por 1.*

iv) Determinar cuántas matrículas son capicúas en la parte de los dígitos.

v) Determinar cuántas matrículas hay con sus tres letras iguales.

vi) Idem que ii) con la matrícula

$$BBC \quad 1122.$$

Ejercicio 82 *Establecemos un juego de azar en el que se venden boletos de 5 cifras (permitiendo que empiecen por 0) y en el que se extrae un número de 5 cifras y se premian todos los boletos que contienen exactamente las mismas cifras que el número extraído. Por ejemplo si sale el* 12345*, es premiado también el* 21345*; si sale el* 22345 *también se premia el* 23245*.*

i) Determinar el número de boletos premiados si sale el 12345*.*

ii) Determinar el número de boletos premiados si sale el 22344

iii) Determinar el número de boletos premiados si sale el 11222*.*

iv) Si el sistema de premios se establece en proporción al dinero recaudado, por ejemplo la tercera parte de la recaudación se reparte entre los ganadores. Determinar cuál es el tipo de boletos que tienen menos posibilidades de salir y cuál es el tipo que más; observar que cuando la dificultad de salir aumenta también lo hace el posible premio.

Resumiendo, es lo mismo:

- El número de ordenaciones de una lista de n elementos con s elementos repetidos, $n_1 > 1$ veces el primero, ..., $n_s > 1$ veces el último.
- El número de permutaciones con repetición de n elementos con las repeticiones que se indican:

$$P_n^{n_1,\ldots,n_s} = \frac{n!}{n_1!...n_s!}.$$

4.5 Combinaciones

El último concepto de combinatoria que vamos a definir es el de *combinaciones* que hace referencia al siguiente problema:

Problema. Dada una clase de 85 alumnos, de cuántas maneras se puede elegir el trío que irá a las reuniones del claustro.

Obsérvese que a diferencia del problema de las variaciones (por ejemplo elección de delegado, subdelegado y vocal) ahora los tríos

$$\{Manuel, Juan, Felipe\}$$

y

$$\{Juan, Manuel, Felipe\}$$

son el mismo (los alumnos elegidos para ir al claustro son los mismos tres), esto es, no influye el orden (mientras que en el problema de la elección de delegado, el trío primero indica que $Manuel$ es el delegado y en el segundo, sin embargo, es $Juan$ el delegado, por lo que son distintos).

Formalmente se puede plantear el siguiente problema equivalente:

Problema. Determinar cuántos subconjuntos de 3 elementos tiene un conjunto de 85 elementos, es decir, de $\mathbb{N}_{85}$.

Si en lugar de resolver el problema planteado, calculamos $V_{85,3}$ entonces estamos computando, no los subconjuntos, sino las selecciones ordenadas de tres elementos. De esta manera cada subconjunto de 3 elementos lo estamos contando más de una vez. Exactamente lo contamos tantas veces como maneras distintas haya de reordenarlo, es decir, P_3 veces. Con un ejemplo: los seis trios siguientes:

$$\{Manuel, Juan, Felipe\}$$

$$\{Manuel, Felipe, Juan\}$$

$$\{Juan, Manuel, Felipe\}$$

$$\{Juan, Felipe, Manuel\}$$

$$\{Felipe, Juan, Manuel\}$$

$$\{Felipe, Manuel, Juan\}$$

son el mismo subconjunto de 3 elementos, ordenados sus elementos de todas las maneras posibles. Por tanto el problema se resuelve computando:

$$Total : \frac{V_{85,3}}{P_3}.$$

Para, efectivamente, sólo contar cada subconjunto una vez.

Definición 4.5.1 *Llamamos* **número de combinaciones de** m **elementos tomados de** n **en** n*, que escribimos* $C_{m,n}$*, al número de subconjuntos de* n *elementos en* $\mathbb{N}_m$*, se calcula*

$$C_{m,n} = \frac{V_{m,n}}{P_n} = \frac{m!}{n!(m-n)!}.$$

Notación. Escribiremos $C_{m,n}$ como $\binom{m}{n}$.

Observación 4.5.2 *Por definición* $0! = 1$ *y* $\binom{m}{0} = 1$.

Definición 4.5.3 *Al número* $\binom{m}{n}$ *se le denomina* **número combinatorio** m **sobre** n.

Ejercicio 83 *Comprobar la fórmula anteriormente escrita:*

$$\binom{m}{n} = \frac{m!}{n!(m-n)!}.$$

Ejemplos 4.5.4 *1) Determinar el número de posibles equipos de baloncesto (5 miembros) que se pueden formar con 10 personas.*

$$C_{10,5} = \frac{V_{10,5}}{P_5} = \frac{10 \times 9 \times 8 \times 7 \times 6}{5 \times 4 \times 3 \times 2}.$$

2) Determinar en cuántos de ellos juega el jugador llamado Martín. (Sólo un jugador se llama Martín.)

$$C_{10,5} - C_{9,5} = \binom{10}{5} - \binom{9}{5}$$

que es calcular el número total de equipos y restar el número de equipos en los que no juega Martín.

Resumiendo, es lo mismo:

- El número de subconjuntos de n elementos de $\mathbb{N}_m$.
- El número de selecciones no ordenadas de n elementos en un conjunto de m elementos.
- El número de combinaciones de m elementos tomadas de n en n, esto es,

$$C_{m,n} = 0 \quad si \quad m < n$$

$$C_{m,n} = \binom{m}{n} = \frac{V_{m,n}}{P_n} \quad si \quad m \geq n.$$

Algunas propiedades de los números combinatorios

Sean m y n números enteros $m \geq n$ entonces:

$$1) \quad \binom{m}{n} = \binom{m}{m-n}.$$

$$2) \quad \binom{m}{n} = \binom{m-1}{n-1} + \binom{m-1}{n}.$$

$$3) \binom{n}{0} + \binom{n}{1} + ... + \binom{n}{n} = 2^n.$$

La propiedad 1) viene de observar el siguiente hecho: interpretemos el número combinatorio $\binom{m}{n}$ como el número de subconjuntos de n elementos de un conjunto A de m elementos. Elegir uno de estos subconjuntos $B \subset A$ consiste en determinar los n elementos que conforman B o equivalentemente señalar los $m-n$ elementos que conforman $A-B$. Por tanto en A hay exactamente tantos subconjuntos de n elementos como de $m-n$ elementos.

La propiedad 2) se deduce de lo siguiente. Interpretemos de nuevo $\binom{m}{n}$ como el número de subconjuntos de n elementos en A ($|A| = m$). Fijamos un elemento $a \in A$. Los subconjuntos de n elementos de A se dividen entonces en los que contienen a a y los que no lo contienen.

El número de subconjuntos de n elementos de A que contienen a a es exactamente:

$$\binom{m-1}{n-1}$$

ya que puede interpretarse como el número de subconjuntos de $n-1$ elementos de $A - \{a\}$.

El número de subconjuntos de n elementos de A que no contienen a a es exactamente:

$$\binom{m-1}{n}$$

ya que puede interpretarse como el número de subconjuntos de n elementos de $A - \{a\}$.

Por tanto la cantidad total de subconjuntos de n elementos es:

$$\binom{m}{n} = \binom{m-1}{n-1} + \binom{m-1}{n}.$$

La propiedad 3) se puede deducir de la fórmula del **binomio de Newton**, que recordamos:

$$(a+b)^n = \binom{n}{0}a^n + \binom{n}{1}a^{n-1}b + ... + \binom{n}{n-1}ab^{n-1} + \binom{n}{n}b^n,$$

es decir:

$$(a+b)^n = \sum_{k=0}^{n}\binom{n}{k}a^{n-k}b^k.$$

Esta fórmula se puede demostrar por inducción sobre n usando las propiedades de los números combinatorios.

Si aplicamos esta fórmula a $a = 1$ y $b = 1$ tenemos:

$$(1+1)^n = 2^n = \binom{n}{0} + \binom{n}{1} + ... + \binom{n}{n-1} + \binom{n}{n}.$$

Esta identidad muestra que, si llamamos **partes de** A, denotado $P(A)$, al conjunto formado por todos los subconjuntos de A, el cardinal de las partes de A verifica:

$$|P(A)| = 2^{|A|}.$$

Porque el conjunto total de subconjuntos se puede escribir como unión de los subconjuntos de 0 elementos con los de un elemento, con los de 2... hasta llegar al único subconjunto de A de $|A|$ elementos que es el propio A. Esta última fórmula también se puede demostrar por inducción sobre el cardinal de A.

Una vez vistas estas propiedades de los números combinatorios veamos como se puede modificar el problema de las combinaciones.

Problema. Sean x, y, z tres indeterminadas, llamamos *monomio de grado* d a una expresión del tipo $x^a y^b z^c$ con $a + b + c = d$. Determinar el número de monomios diferentes de grado 5 en dichas tres indeterminadas.

Observamos que un monomio de grado 5 es por ejemplo x^3yz que podría también escribirse como x^2yxz, o de cualquier otra manera posible resultado de reordenar las variables (siempre tomando tres veces la x, una vez la y y una vez la z). Esto muestra que el orden de las selecciones no es relevante.

Veamos cómo contar estos monomios de forma inteligente. Como podemos reordenar las variables podemos suponer que cada monomio de grado 5 está escrito como en la definición:

$$x^a y^b z^c \quad a+b+c=5$$

de modo que se le puede asignar una palabra formada por unos y ceros de la siguiente manera

$$x^3yz \to 1110101.$$

El número de unos hasta el primer 0 indica cuál es el exponente de la x, después del primer 0 y hasta el segundo 0 hay un 1 que indica el exponente de la y y el último 1 el de la z. Otro ejemplo:

$$x^4y \to 1111010$$

El problema es por tanto dónde colocar esos cinco unos que corresponden a que el grado del monomio es cinco. Hay que elegir cinco lugares entre 7 posibles para colocar los unos, esto es, estamos computando los subconjuntos de cinco elementos de un conjunto de 7 elementos.

$$Total: \binom{7}{5} = 21$$

Los podemos escribir:

$$x^5, x^4y, x^3y^2, x^2y^3, xy^4,$$
$$y^5, y^4z, y^3z^2, y^2z^3, yz^4,$$
$$z^5, x^4z, x^3z^2, x^2z^3, xz^4,$$
$$x^3yz, xy^3z, xyz^3,$$
$$x^2y^2z, xy^2z^2, x^2yz^2.$$

Este mismo razonamiento permite calcular la fórmula general.

Definición 4.5.5 *Sea un conjunto de m elementos, definimos el número de* **combinaciones con repetición de m elementos tomados de n en n** *al número de selecciones no ordenadas de n elementos (con posibles repeticiones) en un conjunto de m elementos. Se denota $CR_{m,n}$ y se calcula mediante la fórmula:*

$$CR_{m,n} = \binom{m+n-1}{n}$$

Ejemplos 4.5.6 *1) Determinar cuántos posibles resultados pueden acontecer al lanzar tres dados indistinguibles simultáneamente:*

$$CR_{6,3} = \binom{8}{3} = 56.$$

2) Si se extraen simultáneamente cinco cartas de cinco barajas españolas (40 cartas) el número de posibilidades es:

$$CR_{40,5} = \binom{44}{5}.$$

Ejercicio 84 *1) Dada una nube de 312 puntos diferentes, donde nunca tres de ellos están alineados, determinar el número de triángulos distintos que se pueden formar con dichos puntos.*

2) Si entre los puntos de 1) existen 5 puntos que están alineados, computar cuántos triángulos se pueden formar.

Ejercicio 85 *1) Sean 6 bombos que contienen cada uno de ellos 5 bolas numeradas del 1 al 5 de modo que las bolas de bombos distintos con el mismo número son indistinguibles entre sí. Por un proceso mecánico cada bombo deposita en una cesta una bola, simultáneamente con el resto de bombos. Determinar el número de resultados posibles.*

2) Determinar cuántos de estos resultados contienen al menos una bola numerada con el 5.

Resumiendo, es lo mismo:

- El número de selecciones no ordenadas (con elementos repetidos) de n elementos en un conjunto de m elementos.

- El número de combinaciones con repetición m elementos tomados de n en n:

$$CR_{m,n} = \binom{m+n-1}{n}.$$

4.6 Probabilidad

Una de las aplicaciónes fundamentales de la combinatoria es el cálculo de probabilidades, es decir, estimar con qué seguridad va a ocurrir un suceso. La regla básica por la que se rigen los experimentos que tienen una cantidad finita y equiprobable de resultados es que la probabilidad de que ocurra un suceso determinado es el cociente del número de casos en los que acontece este suceso entre la cantidad total de casos posibles. Así al tirar un dado perfecto la probabilidad de que salga un 6 es 1/6: los posibles resultados conforman el conjunto $\{1, 2, 3, 4, 5, 6\}$ y el caso que nos interesa (el 6) es uno de estos 6 posibles resultados. La idea de fondo es que si uno tirara un dado una cantidad muy alta de veces, le saldrían (más o menos) la misma cantidad de resultados 1, que 2, ..., que 6. Esta ley se suele conocer como Ley de los grandes números.

Podemos complicar un poco el problema y pedir la probabilidad de que al tirar dos dados consecutivamente se obtenga, por ejemplo, una pareja de seises. La combinatoria es el instrumento adecuado para hacer este cálculo. La cantidad total de resultados es $VR_{6,2} = 36$, el resultado $(6, 6)$ es uno de ellos, por tanto su probabilidad es 1/36. Observemos sin embargo que la probabilidad de que salga un 3 y un 2 es justamente el doble ya que son dos los resultados posibles: $(2, 3)$ y $(3, 2)$.

Este tipo de estudio es fundamental para asignar premios en juegos de azar de manera que sea interesante jugar y rentable el juego: piénsese, por ejemplo, en una máquina tragaperras. Para establecer el precio de una poliza de seguros en función de las características del usuario...

Definamos con precisión los conceptos.

4.6.1 Nociones básicas

Definición 4.6.1 *Un* **experimento aleatorio** *es un procedimiento cuyos posibles resultados forman un conjunto conocido y tal que al efectuar el experimento el resultado obtenido depende del azar.*

Observación 4.6.2 *El término azar proviene del vocablo árabe az-zahr que significa el dado para jugar. Se podrían introducir interesantes consideraciones filosóficas sobre este concepto. Desde el punto de vista de las ciencias de la computación, en algún sentido, azar y algoritmo son antónimos. Es muy interesante el desarrollo de algoritmos para modelar experimentos aleatorios.*

Ejemplos 4.6.3 *Lanzar un dado es un experimento aleatorio porque el resultado (un número del 1 al 6) del lanzamiento depende del azar.*

La extracción de una carta de una baraja.

La extracción consecutiva y sin reposición de dos bolas numeradas de un bombo que contiene cuatro bolas con los números 1 al 4.

La elección de tres nombres por sorteo en un grupo de 100 personas.

Definición 4.6.4 *Se llama* **espacio muestral** *de un experimento aleatorio y se denota Ω al conjunto de posibles resultados de dicho experimento aleatorio.*

Ejemplos 4.6.5 *En el experimento aleatorio del lanzamiento de un dado, su espacio muestral es $\Omega = \{1, 2, 3, 4, 5, 6\}$.*

En la extracción de una carta de una baraja, (digamos española con 40 cartas), el espacio muestral Ω es un conjunto de cardinal 40 formado por cada una de las cartas de la baraja.

En la extracción de las bolas numeradas el espacio muestral es Ω igual a las parejas ordenadas (sin elementos repetidos) de un conjunto de 4 elementos, esto es, $|\Omega| = 4 \times 3$.

El conjunto de tríos de un conjunto de 100 personas es el espacio muestral Ω del último experimento aleatorio de los ejemplos y tiene cardinal $|\Omega| = \binom{100}{3}$.

Definición 4.6.6 *Se denomina* **suceso** *a cada subconjunto del espacio muestral Ω de un experimento aleatorio.*

Observación 4.6.7 *Nos centraremos en experimentos cuyo espacio muestral es un conjunto finito.*

Ejemplos 4.6.8 *En el espacio muestral del lanzamiento del dado podemos elegir el suceso S:* sacar un uno, *correspondiente al subconjunto unitario $S = \{1\}$. O el suceso S':* sacar un número par, *correspondiente a $S' = \{2, 4, 6\}$.*

4.6.2 Espacios muestrales homogéneos

Comencemos estudiando espacios muestrales donde todos los elementos tienen la misma probabilidad de acontecer.

Definición 4.6.9 *Sea E un experimento aleatorio, Ω su espacio muestral, de cardinal finito y de modo todos los elementos de Ω son igualmente posibles. Se define la* **probabilidad de un suceso** $S \subset \Omega$ *como el cociente:*

$$p(S) = |S|/|\Omega|.$$

Es la **Regla de Laplace** que indica que la probabilidad de un suceso la da el cociente del número de **casos favorables** entre el de **casos posibles**.

Ejemplos 4.6.10 *1) Determinar cuál es la probabilidad de acertar 14 en las quinielas. El número de casos favorables es 1, el de los resultados de los partidos, y el espacio muestral es Ω el conjunto de listas de longitud 14 donde cada elemento de la lista es un 1, una x o un 2. Así la probabilidad de acertar una quiniela es*

$$\frac{1}{3^{14}}.$$

2) Cuál es la probabilidad de ganar en la lotería tradicional el premio gordo. El espacio muestral son los números de 5 cifras, por tanto son 100.000, así la probabilidad de ganar es

$$1/10^5.$$

3) En la lotería primitiva se tienen que elegir 6 números entre 49 por tanto la probabilidad de ganar es:

$$1/\binom{49}{6}.$$

4) Se extraen 3 cartas de una baraja española (4 palos, numeradas con las cifras del 1 al 7 y sota, caballo y rey de cada palo), determinar la probabilidad de que las tres cartas extraidas sean la misma carta en distinto palo. El número de casos posibles es $\binom{40}{3}$. Los casos favorables se calculan de la siguiente manera: supongamos que fijo el número 1 (los ases). Entonces la baraja contiene 4 ases, por tanto, exactamente hay $\binom{4}{3} = 4$ posibles tríos de ases. Como tenemos 10 posibles números hay $10 \times 4 = 40$ casos favorables, por tanto

$$40/\binom{40}{3}.$$

Ejercicio 86 *Calcular la probabilidad de acertar 13 y no 14 en las quinielas. Idem para acertar 5 y no 6 en la lotería primitiva.*

Podemos entonces usar las propiedades de los cardinales de conjuntos finitos para calcular probabilidades de sucesos definidos como uniones, intersecciones y complementarios.

Propiedad 1. El suceso Ω. Consideremos el suceso dado por el conjunto Ω, es decir, todo el espacio muestral, se verifica:

$$p(\Omega) = |\Omega|/|\Omega| = 1.$$

Propiedad 2. Subconjuntos. Si $S \subset T \subset \Omega$ entonces $|S| \leq |T| \leq |\Omega|$ y $|\emptyset| = 0$. Por tanto:

$$0 \leq p(S) \leq p(T) \leq 1.$$

La probabilidad de cada suceso es entonces un número comprendido entre 0 y 1.

Propiedad 3. Unión de sucesos. Tomemos dos sucesos S y T entonces

$$p(S \cup T) = p(S) + p(T) - p(S \cap T).$$

Propiedad 4. Complementario de un suceso. Sea $S \subset \Omega$ un suceso entonces el suceso complementario es

$$\overline{S} = \Omega - S = \{x \in \Omega : \ x \notin S\}$$

y su probabilidad es:

$$P(\overline{S}) = 1 - P(S).$$

Esta propiedad 4 es consecuencia de las anteriores ya que

$$S \cup \overline{S} = \Omega \qquad S \cap \overline{S} = \emptyset.$$

Ejemplos 4.6.11 *i) Sea el experimento aleatorio consistente en lanzar dos dados sucesivamente, calcular la probabilidad de que la suma de los resultados sea 4. El suceso S:* dos cifras que suman 4 *se puede poner como unión disjunta del suceso S':* sacar dos doses, *y S'':* sacar un 1 y un 3 *($S' \cap S'' = \emptyset$). De modo que:*

$$p(S) = \frac{1}{36} + \frac{2}{36}.$$

ii) Sea un equipo de gimnasia de 10 miembros de los cuales sólo uno se apellida Muñoz. Determinar la probabilidad de que en un equipo de 5 miembros elegidos entre los 10 esté Muñoz. Sea el suceso S: Muñoz no está en el equipo *(complementario del suceso del que queremos calcular su probabilidad), el número de equipos en los que no está Muñoz es el número de equipos formados con los jugadores restantes, entonces :*

$$1 - p(S) = 1 - \binom{9}{5} : \binom{10}{5}.$$

4.6.3 Espacios muestrales heterogéneos

La regla de Laplace ha funcionado para asignar probabilidades en un espacio muestral donde todos sus elementos son equiprobables, pero además ha mostrado cómo asignar probabilidades en un espacio muestral donde los elementos de Ω no sean necesariamente equiprobables, esto es, de naturaleza heterogénea.

Así, sea Ω el espacio muestral finito de un experimento aleatorio de modo que

$$\Omega = \{a_1, a_2, ..., a_s\}.$$

Si no todos los elementos del espacio muestral son equiprobables podemos asignar (en función de observaciones, por ejemplo) probabilidades a cada elemento del espacio muestral, y deben verificar:

i) $1 \geq p(a_i) \geq 0$ para cada $i \in \{1, ..., s\}$.
ii) $p(a_1) + p(a_2) + ... + p(a_s) = 1$.

Para determinar la probabilidad de un suceso $S \subset \Omega$, sencillamente sumamos la probabilidad de los elementos que la conforman, coherentemente con la propiedad 3 antes señalada:

$$p(S) = \Sigma_{a_i \in S} p(a_i).$$

Las propiedades 1 a 4 antes comentadas son coherentes, por i) y ii), con esta definición.

Observación 4.6.12 *En los casos donde todos los elementos de Ω son equiprobables, para cada $a_i \in \Omega$ se tiene*

$$p(a_i) = 1/|\Omega|.$$

Ejemplo 4.6.13 *Durante varios días observamos que en un juego de dados el resultado 1 sale el doble de veces que el resultado 2, el 2 sale el doble de veces que los demás, mientras los otros resultados salen todos más o menos la misma cantidad de veces, entonces podemos asignar las probabilidades de la siguiente forma:*

$$p(3) = a$$

$$p(4) = a$$

$$p(5) = a$$

$$p(6) = a$$

$$p(2) = 2a$$

$$p(1) = 4a$$

Como $1 \geq a \geq 0$ *y* $4a + 2a + a + a + a + a = 10a = 1$ *entonces* $a = 1/10$. *De este modo:*

$$p(1) = 4/10$$

$$p(2) = 2/10$$

$$p(i) = 1/10 \quad i = 3, 4, 5, 6.$$

Y para calcular la probabilidad de cualquier suceso S *bastará escribirlo como unión disjunta de sus subconjuntos unitarios. Por ejemplo sea* S *el suceso, el resultado sea par, entonces* $S = \{2, 4, 6\}$, *por lo que*

$$p(S) = p(2) + p(4) + p(6) = 2/10 + 1/10 + 1/10 = 4/10.$$

Ejercicio 87 *Sea un conjunto de 15 cartas de barajas iguales, todas del mismo palo: hay 3 ases, 3 doses, 1 tres, 4 cuatros, 2 caballos y 2 reyes. Sea el experimento aleatorio extraer una carta al azar de las 15. Determinar el espacio muestral* Ω *y asignar probabilidades razonables a cada elemento de* Ω. *Calcular la probabilidad del suceso la carta extraída sea una figura.*

4.6.4 Tratamiento numérico de la información: Variables Aleatorias

Vamos a introducir las *variables aleatorias* como un método probabilístico para estudiar características numéricas de conjuntos.

Definición 4.6.14 *Una* **variable aleatoria** *es una función del espacio muestral de un experimento aleatorio en los números reales.*

Ejemplo 4.6.15 *Tomamos un conjunto de 10 personas. Sea el experimento aleatorio consistente en seleccionar una de estas 10 personas al azar, por tanto Ω es el conjunto de las 10 personas. Definimos la variable aleatoria H como la función $H : \Omega \to \mathbb{R}$ que asigna a cada persona su altura. (Otros ejemplos son la asignación de peso, de número de asignaturas en que está matriculado...)*

Con esta definición de variable aleatoria, podemos mostrar dos instrumentos de la estadística descriptiva: la **esperanza matemática** que hace referencia al valor esperado que toma la variable aleatoria y la **varianza** que es una medida de la dispersión de los valores que toma la variable.

Definiciones 4.6.16 *Sea $X : \Omega \to \mathbb{R}$ una variable aleatoria.*

Se define la **esperanza matemática** *de X como:*

$$E(X) = \sum_{s \in \Omega} p(s)X(s).$$

Se define la **varianza** *de X como:*

$$V(X) = \sum_{s \in \Omega} p(s)(X(s) - E(X))^2.$$

Se define la **desviación típica** *de X como:*

$$\sigma(X) = \sqrt{V(X)}.$$

Estas medidas describen la variable aleatoria, indicando su valor esperado o medio, e indicando lo dispersos o concentrados que están sus valores.

Ejemplo 4.6.17 *Sea un conjunto de 40 alumnos cuyos resultados en un examen de Matemáticas es el siguiente (la nota es un número entero del 0 al 10):*

Nota	Número de alumnos	Probabilidad
0	1	1/40
1	2	2/40
2	3	3/40
3	3	3/40
4	3	3/40
5	2	2/40
6	10	10/40
7	10	10/40
8	3	3/40
9	2	2/40
10	1	1/40

La nota es una variable aleatoria cuyo dominio es el conjunto de los alumnos (considerado como el espacio muestral del experimento: elegir un alumno*) en el conjunto de los números enteros del 0 al 10. Agrupando los alumnos que tienen la misma nota se obtiene, con los datos de la tabla:*

$$E(X) = 0(1/40) + 1(2/40) + 2(3/40) + 3(3/40) + 4(3/40) + 5(2/40)+$$

$$+6(10/40) + 7(10/40) + 8(3/40) + 9(2/40) + 10(1/40) = 5.52.$$

$$V(X) = (0-5.52)^2(1/40) + (1-5.52)^2(2/40) + (2-5.52)^2(3/40)+$$

$$+(3-5.52)^2(3/40) + (4-5.52)^2(3/40) + (5-5.52)^2(2/40)+$$

$$+(6-5.52)^2(10/40) + (7-5.52)^2(10/40) + (8-5.52)^2(3/40)+$$

$$+(9-5.52)^2(2/40) + (10-5.52)^2(1/40) = 5.54.$$

$$\sigma(X) = 2.35.$$

Ejercicio 88 *Sea el experimento aleatorio consistente en tirar dos dados consecutivamente y X la variable aleatoria que asigna a cada resultado del experimento la suma de los resultados de ambos dados:*

i) Escribir el espacio muestral del experimento.
ii) Escribir la función $X : \Omega \to \mathbb{R}$.
iii) Calcular la esperanza, la varianza y la desviación típica de X.

4.6.5 Probabilidad condicionada

En esta sección vamos a analizar cómo se modifica la probabilidad de sucesos cuando tenemos una información adicional. Por ejemplo:

Ejemplo 4.6.18 *La probabilidad de que al extraer una carta al azar en una baraja española sea un caballo es* $4/40 = 1/10$*, ya que la baraja de 40 cartas contiene 4 caballos. Pero supongamos que recibimos la información adicional de que la carta extraída es una figura. Estamos entonces interesados en la probabilidad de extraer un caballo sabiendo que ha salido una figura. La baraja contiene 12 figuras, de las cuales 4 son caballos, por tanto la nueva probabilidad es* $4/12 = 1/3$*, lo cual indica que, en este caso, el hecho adicional de ser una figura hace aumentar la probabilidad del suceso.*

Definición 4.6.19 *Sean* S *y* S' *sucesos de un espacio muestral* Ω *de modo que* $p(S') > 0$*. Definimos la* **probabilidad condicionada de** S **por** S' *como*

$$p(S|S') = p(S \cap S')/p(S').$$

Y es un concepto que recoge la idea que señalábamos en el ejemplo porque si vale la regla de Laplace entonces

$$p(S \cap S')/p(S') = \frac{|S \cap S'|}{|\Omega|} : \frac{|S'|}{|\Omega|} = \frac{|S \cap S'|}{|S'|}.$$

Se puede interpretar como que el hecho adicional modifica el espacio muestral (lo reduce a S') y también el conjunto de los casos favorables (lo reduce a $S \cap S'$).

Ejemplo 4.6.20 *Cuál es la probabilidad de que al tirar una moneda sucesivamente 4 veces se obtengan exactamente 3 cruces, sabiendo que la primera tirada fue cruz.*

El espacio muestral es $\Omega = A \times A \times A \times A$ donde A es el conjunto $A = \{cara, cruz\}$, por tanto $|\Omega| = 16$. El suceso S *obtener exactamente tres cruces* tiene cardinal 4 (las 4 posibles tiradas en las que salió la cara). El suceso S' *la primera tirada fue cruz* tiene cardinal 8 (se puede interpretar como $\{cruz\} \times A \times A \times A$). Y el suceso $S \cap S'$ tiene cardinal 3. Entonces

$$p(S|S') = \frac{(3/16)}{(8/16)}.$$

Definición 4.6.21 *Dos sucesos S y S' se dicen independientes si verifican que:*

$$p(S \cap S') = p(S)p(S').$$

Ejemplo 4.6.22 *Extraemos una carta al azar en una baraja española, los sucesos S:* sea figura *y S':* sea de oros *son independientes porque $p(S) = 12/40$, $p(S') = 10/40$ y $p(S \cap S') = 3/40$. Y efectivamente*

$$\frac{3}{40} = \frac{12}{40}\frac{10}{40}.$$

Ejercicio 89 *Comprobar si los sucesos S:* una familia con tres hijos tenga hijos de los dos sexos *y S':* una familia con tres hijos tenga al menos un chico *son independientes.*

El estudio de la probabilidad discreta constituye un interesante campo de las matemáticas y existen profundos teoremas que no vamos a desarrollar (teorema de Bayes, de la probabilidad total). Además el espacio muestral puede no ser finito con lo que se debe cambiar la regla de Laplace o la forma de asignar probabilidades por instrumentos adecuados (que provienen del Análisis Matemático), ya que si el denominador en la regla de Laplace es no finito, el cociente no tiene sentido.

Finalizamos el capítulo con una presentación de los experimentos de Bernoulli, que son experimentos aleatorios cuyo espacio muestral se reduce a dos elementos (usualmente éxito y fracaso).

4.6.6 Experimentos de Bernoulli

Definición 4.6.23 *Un* **experimento de Bernoulli** *es un experimento cuyo espacio muestral tiene dos elementos $\Omega = \{e, f\}$ de manera que p es la probabilidad de e (se suele entender como probabilidad de éxito) y $1 - p$ la probabilidad de f (probabilidad de fracaso).*

Ejemplos 4.6.24 *1) Lanzar una moneda trucada que saca cara con probabilidad 1/3 y cruz con probabilidad 2/3.*

2) Lanzar un dado que tiene las caras coloreadas de dos colores, 5 de ellas en rojo, una de ellas en negro, el rojo tiene probabilidad 5/6 y el negro probabilidad 1/6.

Tomemos entonces un experimento de Bernoulli con probabilidades p de éxito y $1-p$ de fracaso, de tal manera que cada repetición del experimento sea independiente de la repetición anterior. Debemos señalar que, por ejemplo, si tomamos el experimento de extraer una bola de una urna con bolas blancas y negras, para que la repetición del experimento sea independiente de la repetición anterior, la bola extraida debe ser reintroducida en la urna.

Problema 1. Determinar la probabilidad de que al repetir el experimento n veces se tengan exactamente k éxitos. Si $k > n$ la probabilidad es 0 y si $k \leq n$ el resultado es:

$$\binom{n}{k} p^k (1-p)^{n-k}.$$

En efecto, por hipótesis de independencia, los sucesos *tener un éxito en la repetición a-ésima* (resp. un fracaso) y *tener un éxito en la repetición b-ésima* con $a \neq b$, son independientes, por tanto las probabilidades se van construyendo multiplicando. Así por ejemplo el suceso *dadas dos repeticiones obtener dos éxitos* tiene probabilidad p^2.

De esta manera el suceso *obtener exactamente k éxitos en las primeras k repeticiones de las n totales* tiene probabilidad

$$p^k (1-p)^{n-k}.$$

Pero los k éxitos pueden acontecer en cualesquiera k repeticiones de entre las n veces que se ha efectuado el experimento, por lo que esta probabilidad debe ir multiplicada por el número de subconjuntos de k elementos de un conjunto de n elementos, que son exactamente

$$\binom{n}{k}.$$

Por tanto se tiene la fórmula buscada.

Problema 2. Determinar la probabilidad de que al repetir el experimento n veces se tengan al menos k éxitos:

$$\sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}.$$

Esta fórmula sale de entender el suceso *al menos k éxitos* como la unión de los sucesos disjuntos *exactamente k éxitos*, *exactamente $k+1$ éxitos*, ..., *exactamente n éxitos*.

Y resulta obvio que se puede cambiar el papel de los éxitos y de los fracasos.

Ejercicio 90 *En las mismas condiciones de los problemas 1 y 2:*

1) Determinar la probabilidad de obtener como máximo k fracasos.
2) Determinar la probabilidad de no obtener ningún éxito.
3) Determinar la probabilidad de tener exactamente k fracasos.
4) Determinar la probabilidad de que todos sean éxitos.

Ejercicio 91 *Supongamos que la probabilidad de que nazca un niño es* 0.4 *mientras la de que nazca una niña es* 0.6. *Determinar:*

1) La probabilidad de que entre 100 nuevos nacimientos todos sean niñas.
2) Haya al menos 50 niños.
3) Haya como máximo 30 niños.

4.7 Ejercicios

Ejercicio 92. Determinar de cuántas maneras se puede elegir un cuadrado blanco y otro negro en un tablero de ajedrez de modo que los dos cuadrados no estén en la misma fila ni en la misma columna.

Ejercicio 93. Determinar en cuántos números de teléfono de 7 cifras, que pueden empezar por cero, alguna de las cifras está repetida.

Ejercicio 94. Una llave se fabrica haciendo incisiones de profundidad variable en ciertas posiciones fijadas de una llave lisa. Si las profundidades posibles son cuatro, determinar el número de incisiones que debe llevar la llave para que el número de llaves posibles sea mayor que 100.000.

Ejercicio 95. Dado el conjunto de los 54 alumnos de una clase, donde 30 son chicos y 24 son chicas, determinar:
i) El número de equipos de 4 alumnos que se pueden formar.
ii) El número de equipos de 4 alumnos que contengan al menos una chica.
iii) El número de equipos formados por dos chicas y dos chicos.

Ejercicio 96. Determinar cuántas palabras de 10 bytes:
i) Tienen exactamente tres ceros.

ii) Tienen al menos un cero.
iii) Tienen el mismo número de unos y ceros.

Ejercicio 97. Cuántas palabras de 10 letras se pueden construir reordenando las letras de la palabra *dodecaedro*.

Ejercicio 98. Tenemos 10 naipes de los cuáles son 3 ases de oros, 4 caballos de bastos y otras 3 cartas diferentes entre sí y a las anteriores. Si ponemos las 10 cartas en fila, determinar el número de distintas filas que se pueden formar.

Ejercicio 99. Determinar cuántas licencias de uso de un programa, formadas con tres números seguidos de tres letras no contienen la misma letra dos veces ni el mismo número tres veces.

Ejercicio 100. Se tienen entradas para 17 espectáculos distintos, determinar cuántos posibles regalos de tres entradas (posiblemente para el mismo espectáculo) se pueden hacer.

Ejercicio 101. Determinar qué suceso es más probable: *sacar un 8 al tirar dos dados* o *sacar un 8 al tirar tres dados*.

Ejercicio 102. Determinar la probabilidad de que una jugada de póker (5 cartas en una baraja francesa con 52 cartas, sin comodines):
i) Contenga al menos un as.
ii) Obtenga un póker (cuatro cartas iguales).

Ejercicio 103. En un experimento de Bernoulli con probabilidad 0.4 de éxito y 0.6 de fracaso, determinar cuántos lanzamientos hay que realizar para que la probabilidad de obtener al menos un éxito sea mayor o igual que 0.8.

Ejercicio 104. Sea la variable aleatoria X cuyo dominio es el espacio muestral Ω de lanzar un dado 10 veces consecutivas, asignando a cada elemento de Ω el número de seises que han salido. Calcular la esperanza matemática de X, que es el número de seises esperado al lanzar 10 veces el dado.

Ejercicio 105. Cuál es la probabilidad de que escogido un entero al azar menor o igual que 100 sea primo.

Ejercicio 106. Si S y S' son dos sucesos con $p(S) = 0.8$ y $p(S') = 0.6$, demuestra que $p(S \cap S') \geq 0.4$. En general para cada par de sucesos se verifica $p(S \cap S') \geq p(S) + p(S') - 1$.

Ejercicio 107. Demostrar que para dos variables aleatorias X e Y definidas en el mismo espacio muestral la esperanza matemática verifica:

$$E(X + Y) = E(X) + E(Y),$$

siendo $X + Y$ la función suma de las funciones X e Y.

4.8 Ejercicios resueltos

Ejercicio 77. Propiedad 1: Si $|A| = n$ entonces existe una aplicación biyectiva $f : A \to \mathbb{N}_n$. Si $|B| = m$ entonces existe una aplicación biyectiva $g : B \to \mathbb{N}_m$. La aplicación biyectiva que estamos buscando es $F : A \cup B \to \mathbb{N}_{n+m}$ definida como $F(a) = f(a)$ si $a \in A$ y $F(b) = g(b) + n$ si $b \in B$.

Es sobreyectiva ya que si $s \in \mathbb{N}_n$ entonces existe $a \in A$ de modo que $s = f(a) = F(a)$ y si $n < s \leq n + m$ entonces $s - n \in \mathbb{N}_m$ de modo que existe $b \in B$ tal que $g(b) = s - n$ por lo que $F(b) = s$.

Es inyectiva ya si $a \neq b$ entonces $F(a) \neq F(b)$. En efecto, tanto f como g son inyectivas por lo que el resultado es cierto si $a, b \in A$ o si $a, b \in B$. En el caso $a \in A$ y $b \in B$ se verifica que $F(a) \leq n$ y $F(B) > n$ por lo que son distintos necesariamente.

Propiedad 2: se aplica la 1 a B y $\overline{B}$.

Propiedad 3: Como $A \subset B$ entonces $B = A \cup (B - A)$ y se concluye por 1.

Propiedad 4: Basta usar las propiedades anteriores escribiendo $A \cup B = A \cup (B - (A \cap B))$.

Propiedad 5: Por inducción sobre el cardinal de B.

Si $|B| = 0$ entonces B es vacío de modo que el producto cartesiano también lo es.

Si $|B| = n + 1$ entonces $B = B' \cup B''$ con $|B'| = n$ y $|B''| = 1$. De este modo como $A \times B = (A \times B') \cup (A \times B'')$ y $(A \times B') \cap (A \times B'') = \emptyset$ se concluye por la propiedad 1 y la hipótesis de inducción.

Propiedad 3 implica propiedad 6: Si hubiera una función inyectiva $f : A \to B$ entonces la imagen de f es un subconjunto de B de cardinal estrictamente mayor que $|B|$ lo que es una contradicción.

Propiedad 6 implica propiedad 3: Si $A \subset B$ entonces hay una aplicación inyectiva de A en B (la inclusión) de modo que $|A| < |B|$.

Ejercicio 78.

i) Como no se indica nada, las letras se pueden repetir: $VR_{29,5} = 29^5$.

ii) $V_{29,5} = 29 \times 28 \times 27 \times 26 \times 25$.

iii) Las palabras que no contienen a la letra b son 28^5 por lo que la contienen $29^5 - 28^5$.

iv) Razonando como en iii) son $V_{29,5} - V_{28,5}$.

v) Si empiezan por vocal y se pueden repetir son 5×29^4. Si no se pueden repetir son $5 \times V_{28,4}$.

vi) Si se pueden repetir son 5×29^3. Si no se pueden repetir son $5 \times V_{27,3}$.

vii) Si se pueden repetir son $5^3 \times 29^2$. Si no se pueden repetir son $5 \times 4 \times 3 \times 26 \times 25$.

Ejercicio 79.

i) La primera cifra no puede ser 0 y las cifras se pueden repetir: 9×10^3.

ii) Son pares si su última cifra es 0, 2, 4, 6 u 8 entonces: $9 \times 10^2 \times 5$ (exactamente la mitad de i)).

iii) Son impares la otra mitad.

iv) Son múltiplos de 5 si su última cifra es un 0 o un 5: $9 \times 10^2 \times 2$.

v) Si no contienen al 2 son 8×9^3, por tanto los que contienen al 2 son $9 \times 10^3 - 8 \times 9^3$.

vi) S no contienen ni a la cifra 2 ni a la cifra 3 son: 7×8^3 entonces los que contienen al 2 o al 3 o a ambos son: $9 \times 10^3 - 7 \times 8^3$.

Ejercicio 80. Se pueden sentar de 6! maneras y si sólo importa la posición relativa en una mesa circular entonces el resultado hay que dividirlo por 6, esto es, $6!/6 = 5!$. Dividimos por 6 que son las 6 posiciones relativas iguales resultado de moverse todos una silla (dos sillas, ..., 6 sillas) hacia la derecha.

Ejercicio 81.

i) $24^3 \times 10^4$.

ii) $PR_3^2 \times PR_4^2 = 3 \times 4!/2$.

iii) $24^2 \times 10^3$.

iv) Si son capicúas la primera cifra es igual que la cuarta y la segunda es igual que la tercera. Por tanto hay 10^2 dígitos. En total: $24^3 \times 10^2$.

v) 24×10^4.

vi) $PR_3^2 \times PR_4^{2,2} = 3 \times 3!$.

Ejercicio 82.

i) 5!.

ii) 5!/4.

iii) 5!/12.

iv) Los boletos con todas sus cifras iguales son los menos probables de ser premiados y los que tienen todas sus cifras distintas los que más.

Ejercicio 83. Como $V_{m,n} = m(m-1)...(m-(n-1))$ entonces $m! = V_{m,n}(m-n)!$ de modo que:

$$\binom{m}{n} = \frac{V_{m,n}}{n!} = \frac{m!}{n!(m-n)!}.$$

Ejercicio 84.

1) Un triángulo es una terna no ordenada de puntos: $\binom{312}{3}$.

2) Las ternas de puntos que no dan un triángulo se forman eligiendo 3 de los 5 puntos alineados, por tanto:

$$\binom{312}{3} - \binom{5}{3}.$$

Ejercicio 85.

1) $CR_{5,6} = \binom{10}{6}$.

2) Si quitamos el 5 de cada bombo tenemos $CR_{4,6}$ posibles resultados que no contienen al 5. Los que llo contienen son por tanto:

$$CR_{5,6} - CR_{4,6}.$$

Ejercicio 86. Acertar 13 en las quinielas tiene probabilidad $28/3^{14}$. Acertar 5 en la primitiva: $(6 \times 43)/\binom{49}{6}$.

Ejercicio 87. $\Omega = \{as, 2, 3, 4, caballo, rey\}$.

$p(as) = 3/15$

$p(2) = 3/15$

$p(3) = 1/15$
$p(4) = 4/15$
$p(caballo) = 2/15$
$p(rey) = 2/15$
$p(figura) = p(caballo) + p(rey) = 4/15.$

Ejercicio 88. i) Al tirar dos dados consecutivamente $\Omega = \mathbb{N}_6 \times \mathbb{N}_6$.

ii) $X : \Omega \to \mathbb{R}$ asigna $a + b$ a la tirada (a, b).

iii) Escribiendo las fórmulas y haciendo las respectivas cuentas se obtiene: $E(X) = 7$, $V(X) = 6.16$, $\sigma(x) = \sqrt{6.16} = 2.48$.

Ejercicio 89. Se tiene que $|\Omega| = 8$.

El suceso S:*tener hijos de los dos sexos* tienen cardinal 6, ya que hay que descontar los casos en que hay tres varones o tres mujeres.

El suceso S':*tener al menos un hijo varón* tienen cardinal 7, ya que hay que descontar sólo el caso en que son tres mujeres.

Entonces $p(S) = 6/8$ y $p(S') = 7/8$.

Por otro lado $|S \cap S'| = 6$ porque $S \subset S'$ de modo que los sucesos no son independientes.

Ejercicio 90.
1) $\Sigma_{i=0}^{k} \binom{n}{i} (1-p)^i p^{n-i}$.
2) $(1-p)^n$.
3) $\binom{n}{k} (1-p)^k p^{n-k}$.
4) p^n.

Ejercicio 91.
1) $(0.6)^{100}$.
2) $\Sigma_{i=50}^{100} \binom{100}{i} (0.4)^i (0.6)^{100-i}$.
3) $\Sigma_{i=0}^{30} \binom{100}{i} (0.4)^i (0.6)^{100-i}$.

Ejercicio 92. Tomamos un cuadrado blanco de los $64/2$ cuadrados blancos que hay. Escogemos un cuadrado negro de los $64/2 - 8$ posibles (los negros que no están en la misma fila o en la misma columna). El resultado es el producto: 768.

Ejercicio 93. Buscamos el complementario del conjunto de todos los números de 7 cifras tales que ninguna de ellas está repetida:

$$10^7 - V_{10,7}.$$

Ejercicio 94. Las posibilidades con n incisiones son: 4^n. Como $\{4^n : n \in \mathbb{N}\}$ es una sucesión creciente buscamos el primer n que verifique que $4^n > 10^5$ y es $n = 9$.

Ejercicio 95.

i) Es el número de subconjuntos de 4 elementos en un conjunto de 54 alumnos: $\binom{54}{4}$.

ii) Será el número total de equipos de 4 alumnos menos el número de equipos constituidos sólo por chicos:

$$\binom{54}{4} - \binom{30}{4}.$$

iii) El número de subconjuntos de dos chicas por el número de subconjuntos de dos chicos:

$$\binom{30}{2} \times \binom{24}{2}.$$

Ejercicio 96.

i) $PR_{10}^{3,7} = \binom{10}{3}$.

ii) Todos menos el formado por todos unos: $2^{10} - 1$.

iii) $PR_{10}^{5,5} = \binom{10}{5}$.

Ejercicio 97. La d se repite 3 veces, la o se repite dos veces y la e otras 2. Total: $PR_{10}^{3,2,2}$.

Ejercicio 98. Los ases se repiten 3 veces y los caballos 4. Total: $PR_{10}^{3,4}$.

Ejercicio 99. El número de licencias buscado será el total de posibles menos aquéllas con la misma letra (exactamente) dos veces y el mismo número 3 veces. Total: $10^3 \times 29^3 - 3 \times 29 \times 28 \times 10$. Siendo 29×28 el número de parejas de letras posibles, 3 el número de modos de ordenar las tres letras (dos iguales y una distinta) y 10 los posibles tríos de números.

Ejercicio 100. $CR_{17,3} = \binom{19}{3}$.

Ejercicio 101. Si tiramos dos dados: Hay 6^2 posibles lanzamientos. De ellos dan como suma 8 los siguientes:

$$(2,6), (6,2), (3,5), (5,3), (4,4).$$

La probabilidad de obtener un 8 es $5/36 = 0.13889$.

Si tiramos tres dados: Hay 6^3 posibles lanzamientos. De ellos dan como suma 8 los siguientes: $PR_7^{2,5}$ (interpretando cada tirada como una palabra de 7 carácteres que son 5 unos y 2 ceros, por ejemplo la tirada $2, 2, 4$ es 1010111. Los ceros separan las tiradas y ponemos tantos unos como el valor de la tirada menos una unidad). Entonces la probabilidad es $21/216 = 0.0722$ menor que la anterior.

Ejercicio 102.

i) La probabilidad será 1 menos la probabilidad de que no contenga ningún as:

$$1 - \binom{48}{5} / \binom{52}{5}.$$

ii) Hay 13 posibles pókeres (de ases, de doses,...) y por cada uno puede haber una quinta carta que es arbitraria, de entre las 48 cartas restantes (47 si no queremos que sea un repóker). Por tanto la probabilidad de póker es:

$$48 \times 13 / \binom{52}{5}.$$

Ejercicio 103. El suceso *obtener al menos un éxito* es complementario al suceso *no tener ningún éxito*: $1 - 0.6^n$. Buscamos n tal que $1 - 0.6^n \geq 0.8$ o equivalentemente $0.6^n \leq 0.2$. Como la sucesión $\{0.6^n : n \in \mathbb{N}\}$ es decreciente, evaluando, se tiene que el primero que verifica la desigualdad es $n = 4$.

Ejercicio 104. El suceso *obtener exactamente k seises al tirar el dado 10 veces* tiene la siguiente probabilidad:

$$\binom{10}{k} (1/6)^k (5/6)^{10-k}.$$

Por tanto el espacio muestral es $\Omega = \{0\} \cup \mathbb{N}_{10}$ y para cada $k \in \mathbb{N}_{10}$ se tiene la probabilidad mencionada, de modo que la esperanza matemática es:

$$\Sigma_{k=0}^{10} k \binom{10}{k} (1/6)^k (5/6)^{10-k}.$$

Ejercicio 105. Hay 26 primos menores que 100, por tanto $26/100$.

Ejercicio 106. Como $p(S \cup S') = p(S) + p(S') - p(S \cap S')$ entonces:

$$p(S \cap S') = p(S) + P(S') - p(S \cup S').$$

Se concluye la fórmula general del hecho de que $p(S \cup S') \leq 1$. Esta fórmula se aplica al ejemplo $p(S) = 0.8$ y $p(S') = 0.6$.

Ejercicio 107. Como la variable $X + Y$ se define como $(X + Y)(a) = X(a) + Y(a)$ para cada $a \in \Omega$, el ejercicio es consecuencia de la propiedad distributiva.

Capítulo 5

Grafos

La teoría de grafos tiene su origen en un artículo publicado por Euler en 1736 en el que se daba solución al problema de los siete puentes de la ciudad de Königsberg, puentes que conectaban dos islas con las márgenes del río según el esquema de la figura que se adjunta (Figura 5.1).

El problema consistía en realizar un paseo que atravesase cada uno de los siete puentes una única vez (los puentes en el gráfico son los que unen las regiones A,B,C y D).

Euler demostró que esto no era posible sin necesidad de comprobar todos los posibles paseos.

Desde entonces los resultados y aplicaciones de la teoría de grafos han ido aumentando y actualmente se utiliza para resolver problemas en ramas de la ciencia muy diferentes. Los grafos sirven, por ejemplo, para construir modelos

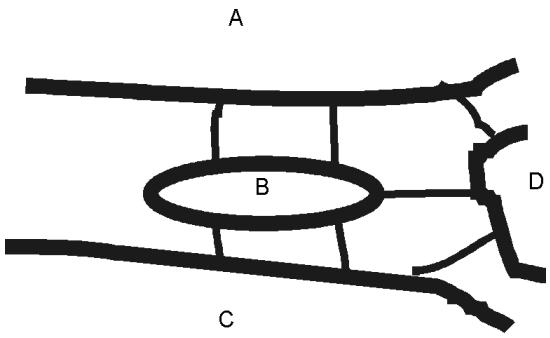


Figura 5.1: Puentes de Könisberg

de redes de ordenadores y determinar si dos ordenadores están conectados entre sí; para discriminar si un circuito puede ser implementado sobre un tablero plano; para distinguir compuestos químicos con la misma fórmula molecular o para encontrar el camino más corto entre dos ciudades en una red de transporte.

Se pretende que el alumno al finalizar el capítulo:

- Entienda los conceptos y definiciones básicas de la teoría de grafos.
- Utilice las distintas representaciones de grafos.
- Utilice diferentes métodos para reconocer si dos grafos son o no isomorfos.
- Conozca los conceptos de grafo conexo, euleriano y hamiltoniano, las técnicas para reconocerlos, sus propiedades y algunos resultados relacionados con ellos.
- Aplique la definición más adecuada para verificar si un grafo es un árbol.
- Conocer algunas aplicaciones de la teoría de grafos a la fundamentación de la informática, en concreto, complejidad de problemas.

5.1 Grafos, digrafos y multigrafos

Los grafos son estructuras combinatorias que constan de vértices y aristas que conectan estos vértices. En esta sección introduciremos algunos conceptos de la teoría de grafos.

Definición 5.1.1 *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,*

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

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

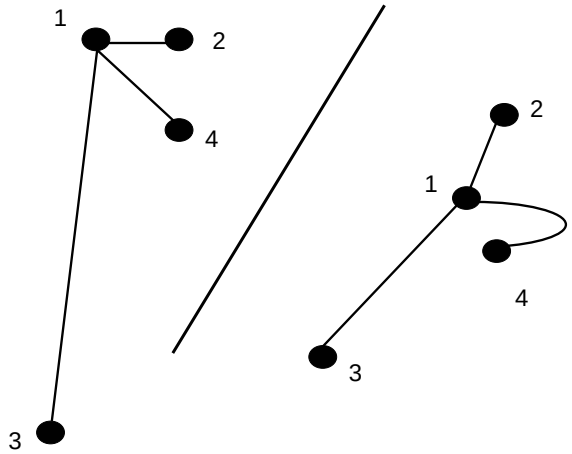


Figura 5.2: Dos representaciones del mismo grafo

Ejemplo 5.1.2 *Un grafo simple es, por ejemplo, el grafo* $G = (V, E)$ *donde* $V = \{1, 2, 3, 4\}$ *y* $E = \{\{1, 2\}, \{1, 4\}, \{1, 3\}\}$.

Observación 5.1.3 *Podemos representar los vértices como puntos del plano y las aristas* $\{u, v\}$ *como curvas que unen* u *y* v. *Conviene observar que la representación no es necesariamente única. Por ejemplo la figura 5.2 recoge dos representaciones del grafo anterior.*

Definición 5.1.4 *Un* **multigrafo** *es un par* (V, E) *formado por un conjunto finito de vértices* V *y una familia finita* E *de aristas no orientadas*

$$E = \{e_i\}_{i \in I}$$

donde I *es un conjunto finito y* $\forall i \in I$ *se verifica que* $e_i = \{u_i, v_i\}$ *con* $u_i, v_i \in V$, *posiblemente iguales.*

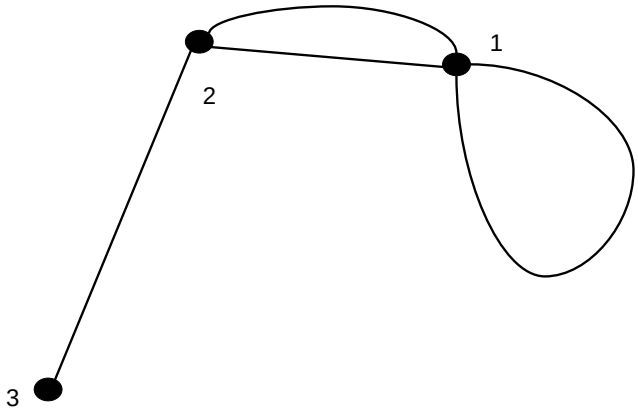


Figura 5.3: Un multigrafo

Observación 5.1.5 *Para obviar el conjunto de índices de una familia finita, la denotaremos en lo sucesivo como un n-tupla. Por ejemplo si* $|I| = n$ *entonces* $E = \{e_i\}_{i \in I}$ *se denotará como* $(e_1, ..., e_n)$.

Ejemplo 5.1.6 *Un ejemplo de multigrafo es*

$$(\{1,2,3\}, (\{1,1\}, \{1,2\}, \{1,2\}, \{2,3\}))$$

y se puede representar como la figura 5.3.

Observación 5.1.7 *Nótese que, en un multigrafo, dos aristas distintas pueden conectar los mismos vértices, esto es, que* E *es una familia y no un conjunto, pudiendo tener elementos repetidos. Nótese también que estamos permitiendo aristas del tipo* $\{u, u\}$ *denominadas* **lazos** *o* **bucles**.

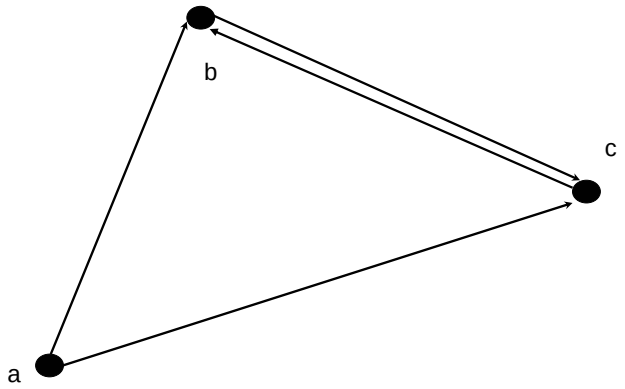


Figura 5.4:

Definición 5.1.8 *Un* **digrafo** *es un par* (V, E) *donde* V *es un conjunto finito y* $E \subset (V \times V) - \Delta$, *siendo* $\Delta = \{(x, x) : x \in V\}$. *A los elementos de* V *se les denomina* **vértices** *y a los de* E **aristas (dirigidas u orientadas)**.

Ejemplo 5.1.9 *({a,b,c},{(a,b),(a,c),(b,c),(c,b)}) es un digrafo que podemos representar utilizando puntos para representar los vértices y flechas para representar las aristas, según la figura 5.4. Observar que la arista* (b, c) *es distinta de la arista* (c, b).

Definición 5.1.10 *Un* **multidigrafo** *es un par* (V, E) *formado por un conjunto finito de vértices* V *y una familia finita* E *de aristas orientadas*

$$E = \{e_i\}_{i \in I}$$

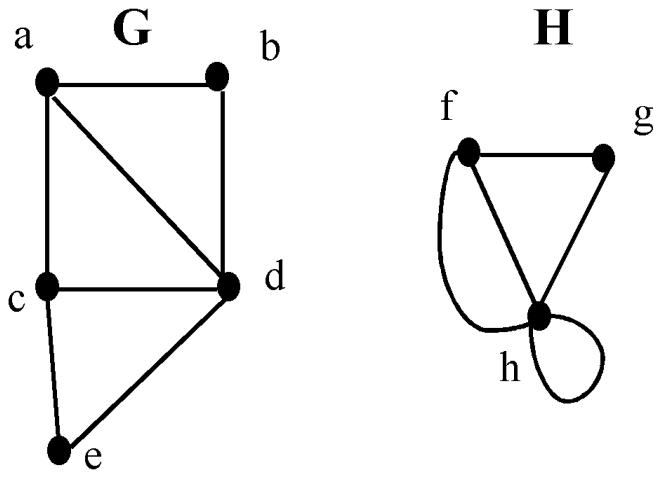


Figura 5.5:

donde I es un conjunto finito y $\forall i \in I$ se verifica que $e_i \in V \times V$.

En lo sucesivo, el término **grafo** se empleará en sentido general, para describir grafos con aristas dirigidas o no dirigidas, con o sin lazos y con o sin aristas múltiples. Por otro lado, el término **grafo no dirigido** se entenderá como sinónimo de multigrafo, admitiendo, por tanto, la existencia de aristas múltiples y lazos. De igual manera, el término **grafo dirigido** se entenderá como multidigrafo.

Definición 5.1.11 *Se dice que dos vértices u y v de un grafo no dirigido $G = (V, E)$ son* **adyacentes** *si $\{u, v\} \in E$. En ese caso se dice que la arista $e = \{u, v\}$* **conecta** *los vértices u y v, que es* **incidente** *con los vértices u y v, y que los vértices u y v son los* **extremos** *de la arista e.*

Definición 5.1.12 *El* **grado** *de un vértice en un grafo no dirigido es el número de aristas incidentes con él, imponiendo por conveniencia que un lazo en un vértice contribuye dos veces al grado de ese vértice. Denotaremos el grado de un vértice u por $gr(u)$.*

Ejemplo 5.1.13 *Considérense los grafos G y H de la figura 5.5. En el grafo G se verifica que $gr(a) = 3 = gr(c)$, $gr(b) = 2 = gr(e)$ y $gr(d) = 4$. En el grafo H se verifica que $gr(f) = 3$, $gr(g) = 2$ y $gr(h) = 5$.*

Si un vértice tiene grado cero se dice que es un **vértice aislado**.

Los dos siguientes teoremas son consideraciones que relacionan entre sí el número de aristas y los grados de los vértices.

Teorema 5.1.14 *Sea $G = (V, E)$ un grafo no dirigido. Se verifica que*

$$\sum_{v \in V} gr(v) = 2\,|E|\,.$$

Demostración. La demostración es consecuencia de que cada arista contribuye dos veces a la suma de los grados de los vértices ya que una arista es incidente con exactamente dos vértices (que para los lazos son iguales).

Corolario 5.1.15 *Cualquier grafo no dirigido tiene un número par de vértices de grado impar.*

Demostración. Sean V_1 y V_2 los conjuntos de vértices de grado par e impar respectivamente del grafo $G = (V, E)$. En ese caso

$$2\,|E| = \sum_{v \in V} gr(v) = \sum_{v \in V_1} gr(v) + \sum_{v \in V_2} gr(v).$$

o equivalentemente

$$2\,|E| - \sum_{v \in V_1} gr(v) = \sum_{v \in V_2} gr(v).$$

Puesto que para cada $v \in V_1$ se tiene que $gr(v)$ es un número par y $2|E|$ es par entonces necesariamente $\sum_{v \in V_2} gr(v)$ es un número par.

Definición 5.1.16 *Si $G = (V, E)$ es un grafo dirigido, y $(u, v) \in E$, se dice que u es el* **vértice inicial** *de la arista (u, v) y que v es el* **vértice final** *de dicha arista. Asimismo, dado $u \in V$, se denomina* **grado de entrada** *de u al número de aristas que tienen a u como vértice final. Al grado de entrada de u se le denota por $gr^+(u)$. Del mismo modo, se denomina* **grado de salida** *de u al número de aristas que tienen a u como vértice inicial. Al grado de salida de u se le denota por $gr^-(u)$.*

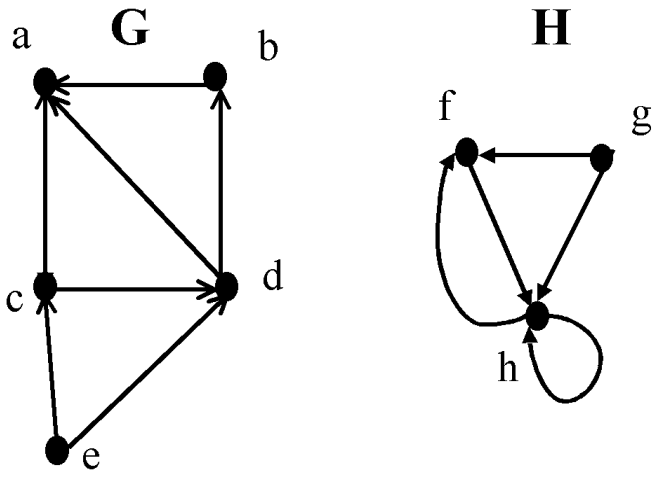


Figura 5.6:

Observación 5.1.17 *Un lazo en un vértice u suma uno al grado de entrada y uno al grado de salida.*

Ejemplo 5.1.18 *En el grafo G de la figura 5.6, se verifica que* $gr^+(a) = 3$, $gr^-(a) = 0$, $gr^+(c) = 1$, $gr^-(c) = 2$, $gr^-(d) = gr^+(d) = 2$, $gr^-(b) = gr^+(b) = 1$, $gr^+(e) = 0, gr^-(e) = 2$. *Por otra parte, en el grafo H tenemos que* $gr^-(f) = 1$, $gr^+(f) = 2$, $gr^+(g) = 2$, $gr^-(g) = 0, gr^-(h) = 2, gr^+(h) = 3$.

Teorema 5.1.19 *En cualquier grafo con aristas dirigidas* $G = (V, E)$ *se verifica que*

$$|E| = \sum_{v \in V} gr^+(v) = \sum_{v \in V} gr^-(v).$$

Demostración. Es consecuencia del hecho de que cualquier arista dirigida tiene un vértice inicial y un vértice final.

5.2 Isomorfismo de grafos

Supongamos que tomamos los grafos simples

$$G = (\{1, 2\}, \{\{1, 2\}\})$$

y

$$G' = (\{a, b\}, \{\{a, b\}\}).$$

Según la definición de grafo G y G' son grafos distintos pues sus conjunto de vértices (y aristas) son distintos. Pero parece que estos dos grafos son muy similares, pues su naturaleza combinatoria es la misma, a saber, están formados por dos vértices distintos que son adyacentes. Ambos admiten una representación igual por dos puntos en el plano unidos por un segmento. En este sentido diremos que los grafos G y G' son *isomorfos*. Hay un cambio de nombre de los vértices de G, por ejemplo llamar a al vértice 1 y b al vértice 2, de manera que el grafo G se convierte en el grafo G' tras este cambio de nombres. A este cambio de nombres lo denominaremos *isomorfismo de grafos* y se define con precisión más abajo.

En Química, por ejemplo, los grafos se emplean para representar y dar modelos de los compuestos químicos. Dos compuestos diferentes pueden tener la misma composición pero diferir en su estructura. En ese caso dichos compuestos se modelizarán por grafos que no se pueden expresar de la misma forma, en el sentido del párrafo anterior. Los grafos que representan compuestos conocidos pueden utilizarse, por ejemplo, para determinar si un supuestamente nuevo compuesto químico ha sido estudiado antes.

En estos términos, diremos que dos grafos simples que tienen la misma estructura *son isomorfos*, según la siguiente definición:

Definición 5.2.1 *Se dice que dos grafos simples $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ son* **isomorfos** *si existe una biyección $f : V_1 \to V_2$ tal que $\forall u, v \in V_1$*

$$\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2.$$

De la función f que satisface dicha condición se dice que es un **isomorfismo de grafos** *entre los grafos G_1 y G_2.*

En otras palabras, dos grafos simples son isomorfos si existe una función biyectiva entre los dos conjuntos de vértices que preserva las adyacencias. Naturalmente esta definición se puede extender a multigrafos y multidigrafos teniendo en cuenta el número de aristas entre cada par de vértices y, en su caso, la orientación de las aristas.

Ejemplo 5.2.2 *Los dos grafos de la figura 5.7 son isomorfos. Para verlo basta comprobar que la función $f : \{a, b, c, d\} \to \{u, v, w, p\}$, tal que $f(a) = u$, $f(b) = v$, $f(c) = w$, $f(d) = p$ es un isomorfismo de grafos.*

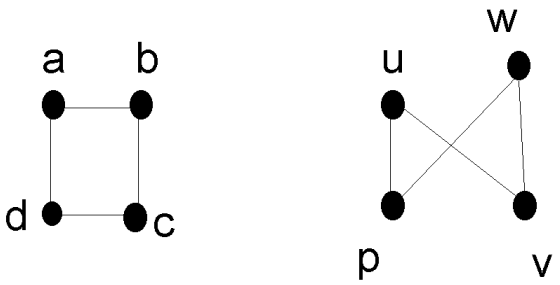


Figura 5.7:

Observación 5.2.3 *Obsérvese que en el ejemplo anterior el cruce de las dos aristas del grafo de la derecha* **no** *es un vértice.*

A menudo es difícil determinar si dos grafos simples son isomorfos. De hecho, como vimos en el capítulo anterior, hay $n!$ aplicaciones biyectivas entre los conjuntos de vértices de dos grafos con n vértices, por lo que comprobar una por una si dichas biyecciones preservan la adyacencias no es un buen método, sobre todo si n es un número grande.

Debemos entonces encontrar criterios para determinar si dos grafos simples son isomorfos o no lo son que no precisen una comprobación exhaustiva.

Estos criterios se apoyan en el hecho de que hay ciertas propiedades, denominadas **invariantes** por isomorfismo, que, si un grafo las verifica, cualquier otro grafo isomorfo a él las debe también verificar. Por ejemplo:

(i) dos grafos isomorfos deben tener el mismo número de vértices y el mismo número de aristas.

(ii) Si $f : V_1 \to V_2$ establece un isomorfismo entre los grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, entonces para cada $u \in V_1$ se tiene que $gr(u) = gr(f(u))$.

Este tipo de resultados sirve para comprobar con cierta facilidad que algunos grafos no son isomorfos.

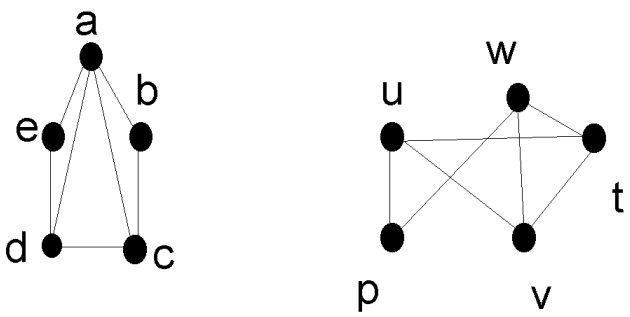


Figura 5.8:

Ejemplo 5.2.4 *Los dos grafos de la figura 5.8 no son isomorfos pues, aunque ambos tienen el mismo número de vértices y de aristas, en el primero* $gr(a) = 4$, *mientras que en el segundo no hay ningún vértice cuyo grado sea 4.*

Ejercicio 108 *Estudiar si los grafos de la figura 5.9 son isomorfos. Idem para la figura 5.10*

De ahora en adelante **identificaremos los grafos isomorfos**. Esto quiere decir que no distinguiremos entre el grafo G y los grafos isomorfos a él. De hecho, puesto que en la definición de grafo, el conjunto de vértices y el de aristas son conjuntos finitos, estamos identificando todos los grafos de la misma clase de equivalencia módulo isomorfismo de grafos.

5.3 Algunos Grafos

Vamos a establecer una nomenclatura especial para algunos grafos particularmente interesantes. Recordamos el último párrafo de la sección anterior: identificamos los grafos isomorfos.

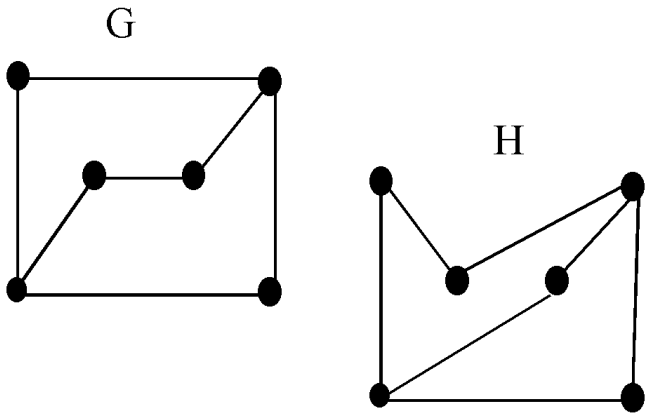


Figura 5.9:

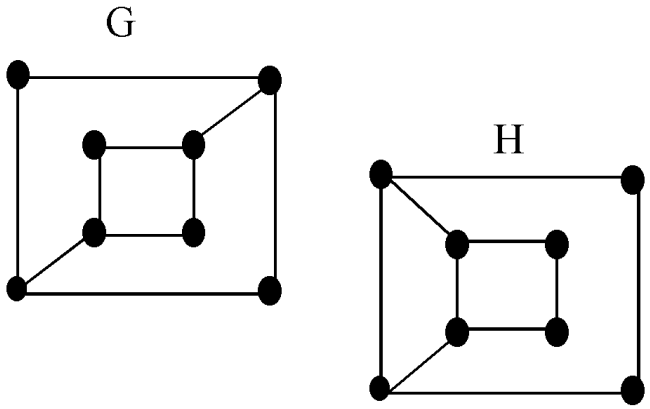


Figura 5.10:

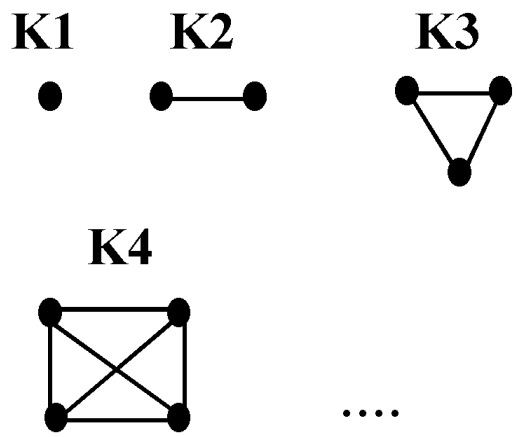


Figura 5.11:

Definición 5.3.1 *Se denomina* **grafo completo** *de n vértices al grafo simple $K_n = (\mathbb{N}_n, \{\{i,j\} : 1 \leq i < j \leq n\})$. Esto significa que cada par de vértices distintos son adyacentes (ver figura 5.11).*

Definición 5.3.2 *El* **ciclo** *de n vértices C_n ($n \geq 3$) es el grafo simple que tiene como conjunto de vértices $V = \mathbb{N}_n$, y como conjunto de aristas $E = \{\{1,2\},\{2,3\},\ldots,\{n-1,n\}\},\{n,1\}\}$. (Ver figura 5.12.)*

Definición 5.3.3 *La* **rueda** *W_n se obtiene añadiendo un vértice adicional al ciclo C_n y las n aristas que conectan dicho vértice adicional con todos los de C_n. Podemos escribirlo así:*

$$M_n = (\mathbb{N}_n \cup \{o\}, \{\{1,2\},\{2,3\},\ldots,\{n-1,n\},\{1,o\},\ldots,\{n,o\}\}).$$

Ejercicio 109 *Dibujar las ruedas W_3, W_4, W_5 y W_6. (Obsérvese que la rueda W_n tiene $n+1$ vértices).*

Definición 5.3.4 *El* **n-cubo** *$Q_n = (V,E)$ sirve para representar las secuencias de n elementos del conjunto $A = \{0,1\}$ (es decir, las secuencias*

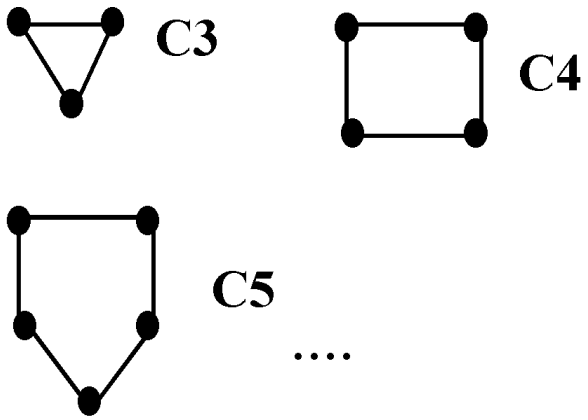


Figura 5.12:

de n bits). Así $V = A^n$ de modo que Q_n tiene 2^n vértices (uno por cada secuencia de n bits). Las aristas se definen según la siguiente condición: $\{(a_1, \ldots, a_n), (b_1, \ldots, b_n)\} \in E$ si y solamente si $\Sigma_{i=1}^{n} |a_i - b_i| = 1$. Esto es, hay una arista entre cada par de vértices que satisfagan la condición de que su secuencia de bits asociada difiere en un único bit (Ver figura 5.13).

Ejercicio 110 *Dibujar el cubo Q_3.*

Algunas veces un grafo tiene la propiedad de que su conjunto de vértices se puede dividir en dos subconjuntos disjuntos de manera que todas las aristas satisfacen la condición de conectar un vértice de uno de los dos subconjuntos con un vértice del otro. Por ejemplo, consideremos el grafo que representa los matrimonios de una determinada ciudad. Los vértices son las personas casadas de esa ciudad, y se establece una arista entre los dos miembros de cada matrimonio. El grafo así obtenido tiene la propiedad descrita.

Definición 5.3.5 *Se dice que un grafo simple $G = (V, E)$ es* **bipartido** *si su conjunto de vértices V se puede expresar como la unión de dos subconjuntos no vacíos disjuntos V_1 y V_2 de manera que cada arista del grafo conecta un vértice de V_1 con un vértice de V_2. Esto es, no existe ninguna arista*

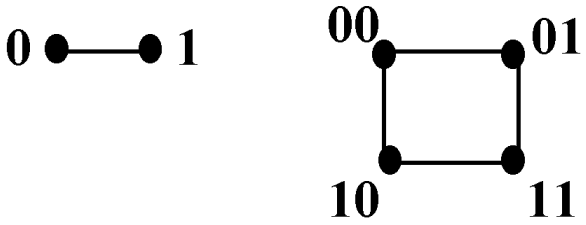


Figura 5.13:

entre dos vértices de V_1 ni entre dos vértices de V_2: si $\{u, v\} \in E$ entonces $|\{u, v\} \cap V_i| = 1$, $i = 1, 2$. (Ver un ejemplo en la figura 5.14.)

Ejercicio 111 *Comprobar que K_3 no es bipartido, y que C_6 es bipartido.*

Definición 5.3.6 *Sea $V_1 = \mathbb{N}_m$ y $V_2 = \{1', ..., m'\}$. El grafo* **bipartido completo** *$K_{m,n} = (V, E)$ se define como $V = V_1 \cup V_2$ y $E = \{\{n, n'\} : n \in V_1,\ n' \in V_2\}$. Esto es, V se puede expresar como la unión de dos subconjuntos disjuntos V_1 de m vértices y V_2 de n vértices, de manera que cada vértice de V_1 está conectado con todos los vértices de V_2 y ninguna arista conecta un par de vértices de V_1 ni de V_2. (Ver figura 5.15.)*

Ejercicio 112 *Dibujar los grafos bipartidos completos $K_{2,5}$ y $K_{3,4}$. Determinar el número de aristas del grafo $K_{m,n}$.*

Ejercicio 113 *Verificar si alguno de los grafos de la siguiente lista es isomorfo a alguno de los restantes: K_4, W_3 y $K_{1,3}$.*

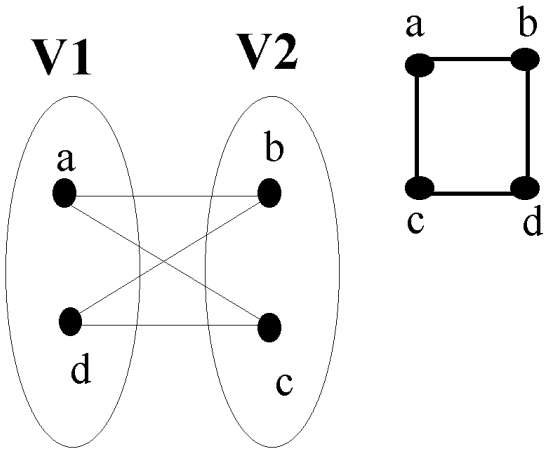


Figura 5.14:

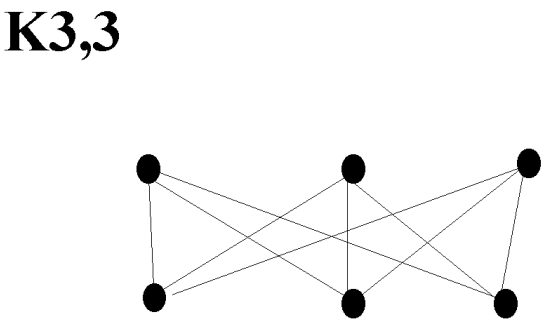


Figura 5.15:

Ejemplo 5.3.7 *Una red de área local permite conectar ordenadores entre sí y con diferentes periféricos, como impresoras, scanners, etc. Algunas de estas redes están diseñadas considerando una topología tipo estrella, caracterizada por el hecho de que todos los dispositivos están conectados a un dispositivo de control central. En una red de este tipo, los mensajes que se envían de un dispositivo a otro pasan siempre por el dispositivo de control central. Una red de área local de este tipo se puede representar utilizando un grafo completo $K_{1,n}$. Otras redes de área local están basadas en una topología tipo anillo, en la que cada dispositivo está conectado únicamente con otros dos y los mensajes se envían de un dispositivo a otro alrededor del ciclo, hasta que el mensaje en cuestión llega a su receptor. Las redes de área local de este tipo se modelizan utilizando los n-ciclos C_n. Otras redes de área local son de un tipo híbrido de las dos topologías anteriores. Tal sería el caso de las redes que modelizaríamos utilizando el grafo W_n.*

Analicemos algunos datos sobre cada uno de estos grafos. Usamos la notación $G = (V, E)$. Los cómputos se basan en las definiciones de los distintos grafos y en el teorema 5.1.14. La veracidad de las fórmulas se demuestra por induccion:

Grafo	$\|V\|$	$\|E\|$	$gr(v)$
K_n	n	$\frac{n(n-1)}{2}$	$n-1$
C_n	n	n	2
W_n	$n+1$	$2n$	$gr(i) = 3,\ (1 \leq i \leq n)$ $gr(o) = n$
Q_n	2^n	$n2^{n-1}$	n
$K_{n,m}$	$m+n$	$m\dot{n}$	$gr(i) = m,\ (i \in V_1)$ $gr(j) = n,\ (j \in V_2)$

5.4 Construcción de grafos

En esta sección presentamos algunas operaciones básicas con grafos, esto es, maneras de construir grafos nuevos a partir de otros.

Definición 5.4.1 *Un* **subgrafo** *(respectivamente* **subdigrafo***) de un grafo simple (de un digrafo) $G = (V, E)$ es un grafo (respectivamente digrafo) $H = (V', E')$ tal que $V' \subseteq V$ y $E' \subseteq E$.*

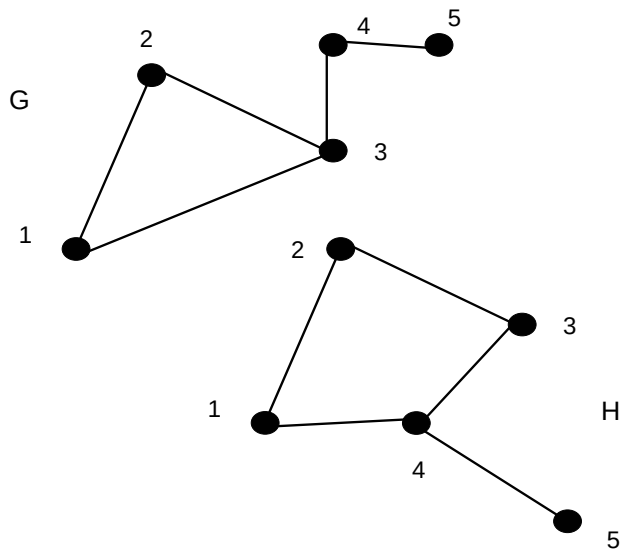


Figura 5.16:

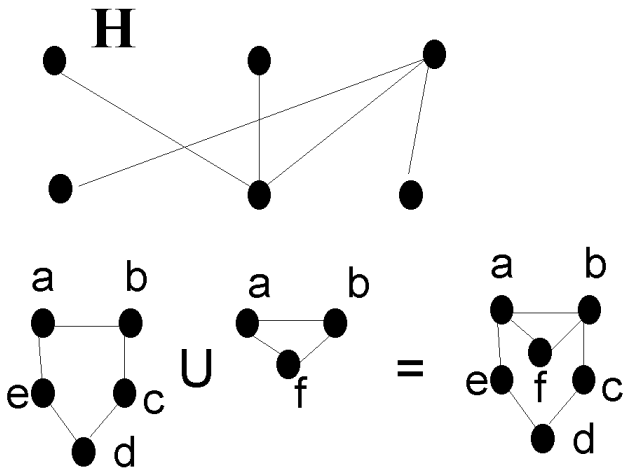


Figura 5.17:

Obsérvese que esta definición se extiende de forma natural a multigrafos o multidigrafos, siendo E' una subfamilia de E.

Observación 5.4.2 *Si un grafo G tiene un subgrafo G' y otro grafo H no tiene ningún subgrafo isomorfo a G' entonces G y H no pueden ser isomorfos. En la figura 5.16 el grafo G tiene un grafo isomorfo a C_3, mientras que H no lo tiene, de modo que G y H no son isomorfos.*

Definición 5.4.3 *La* **unión** *de dos grafos simples $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ es el grafo simple $(V_1 \cup V_2, E_1 \cup E_2)$, grafo que se denota por $G_1 \cup G_2$.*

Ejemplo 5.4.4 *El grafo H de la figura 5.17 es un subgrafo del grafo $K_{3,3}$, y la unión de los grafos G_1 y G_2 de la figura es el grafo $G_1 \cup G_2$ allí representado.*

Ejercicio 114 *Demostrar las afirmaciones del ejemplo anterior.*

Definición 5.4.5 *El* **producto** *de dos grafos simples $G = (V, E)$ y $G' = (V', E')$ es el grafo simple*

$$G \times G' = (V \times V', E")$$

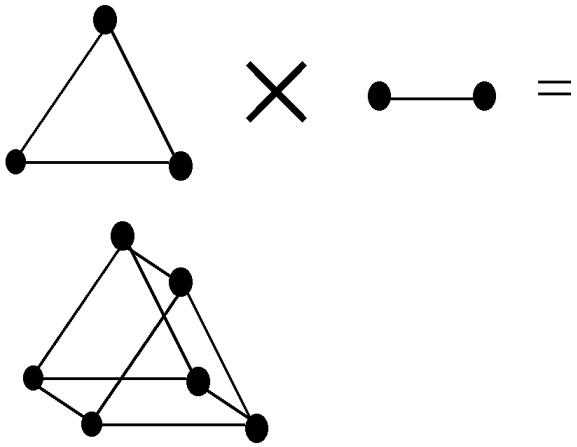


Figura 5.18: Un producto de grafos

donde E" es el conjunto formado por todas las aristas de la forma $\{(a,b),(a',b')\}$ *tales que*

o bien $a = a'$ *y* $\{b,b'\}$ *es una arista de* G',

o bien $b = b'$ *y* $\{a,a'\}$ *es una arista de* G.

Ejemplo 5.4.6 *La figura 5.18 muestra un producto de grafos.*

Ejercicio 115 *Comprobar que el cubo* Q_3 *es isomorfo al grafo producto del cubo bidimensional* Q_2 *y el cubo unidimensional* Q_1. *En general se puede demostrar que* $Q_n = Q_{n-1} \times Q_1$.

Observación 5.4.7 *El grado de un vértice* (v,v') *en el grafo producto* $G \times G'$ *es* $gr((v,v')) = gr(v) + gr(v')$. *Donde* $gr(v)$ *es el grado del vértice* v *de* G *y* $gr(v')$ *es el grado del vértice* v' *en* G'.

Ejercicio 116 *Demostrar que el producto de un grafo simple* G *con* n *vértices y* m *aristas y otro* G' *con* n' *vértices y* m' *aristas es un grafo simple con* nn' *vértices y* $nm' + mn'$ *aristas.*

Definición 5.4.8 *Sea un grafo $G = (V, E)$, diremos que $G' = (V', E')$ es una* **partición** *de G si es un grafo que se obtiene dividiendo algunas de las aristas de G. Este proceso de división consiste en introducir un vértice $a \notin V$ al conjunto de vértices y sustituir una arista $\{u, v\} \in E$ (respectivamente (u, v) si G es dirigido) por dos aristas nuevas, de la forma $\{u, a\}$, $\{a, v\}$ (respectivamente (u, a), (a, v)).*

Ejemplo 5.4.9 *Sea $G = (\{1, 2, 3\}, \{\{1, 2\}, \{2, 3\}\})$ entonces*

$$G' = (\{1, 2, 3, 4\}, \{\{2, 3\}, \{1, 4\}, \{4, 2\}\})$$

es una partición de G, ver figura 5.19.

5.5 Representación de grafos

En esta sección veremos distintas maneras de representar grafos. Según el problema que se quiera abordar, unas u otras representaciones resultan más adecuadas.

5.5.1 Mediante una matriz de adyacencias

Una de las formas de representar un grafo **simple** es mediante una de sus **matrices de adyacencias**.

Dado un grafo $G = (V, E)$, para construir una de sus matrices de adyacencias, necesitamos ordenar sus vértices. Si el grafo tiene n vértices, $|V| = n$, y los ordenamos como $V = \{v_1, v_2, ..., v_n\}$, la matriz de adyacencia de G con respecto a esa ordenación de los vértices es la matriz $A = (a_{ij})$ de n filas y n columnas determinada por la siguiente condición:

$$a_{ij} = \begin{cases} 1 & si \quad \{v_i, v_j\} \in E \\ 0 & si \quad \{v_i, v_j\} \notin E. \end{cases}$$

Ejemplo 5.5.1 *Sea por ejemplo el grafo K_3. Como tiene 3 vértices, cualquier matriz de adyacencias de K_3 debe ser de tamaño 3×3. Siendo el grafo completo, toda matriz de adyacencias está formada por unos salvo en la diagonal, donde hay ceros (obsérvese que en este ejemplo la matriz de adyacencias es independiente de la ordenación del los vértices):*

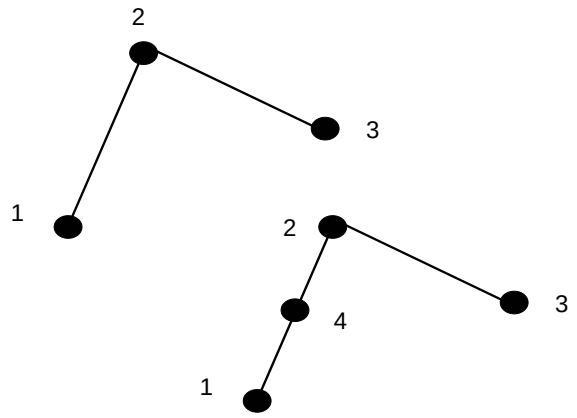


Figura 5.19: Una partición de un grafo

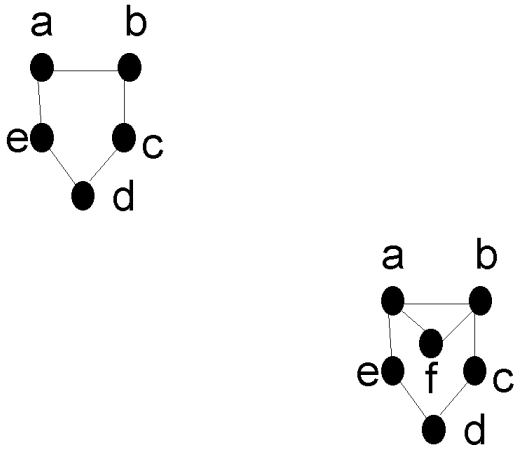


Figura 5.20:

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Observación 5.5.2 *La matriz de adyacencias de un grafo simple es una matriz simétrica ($a_{ij} = a_{ji}$ para cualesquiera i y j) y $a_{ii} = 0$ para cada i.*

Ejercicio 117 *Construir una matriz de adyacencias para cada uno de los siguientes grafos: K_4, C_4, W_3 y $K_{3,2}$.*

Ejercicio 118 *Utilizar una matriz de adyacencias para representar los grafos de la figura 5.20.*

Observación 5.5.3 *Las matrices de adyacencias también se pueden utilizar para representar grafos no dirigidos con lazos y aristas múltiples. Así, un lazo en el vértice v_i viene representado por un 1 en la posición a_{ii} de la matriz de adyacencia. Si se trata de multigrafos, en la posición a_{ij} de la matriz colocaremos el número de aristas que conectan el vértice v_i y el v_j. Así, si tenemos 3 aristas entre el vértice v_i y el v_j, pondremos $a_{ij} = 3$. En cualquier caso, todos los grafos no dirigidos tienen asociadas matrices simétricas.*

Observación 5.5.4 *En el caso de los grafos dirigidos la situación es similar. En la posición a_{ij} aparecerá un 1 si hay una arista dirigida cuyo vértice inicial es v_i y cuyo vértice final es v_j y un cero en caso contrario.*

Obsérvese que las matrices de adyacencias asociadas a grafos dirigidos no son necesariamente simétricas.

5.5.2 Mediante una matriz de incidencias

Otro modo usual de representar grafos es utilizando matrices de incidencias.

Sea $G = (V, E)$ un grafo no dirigido con $|V| = n$ y $|E| = m$. Sea una ordenación de los vértices de G, digamos $v_1, v_2, ..., v_n$, y una ordenación de las aristas de G, digamos $e_1, e_2, ..., e_m$. La **matriz de incidencias** de G con respecto a esa ordenación de los elementos de V y E es la matriz $B = (b_{ij})$ de n filas y m columnas definida por la siguiente condición:

$$b_{ij} = \begin{cases} 1 & si \quad v_i \in e_j \\ 0 & si \quad v_i \notin e_j. \end{cases}$$

Ejemplo 5.5.5 *Siendo $G = (\{a, b, c, d\}, \{\{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}, \{d, b\}\})$, la matriz de incidencias de G con respecto a la ordenación a, b, c, d de sus vértices y $\{a, b\}$, $\{b, c\}$, $\{c, d\}$, $\{d, a\}$, $\{d, b\}$ de sus aristas, es la matriz*

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Observación 5.5.6 *De manera análoga a las matrices de adyacencias, la definición de matriz de incidencias se puede aplicar (mutatis mutandis) a grafos dirigidos o con lazos.*

Observación 5.5.7 *Es importante observar que si dos grafos G y G' son isomorfos, necesariamente existe una ordenación de los vértices (respectivamente de vértices y aristas) de ambos de manera que la matriz de adyacencias (respectivamente de incidencias) es la misma para los dos grafos. Esto no quiere decir, sin embargo, que cada matriz de adyacencias de G sea igual a cada una de G'.*

5.6 Caminos, ciclos y grafos conexos

Los grafos se aplican, entre otras cosas, en modelos de redes de transporte o equivalentemente redes de comunicación. Hay cuestiones importantes sobre estas redes que se relacionan con el concepto de *conexión* y con el de *camino* uniendo dos vértices. Por ejemplo es interesante saber si dos nodos de la red están comunicados entre sí, si todos los nodos están comunicados con todos, o cuál es el mínimo número de cables (o carreteras) que deben extropearse para que la red no conecte todos los nodos entre sí. En este sentido las siguientes definiciones son de utilidad.

Definición 5.6.1 *Un* **camino de longitud** n *entre los vértices* a *y* b *de un grafo no dirigido es una sucesión finita* $(e_0, ..., e_{n-1})$ *de aristas del grafo*

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

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* $(e_0, ..., e_{n-1})$ *queda perfectamente determinado por la sucesión de vértices*

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

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**.

Ejemplo 5.6.2 *En el grafo simple de la figura 5.21,* (a, b, g, d, c, a) *es un circuito simple de longitud* 5 *y* (a, b, d, c, e, f, d) *es un camino simple de longitud* 6 *entre los vértices* a *y* d.

La definición anterior se puede extender a grafos dirigidos (digrafos y multidigrafos):

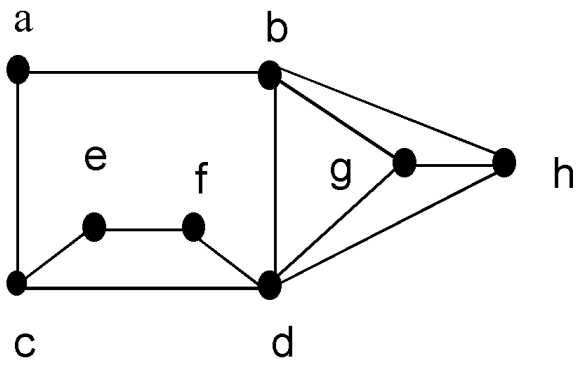


Figura 5.21:

Definición 5.6.3 *Un* **camino de longitud** n *entre los vértices a y b de un multigrafo dirigido es una sucesión finita $(e_0, ..., e_{n-1})$ de aristas del multigrafo dirigido:*

$$e_0 = (v_0, v_1), e_1 = (v_1, v_2), ..., e_{n-1} = (v_{n-1}, v_n)$$

de manera que $v_0 = a$, $v_n = b$ y cada arista sucesiva empieza donde terminó la anterior. Cuando se trata de un digrafo, el camino $(e_0, ..., e_{n-1})$ queda perfectamente determinado por la sucesión de vértices

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

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 $a = b$.*

Al igual que en el caso de los grafos no dirigidos, se dice que un camino es **simple** *si no contiene a la misma arista más que una vez.*

5.6.1 Conexión

Definición 5.6.4 *Se dice que un grafo no dirigido G es* **conexo** *si para cualquier par de vértices a y b de G existe un camino entre a y b.*

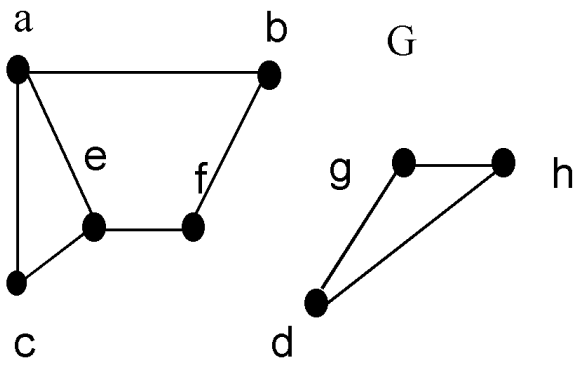


Figura 5.22:

Ejemplo 5.6.5 *El grafo G de la figura 5.22, cuyo conjunto de vértices es*

$$V = \{a, b, c, d, e, f, g, h\}$$

no es conexo pues, por ejemplo, no existe ningún camino entre a y g.

Si un grafo no es conexo, se puede expresar como la unión de dos o más subgrafos conexos de manera que los conjuntos de vértices y de aristas de cada par de estos subgrafos son disjuntos entre sí. A estos subgrafos se les denomina **componentes conexas** del grafo dado. De esta manera un grafo es conexo si y sólo si tiene una única componente conexa.

Definición 5.6.6 *Dado $G = (V, E)$ un grafo simple, $H = (V', E')$ subgrafo de G es una* **componente conexa** *de G si verifica:*

i) H es conexo;
ii) si $\{u, v\} \in E$ y $u \in V'$ entonces $v \in V'$ y $\{u, v\} \in E'$.

Ejemplo 5.6.7 *El grafo de la figura 5.18 tiene dos componentes conexas: la que tiene como vértices al conjunto $\{a, b, c, e, f\}$ y las correspondientes aristas, y la que tiene como conjunto de vértices $\{g, h, d\}$ y las correspondientes aristas.*

Teorema 5.6.8 *Existe un camino simple entre cualquier par de vértices distintos de un grafo conexo no dirigido.*

Demostración. Lo demostraremos para grafos simples, si hay aristas múltiples, el mismo razonamiento es válido definiendo los caminos por las aristas y no por los vértices que recorren.

Sean a y b dos vértices cualesquiera de un grafo simple $G = (V, E)$. Puesto que G es conexo, existe (al menos) un camino entre a y b. Sea (v_0, v_1, v_2, ..., v_{n-1}, v_n) un camino entre a y b de la menor longitud posible. En ese caso dicho camino es simple: lo demostramos por reducción al absurdo. Si no fuera simple, en particular tendríamos que $v_i = v_j$ para algunos $i, j \in \{1, ..., n\}$, $i < j$. Pero entonces el camino v_0, ..., v_{i-1}, v_j, v_{j+1} ..., v_n es un camino entre a y b de longitud estrictamente menor que n (puesto que se ha obtenido eliminando una o varias aristas del camino anterior), en contradicción con que el camino considerado originalmente entre a y b sea de la menor longitud posible.

Conexión e isomorfismo

Los conceptos relacionados con la conexión nos proporcionan nuevos invariantes para averiguar si dos grafos son o no isomorfos.

Observación 5.6.9 *Dos grafos isomorfos tienen la misma cantidad de componentes conexas.*

Definición 5.6.10 *Se dice que un vértice $a \in V$ de un grafo $G = (V, E)$ es un* **vértice de corte** *si el grafo obtenido al eliminar del grafo G el vértice a junto con las aristas incidentes con él tiene más componentes conexas que el grafo original. Del mismo modo se dice que una arista $e \in E$ es una* **arista de corte** *de G si el subgrafo obtenido al eliminar e del grafo original tiene más componentes conexas que el grafo original.*

Los puntos de corte y las aristas de corte son invariantes que nos pueden permitir concluir que dos grafos no son isomorfos. Si un grafo presenta un vértice de corte y otro no, los dos grafos no pueden ser isomorfos (igualmente con las aristas).

Ejemplo 5.6.11 *Los grafos G y H de la figura 5.23 no pueden ser isomorfos, pues aunque ambos tienen 6 vértices y siete aristas, el grafo H no tiene*

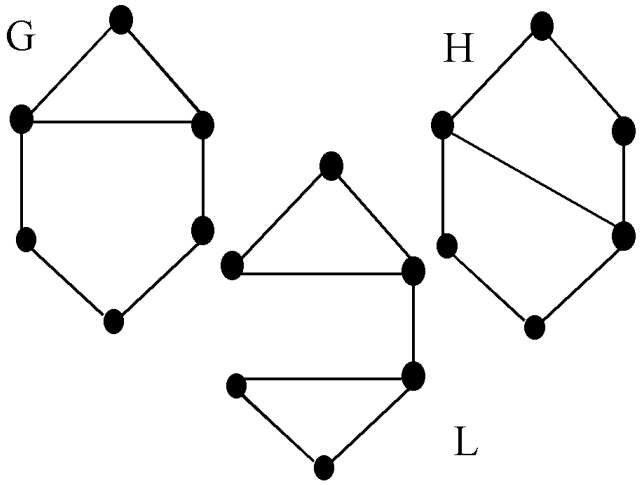


Figura 5.23:

ningún subgrafo isomorfo a C_3, mientras que el grafo G sí lo tiene. Por otra parte, ni G ni H son isomorfos al grafo L, pues L tiene una arista de corte y G y H no tienen ninguna.

Conexión y matrices

La representación mediante matrices de los grafos aporta información referida al número de caminos que unen dos vértices y a la conexión del grafo.

Teorema 5.6.12 *Sea G un grafo (dirigido o no dirigido, con aristas múltiples y lazos o no) y sea $A = (a_{ij})$ su matriz de adyacencias con respecto al orden $v_1, v_2, ..., v_{n-1}, v_n$ de su conjunto de vértices. En estas condiciones el número de caminos de longitud m entre el vértice v_i y el vértice v_j es igual al coeficiente situado en el lugar (i, j) de la potencia m-ésima de la matriz A (con respecto al producto de matrices usual).*

Demostración. La hacemos para grafos no dirigidos (la demostración es similar para grafos dirigidos). Razonamos por inducción sobre la longitud del camino entre dos vértices cualesquiera v_i y v_j de un grafo G.

Base de inducción: El número de caminos de longitud 1 entre dos vértices cualesquiera v_i y v_j es el coeficiente (i,j) de la matriz A, ya que dicho coeficiente es el número de aristas entre v_i y v_j.

Paso de inducción: Supongamos que el número de caminos de longitud m entre dos vértices cualesquiera v_i y v_j es el coeficiente (i,j) de la matriz A^m. Tenemos que comprobar que el número de caminos de longitud $m+1$ entre v_i y v_j es el coeficiente (i,j) de la matriz A^{m+1}. Cómo $A^{m+1} = A^m \cdot A$, siendo $A^m = (b_{ij})$ y $A = (a_{ij})$, el coeficiente (i,j) de la matriz A^{m+1} es

$$b_{i1}a_{1j} + b_{i2}a_{2j} + ... + b_{in}a_{nj}.$$

Por hipótesis de inducción, b_{ik} es el número de caminos de longitud m entre v_i y v_k. Un camino de longitud $m+1$ entre v_i y v_j es un camino de longitud m entre v_i y un vértice adyacente a v_j, al que denotamos por v_k, seguido de la arista que une v_k y v_j. Pero el número de caminos entre v_i y v_k es b_{ik}, y el número de aristas entre v_k y v_j es a_{kj}, por lo que el número de caminos de longitud $m+1$ entre v_i y v_j viene dado por la expresión:

$$b_{i1}a_{1j} + b_{i2}a_{2j} + ... + b_{in}a_{nj}.$$

Como queríamos demostrar.

La observación siguiente es una consecuencia del Principio del Palomar y la demostración es idéntica a la del teorema 5.6.8.

Observación 5.6.13 *Sea el grafo $G = (V,E)$ con $|V| = n$. Si $u, v \in V$, $u \neq v$, están unidos por un camino en G, entonces u y v están unidos por un camino de longitud menor o igual que $n-1$ en G.*

De la observación anterior y los resultados anteriores se concluyen los siguientes corolarios.

Corolario 5.6.14 *Dado un grafo $G = (V,E)$ tal que $|V| = n$, y siendo $A = (a_{ij})$ una matriz de adyacencia de G, se verifica que existe un camino entre v_i y v_j si y sólo si la matriz $C = (c_{ij})$, definida como*

$$C = I_n + A + A^2 + ... + A^{n-1},$$

satisface que el coeficiente $c_{ij} \neq 0$.

Corolario 5.6.15 *Dado un grafo $G = (V, E)$ tal que $|V| = n$, se verifica que G es conexo si y sólo si la matriz*

$$I_n + A + A^2 + ... + A^{n-1}$$

tiene todos los coeficientes distintos de cero.

Ejemplo 5.6.16 *Sea G un grafo cuya matriz de adyacencias es:*

$$A := \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Entonces su cuadrado es

$$A^2 := \begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Por tanto $I+A+A^2$ tiene todas sus entradas no nulas y el grafo es conexo. El corolario anterior nos permite obtener un algoritmo para determinar si un grafo $G = (V, E)$ tal que $|V| = n$ es o no conexo. Usaremos un algoritmo llamado $Hayceros(M)$ que determina si en la matriz M hay alguna entrada cero (tiene como salida un 1 si hay algún cero y 0 en caso contrario):

Entrada: *La matriz de adyacencia A del grafo*
$j := 1$, $B := I_n$
while $j \leq n - 1$
$B := B + A^j$
$j := j + 1$
If $Hayceros(B) = 0$ then $s :=$conexo else $s :=$no conexo.
Salida: *s.*

Ejercicio 119 *Estudiar la complejidad del algoritmo anterior en el peor de los casos, para lo cual hay que abordar el problema de computar la complejidad de la operación producto de matrices.*

5.7 Grafos eulerianos y hamiltonianos.

En esta sección daremos las definiciones precias para resolver el problema de los puentes de Könisberg.

Definición 5.7.1 *Un* **camino euleriano** *en un grafo no dirigido G es un camino simple que contiene a todas las aristas de G. Un* **circuito euleriano** *en G es un circuito simple que contiene todas las aristas del grafo G.*

Definición 5.7.2 *Se dice que un grafo no dirigido es* **euleriano** *si contiene un circuito euleriano.*

De la definición de grafo euleriano se sigue que, o bien el grafo es conexo, o bien hay una componente conexa que contiene todas las aristas (siendo el resto de las componentes conexas vértices aislados).

Teorema 5.7.3 *Un grafo $G = (V, E)$ con aristas no dirigidas es euleriano si y solo si todas las aristas están en la misma componente conexa y todos los vértices tienen grado par.*

Demostración. Si el grafo G es euleriano, entonces contiene un circuito euleriano que, por definición, contiene a todas las aristas. En consecuencia, todas las aristas están en la misma componente conexa. Por otra parte, todos los vértices tienen grado par, pues cada vez que el circuito euleriano pasa por un vértice lo hace una vez para entrar y otra para salir con lo que el grado de cada vértice es par. Obsérvese que los posibles vértices aislados tienen grado 0, un número par.

Para la demostración del recíproco, el siguiente algoritmo muestra un procedimiento constructivo para generar un circuito euleriano en un multigrafo conexo en el que todos los vértices tienen grado par.

Por hipótesis, todas las aristas están en la misma componente conexa, por lo que se puede suponer que el grafo es conexo, prescindiendo de los vértices aislados. Sea entonces G un multigrafo conexo con todos los vértices de grado par. Además, podemos prescindir de los posibles lazos: si G multigrafo conexo tiene lazos, es equivalente G euleriano que G' euleriano, siendo G' el grafo resultante de quitar los lazos a G. Por tanto G es un multigrafo conexo sin lazos.

Necesitamos unos algoritmos previos que describimos en los párrafos siguientes.

El primero de ellos elimina de un grafo G las aristas de un circuito previamente elegido C que es subgrafo de G y los vértices que quedan aislados tras esta sustracción. Lo denominamos borrado y será llamado luego al construir el algoritmo de búsqueda de un circuito euleriano:

Algoritmo $Borrado(C, G)$

Entrada: $G = (V, E), C = (V_C, E_C)$,
(donde G es un grafo conexo y sin lazos y C es un circuito simple de G)
$E_H = E - E_C$
$V_{aislados}$ son los vértices aislados de (V, E_H)
$V_H = V - V_{aislados}$
Salida: (V_H, E_H)

Queda como ejercicio diseñar el algoritmo que computa los vértices aislados, denominado $V_{aislados}$.

El siguiente algoritmo inserta un circuito C' en un circuito C de manera que se forma un circuito de longitud mayor:

Algoritmo $Insertado(C', C)$

Entrada: G, C, C'
(donde G es un grafo conexo y sin lazos, C y C' son circuitos simples de G, C' parte de un vértice interior de C)
$C := v_1, \ldots, v_s, v_1$
$C' := w_1, \ldots, w_t, w_1$
$v_i = w_1$
(consideramos C y C' como listas)
$V := v_1, \ldots, v_i, w_2, \ldots, w_t, v_i, v_{i+1}, \ldots, v_s, v_1$
(V es una lista, resultado de insertar adecuadamente la lista C' en la lista C, esto se puede hacer llamando a los apropiados algoritmos sobre listas del capítulo segundo.)

$E := \emptyset$
for $i = 1$ to $t + s + 1$
$E := E \cup \{\{v_i, v_{i+1 \bmod t+s}\}\}$

Salida: $(\{V\}, E)$

De esta manera podemos construir el algoritmo buscado:

Algoritmo de búsqueda de un circuito euleriano

Entrada: $G = (V, E)$
(donde G es un grafo conexo y sin lazos)
$C :=$ es un circuito simple de G
$H = (V_H, E_H) := Borrado(C, G)$
while $V_H \neq \emptyset$

C' cualquier circuito simple de H con origen en un vértice interior de C

$H := Borrado(C', H)$
$C := Insertado(C', C)$

Salida: C

Para que el algoritmo funcione necesitamos asegurar la existencia del circuito simple denominado C. Esto es, que dado un multigrafo conexo con todos sus vértices de grado par, existe un circuito simple C. Tomamos un vértice cualquiera u. Vamos construyendo un camino simple $T = (u_0 = u, u_1, ..., u_n)$ de longitud máxima. Si existen $i, j \in \{0, ..., n\}$ tales que $u_i = u_j$ hemos concluido. En caso contrario, como u_n tiene grado par, existe $v \neq u_{n-1}$ adyacente con u_n. Si $v \in \{u_0, ..., u_{n-1}\}$ aparece un ciclo. Si no, se contradice el hecho de que T es de longitud máxima. Por tanto el algoritmo funciona.

Veamos como se aplica el algoritmo anterior en un ejemplo concreto:

Ejemplo 5.7.4 *Un cartero tiene que repartir sus cartas en la red de calles representada por el grafo de la figura 5.24. Para realizar el reparto, el cartero debe empezar y terminar en la estafeta de correos que se encuentra en el vértice i. Teniendo en cuenta que todos los vértices tienen grado par, el cartero sabe que puede efectuar el reparto sin recorrer dos veces la misma calle, construyendo para ello un circuito euleriano. Comenzamos con el circuito (i, j, h, i) y borramos sus aristas del grafo, junto con el vértice i que queda aislado (ver figura 5.25).*

A continuación, construimos un nuevo circuito en el grafo que queda, por ejemplo (j, m, l, k, j), lo insertamos en el circuito anterior en el lugar adecuado (vértice j) obteniendo

$$(i, j, m, l, k, j, h, i).$$

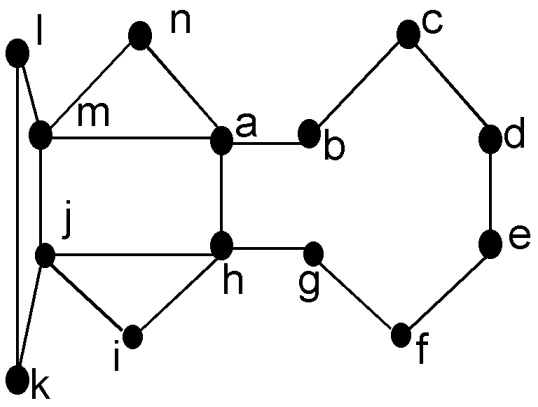


Figura 5.24:

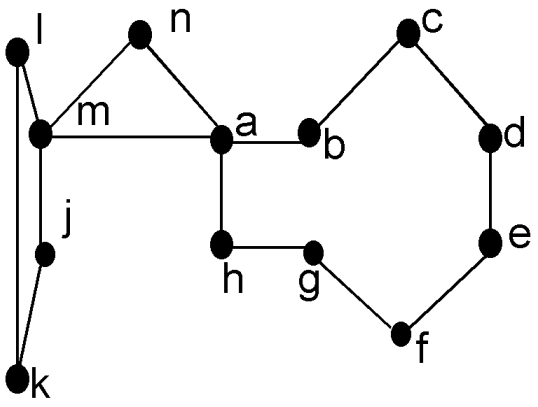


Figura 5.25:

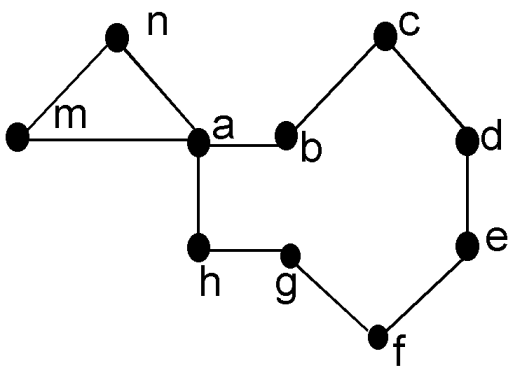


Figura 5.26:

Borramos las aristas y vértices aislados (Ver figura 5.26).

Prosiguiendo con el algoritmo, construimos el circuito (m, n, a, m) *y lo insertamos en el lugar adecuado (vértice* m*) obteniendo*

$$(i, j, m, n, a, m, l, k, j, h, i).$$

Finalmente, nos queda un el circuito $(a, b, c, d, e, f, g, h, a)$ *que, insertado en el lugar adecuado, da lugar al siguiente circuito euleriano del grafo inicial:*

$$(i, j, m, n, a, b, c, d, e, f, g, h, a, m, l, k, j, h, i).$$

Ejercicio 120 *¿Es posible disponer todas las fichas de un dominó de manera que estén todas encajadas? Indicación: utilizar un multigrafo con 7 vértices y verificar que es euleriano y conexo.*

El siguiente resultado nos da una condición para la existencia de un camino euleriano no cerrado:

Proposición 5.7.5 *Un grafo* $G = (V, E)$ *con aristas no dirigidas tal que todas sus aristas están en la misma componente conexa admite un camino euleriano no cerrado si y sólo si contiene exactamente dos vértices de grado impar.*

Demostración. Supongamos que $G = (V, E)$ admite un camino euleriano no cerrado

$$a, v_1, v_2, ..., v_{n-1}, b$$

siendo a y b los extremos del camino. Sea w un nuevo vértice no perteneciente a V y $G' = (V', E')$ donde $V' = V \cup \{w\}$ y $E' = E \cup \{\{w, a\}, \{b, w\}\}$. Es evidente que G' admite el circuito euleriano

$$w, a, v_1, v_2, ..., v_{n-1}, b, w$$

y, en consecuencia, todos los vértices de G' son de grado par. Al eliminar las aristas $\{w, a\}$ y $\{b, w\}$ y el vértice w de G' para obtener G concluimos que todos los vértices son de grado par salvo a y b.

Recíprocamente, si todos los vértices de $G = (V, E)$ salvo dos, a y b, tienen grado par, construyendo el grafo $G' = (V', E')$ donde $V' = V \cup \{w\}$ y $E' = E \cup \{\{w, a\}, \{b, w\}\}$ con w un nuevo vértice no perteneciente a V, obtendremos un grafo en el que todos los vértices son de grado par. Siendo

$$w, a, v_1, v_2, ..., v_{n-1}, b, w$$

un circuito euleriano,

$$a, v_1, v_2, ..., v_{n-1}, b$$

es un camino euleriano no cerrado que conecta a y b.

Ahora ya estamos en condiciones de demostrar que no es posible encontrar un circuito que resuelva el problema de los puentes de Königsberg, pues el grafo que representa el problema tiene más de dos vértices de grado impar.

Ejercicio 121 *Construir el grafo que representa la configuración del problema de los puentes de Könisberg. Demostrar que dicho grafo no es euleriano ni admite un camino euleriano.*

Un problema análogo al de saber si un grafo es euleriano, pero referido a las aristas, es el que sigue.

Definición 5.7.6 *Se denomina* **camino hamiltoniano** *en un grafo con aristas no orientadas $G = (V, E)$ a cualquier camino simple que contenga a todos los vértices de G pasando una sola vez por cada uno de ellos, pero permitiendo que el vértice inicial de dicho camino sea igual al vértice final. Si el camino hamiltoniano es cerrado, a dicho camino se le denomina* **circuito hamiltoniano**.

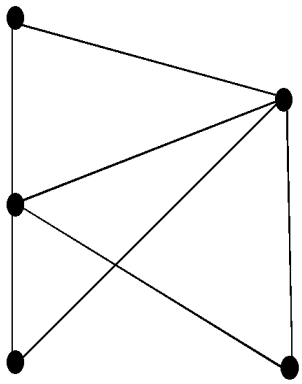


Figura 5.27: Un grafo Euleriano

Definición 5.7.7 *Se dice que un* **grafo** *no dirigido* $G = (V, E)$ *es* **hamiltoniano** *si contiene un circuito hamiltoniano.*

Los grafos hamiltonianos tienen su origen en el siguiente juego propuesto por Hamilton: encontrar un circuito que, pasando por todos los vértices del dodecaedro (a través de las aristas), termine en el de vértice de partida sin pasar dos veces por el mismo vértice (salvo el de partida).

Pese a la aparente similitud con la definición de los grafos eulerianos (en un caso el énfasis del estudio está puesto en los vértices y en el otro en las aristas) encontrar un circuito hamiltoniano en un grafo puede no ser tarea fácil pues no se conoce una caracterización de los grafos hamiltonianos.

Veamos algunos resultados sencillos relacionados con los grafos hamiltonianos (alguno de ellos sin demostración).

Teorema 5.7.8 *El cubo* Q_n *es hamiltoniano para cada* $n \geq 2$.

Demostración. Razonamos por inducción sobre n.

Base de inducción: Q_2 es hamiltoniano $((0,0),(0,1),(1,1),(1,0),(0,0)$ es un ciclo hamiltoniano de Q_2).

Paso de inducción: Supongamos que Q_n es hamiltoniano e identificamos Q_{n+1} con el producto $Q_n \times Q_1$. Por hipótesis de inducción Q_n tiene

un circuito hamiltoniano. Eliminando la misma arista en cada uno de los circuitos hamiltonianos de las dos copias de Q_n y cerrando el grafo obtenido añadiendo las aristas que conectan los extremos de los caminos simples resultantes se obtiene un circuito hamiltoniano de Q_{n+1}.

Los circuitos de hamilton tienen aplicación al denominado problema de los códigos de Gray, consistente en determinar si hay una ordenación de las palabras de n bits, de manera que situando una palabra en cada una de las secciones circulares (quesitos) en los que hemos dividido un círculo, dos secciones contiguas difieran en un único bit.

Veamos cómo se pueden construir códigos de Gray a partir de circuitos hamiltonianos en Q_n.

Consideremos los vértices de Q_n (i.e. las palabras de n bits) según indicamos en su construcción (i.e., de forma que dos vértices adyacentes difieran en un único bit). Cualquier circuito hamiltoniano sobre el cubo Q_n da lugar a un código de Gray. Por ejemplo, el siguiente camino hamiltoniano de Q_3 da lugar a un código de Gray (para ello, es suficiente representarlo de forma circular según se ha indicado):

$$(0,0,0),(1,0,0),(1,1,0),(1,1,1),$$

$$(1,0,1),(0,0,1),(0,1,1),(0,1,0),(0,0,0).$$

Ejemplo 5.7.9 *El grafo de la figura 5.28 es hamiltoniano. Se muestra un circuito hamiltoniano.*

Como se comentó, no siempre es sencillo determinar si un grafo simple y conexo dado es hamiltoniano pues, para ello, el único criterio que tenemos a priori es nuestra habilidad para encontrar o no un circuito hamiltoniano en el grafo dado. La siguiente proposición nos aporta un resultado que permite concluir que ciertos grafos no son hamiltonianos.

Proposición 5.7.10 *Sea $G = (V, E)$ un grafo simple conexo tal que $|V| \geq 3$. Si G es hamiltoniano, entonces para cada subconjunto $U \subseteq V$ el grafo obtenido al eliminar de V los vértices de U y las aristas incidentes con dichos vértices tiene a lo sumo $|U|$ componentes conexas.*

Demostración. Si el grafo G es hamiltoniano, entonces contiene un circuito hamiltoniano

$$a, v_1, v_2, ..., v_{n-1}, a.$$

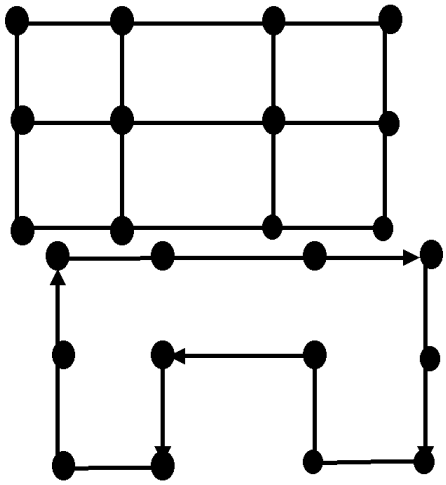


Figura 5.28:

Sea H el grafo formado por los vértices y aristas del circuito anterior, y sea $U \subseteq V$. (Obsérvese que los vértices de H son los mismos que los del grafo original G). Sea k el número de componentes conexas del grafo $(V-U, E')$ donde E' es el subconjunto de aristas de G caracterizado por el hecho de que sus extremos pertenecen a $V-U$, y sea k' el número de componentes conexas del subgrafo de H obtenido al eliminar de H los vértices pertenecientes a U junto con las aristas incidentes con ellos. Evidentemente, $k \leq k'$. Puesto que el grafo H se puede representar como en la figura 5.29, es obvio que al eliminar un punto (y las aristas incidentes con él) el nuevo grafo así obtenido es conexo (es decir, tiene una única componente conexa), al eliminar dos puntos (y las aristas incidentes con ellos) el nuevo grafo así obtenido tiene como mucho dos componentes conexas y así sucesivamente. En general si quitamos p vértices (junto con sus aristas incidentes) obtenemos un grafo con, a lo sumo, p componentes conexas. Por consiguiente $k \leq k' \leq p = |U|$.

El resultado anterior nos aporta una nueva herramienta con la que podemos demostrar que algunos grafos no son hamiltonianos.

Ejemplo 5.7.11 *Si al grafo G de la figura 5.30 le quitamos el único vértice de grado 5 que tiene y sus aristas incidentes, el grafo resultante tiene 3 componentes conexas, por tanto no es hamiltoniano.*

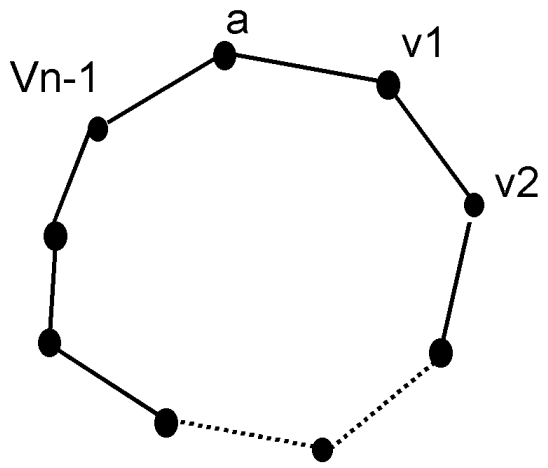


Figura 5.29:

Ejercicio 122 *Determinar si el grafo H de la figura 5.30 es Hamiltoniano.*

5.8 Grafos etiquetados y algoritmo de Dijkstra

Puede ser interesante, por ejemplo cuando un grafo representa una red de comunicaciones, añadir alguna información a las aristas, que mida por ejemplo dificultades orográficas, longitudes de caminos, resistencias de materiales... En este sentido es de utilidad la siguiente definición.

Definición 5.8.1 *Un* **grafo etiquetado** *es una 3-tupla (V, E, d), en la que (V, E) es un grafo simple y d es una función*

$$d : E \to \mathbb{R}.$$

A la imagen de una arista mediante d se le denomina **etiqueta** *o* **peso** *de la arista. (Ver como ejemplo la figura 5.31)*

El siguiente algoritmo, conocido como algoritmo de Dijkstra, resuelve el denominado problema del camino mínimo.

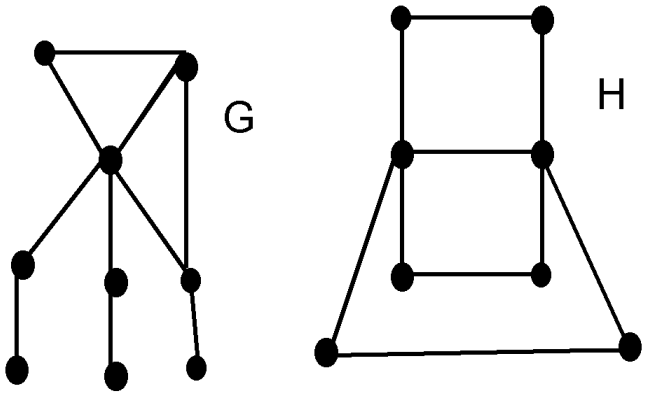


Figura 5.30:

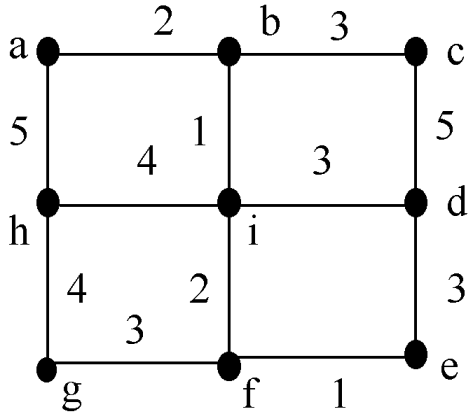


Figura 5.31:

Problema. Dados un grafo etiquetado y dos vértices u y v, determinar un camino entre u y v tal que la suma de las etiquetas de las aristas que lo componen sea mínima. A dicha suma se le denomina longitud del camino en el grafo etiquetado

Nota: En el algoritmo siguiente se entiende que:

i) ∞ es mayor que cualquier número real,
ii) G es un grafo etiquetado conexo,
iii) para cada par de vértices no adyacentes x, y se tiene que $d(\{x, y\}) = \infty$.

Algoritmo de Dijkstra.

Entrada: $G = (V, E, d)$, $u, v \in V$
$L(u) := 0$
$L(x) := \infty$ para cada $x \neq u$
$T := \emptyset$
while $v \notin T$
  $x := a$ siendo a un vértice que no esté en T con $L(a)$ mínimo
  $T := T \cup \{x\}$
  for $y \notin T$
    if $L(x) + d(\{x, y\}) < L(y)$ then
      $L(y) := L(x) + d(\{x, y\})$,
      $f(y) := x$
Salida: $L(v)$ es la longitud del camino mínimo.
  $(v, f(v), f(f(v)), ..., u)$ es el camino mínimo de v a u.

Ejemplo 5.8.2 *Buscamos por este algoritmo el camino mínimo entre a y d en el grafo etiquetado de la figura 5.31.*

Comienza el algoritmo asignando $L(a) = 0$ y $L(x) = \infty$ para el resto de vértices. De este modo:

$$T = \{a\}.$$

Ahora se cumple que:

$$L(a) + 2 = 2 < L(b) = \infty \qquad L(a) + 5 = 5 < L(h) = \infty.$$

Con lo que los únicos valores para los que L cambia son:

$$L(b) = 2 \qquad L(h) = 5.$$

Y además:

$$f(b) = f(h) = a.$$

Ahora el valor mínimo de L lo tiene el vértice b y $b \notin T$, con lo que

$$T = \{a, b\}.$$

Ahora se cumple que:

$$L(b) + 3 = 5 < L(c) = \infty \qquad L(a) + 1 = 3 < L(i) = \infty.$$

Con lo que los únicos valores para los que L cambia son:

$$L(c) = 5 \qquad L(i) = 3.$$

Y además:

$$f(c) = f(i) = b.$$

Ahora el valor mínimo de L lo tiene el vértice i con lo que

$$T = \{a, b, i\}.$$

Ahora se cumple que:

$$L(i) + 3 = 6 < L(d) = \infty \qquad L(i) + 2 = 5 < L(f) = \infty.$$

Con lo que los únicos valores para los que L cambia son:

$$L(d) = 6 \qquad L(f) = 5.$$

Y además:

$$f(d) = f(f) = i.$$

Ahora el valor mínimo de L lo tienen los vértices c, f y h con lo que elijo uno de ellos, por ejemplo el f, de modo que:

$$T = \{a, b, i, f\}.$$

Ahora se cumple que:

$$L(f)+1=6<L(e)=\infty \qquad L(f)+3=8<L(g)=\infty.$$

Con lo que los únicos valores para los que L cambia son:

$$L(e)=6 \qquad L(g)=8.$$

Y además:

$$f(e)=f(g)=f.$$

Ahora el valor mínimo de L lo tienen los vértices c, y h con lo que elijo uno de ellos, por ejemplo el c, de modo que:

$$T=\{a,b,i,f,c\}.$$

No se cumple nunca la condición para que haya cambios. El valor mínimo de L lo tiene entonces el vértice h con lo que:

$$T=\{a,b,i,f,c,h\}.$$

Tampoco ahora se cumple nunca la condición para que haya cambios. Ahora el valor mínimo de L lo tienen los vértices d, e con lo que elijo uno de ellos, por ejemplo el d (así el algoritmo terminará en el paso siguiente), de modo que:

$$T=\{a,b,i,f,c,h,d\}.$$

Tampoco ahora se cumple nunca la condición para que haya cambios.

De esta manera el camino más corto entre a y d tiene longitud 6 y es:

$$(d, f(d)=i, f(i)=b, f(b)=a).$$

Ejercicio 123 *Utilizar el algoritmo de Dijkstra para encontrar el camino mínimo entre los vértices a y e del grafo etiquetado de la figura 5.28. Hallar también el camino más corto entre los vértices g y c.*

Los caminos mínimos son muy útiles en redes de comunicación o redes de transporte para optimizar la manera de enviar información o carga entre dos puntos conectados por una red. Los pesos pueden representar distancias o pueden recoger otro tipo de información: tiempo, coste, dificultad orográfica...

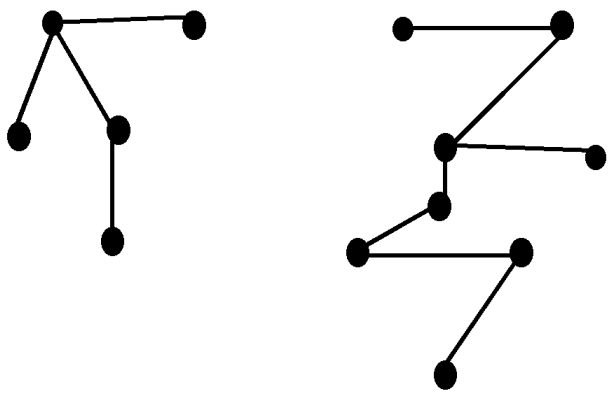


Figura 5.32:

5.9 Árboles

Especialmente útil en lo que a aplicaciones informáticas se refiere es un cierto tipo de grafo simple, llamado árbol, que se emplea, entre otras cosas, para construir algoritmos eficientes destinados a localizar items en una lista, para construir redes de ordenadores con el mínimo coste, para construir códigos eficientes destinados a almacenar y transmitir datos, para analizar algoritmos de ordenación...

Definición 5.9.1 *Un* **árbol** *es un grafo no dirigido, conexo y sin circuitos simples.*

Observación 5.9.2 *Como un árbol no tiene circuitos simples, tampoco puede tener aristas múltiples o lazos, por lo que cualquier árbol es un grafo simple.*

Ejemplo 5.9.3 *Los grafos de la figura 5.29 son árboles.*

Veamos una caracterización de los árboles.

Proposición 5.9.4 *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.*

Demostración. Si G es un árbol entonces es conexo, por lo que existe un camino simple que une u con v. Supongamos que existieran dos caminos simples distintos entre u y v, digamos $C_1 = (u_0 = u, ..., u_n = v)$ y $C_2 = (v_0 = u, ..., v_m = v)$. Como $C_1 \neq C_2$ existe $i \in \{1, ..., n\}$ mínimo tal que $u_i \neq v_i$. Pero como C_1 y C_2 terminan en v existen $j_1 > i$ y $j_2 > i$ mínimos tales que $u_{j_1} = v_{j_2}$. Por tanto se forma un circuito simple

$$(u_{i-1}, u_i, u_{i+1}, ...u_{j_1} = v_{j_2}, v_{j_2-1}..., v_{i-1} = u_{i-1})$$

y tenemos una contradicción.

Si para cada par de vértices u y v hay un camino que los une, entonces G es conexo por definición de conexión. Si G contuviera un circuito simple C, tomando un par de vértices del circuito u y v, existen dos caminos simples distintos que los unen, en contradicción con la hipótesis. Esto demuestra la otra implicación.

En muchas aplicaciones de los árboles se suele escoger un vértice particular al que se denomina **raíz**. Así pues:

Definición 5.9.5 *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, como en la figura 5.30.*

Observación 5.9.6 *Cabe observar que un árbol puede dar lugar a varios árboles con raíz, dependiendo del vértice al que se distinga del resto como raíz. La elección de una raíz lleva aparejada la trasformación del árbol considerado en un árbol con aristas dirigidas, en el que la dirección de las aristas incidentes con la raíz es la que parte de la raíz hacia los vértices unidos a él y a partir de estos sucesivamente hacia el resto de los vértices del árbol.*

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

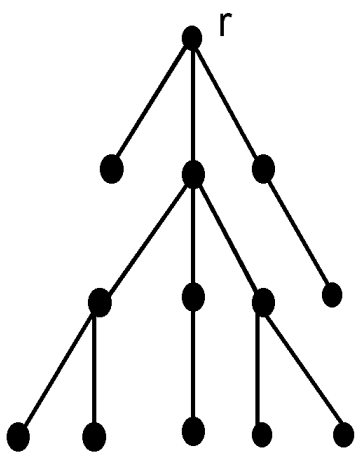


Figura 5.33:

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.

Definición 5.9.7 *Dado un árbol con raíz (G, r) se denominan* **hojas** *a los vértices de G distintos de r que tienen grado 1.*

Es fácil darse cuenta de que las hojas de un árbol dirigido son los vértices que no tienen descendientes. A los vértices distintos de la raíz que no son hojas de un árbol con raíz se les denomina **vértices internos**.

Definición 5.9.8 *Se denomina* **nivel** *(o* **profundidad***) de un vértice en un árbol con raíz a la longitud del camino que une la raíz con dicho vértice. La* **altura** *de un árbol con raíz es el mayor de los niveles de sus vértices (i.e., la longitud del camino más largo posible entre la raíz y un vértice del árbol).*

Definición 5.9.9 *Si a es un vértice de un árbol con raíz (T, r), el* **subárbol** *de T que tiene a a como raíz es el subgrafo del árbol formado por el vértice a, todos sus descendientes y todas las aristas incidentes con sus descendientes.*

Definición 5.9.10 *Se dice que un árbol con raíz es* **m-ario** *si todos los vértices tienen a lo sumo m hijos. Si todos los vértices internos (es decir, todos salvo las hojas) de un árbol $m-$ario con raíz tienen exactamente m hijos, se dice que el árbol es un árbol* **m-ario completo**. *A los árboles 2-arios se les denomina* **árboles binarios**.

Ahora estamos en condiciones de establecer la siguiente propiedad:

Proposición 5.9.11 *Un árbol con n vértices tiene n-1 aristas.*

Demostración. Sea $T = (V, E)$. Elijamos un vértice r como raíz del árbol T. A continuación consideramos la función que asigna a cada arista su vértice final, considerando la dirección determinada por la elección de la raíz. Evidentemente, esta función es una función biyectiva entre E y $V - \{r\}$, y en consecuencia si $|V| = n$, necesariamente $|E| = n - 1$.

Proposición 5.9.12 *En un árbol m-ario de altura h hay a lo sumo m^h hojas.*

Demostración. Razonamos por inducción completa sobre la altura.

Base de inducción: Si T es un árbol cuya altura es 1, entonces T consta de una raíz y no más de m hijos, cada uno de los cuales es una hoja. Por consiguiente no hay más que $m^1 = m$ hojas en un árbol $m-$ario de altura 1.

Paso de inducción: Supongamos que el resultado es válido para todos los árboles $m-$arios de altura menor que h. Sea T un árbol de altura h. Si borramos las aristas que parten de la raíz hacia la primera generación, construimos una serie de árboles (como máximo m ya que T es m-ario) de altura estrictamente menor que h. Por hipótesis de inducción cada uno de estos árboles tiene como máximo m^{h-1} hojas. Como hay como máximo m de estos árboles, el número de hojas de T es menor o igual que

$$m(m^{h-1}) = m^h,$$

como queríamos demostrar.

Corolario 5.9.13 *Si un árbol m-ario de altura h tiene l hojas, entonces $h \geq \log_m(l)$.*

Demostración. De la proposición anterior se sigue que $l \leq m^h$ y, en consecuencia, que $\log_m(l) \leq h$.

5.9.1 Árboles de búsqueda binarios

Uno de los problemas en los que se pueden utilizar árboles para encontrar una solución está relacionado con la siguiente pregunta: ¿Cómo debería almacenarse un conjunto de datos para ser fácilmente localizados? Localizar datos es una de las tareas más importantes que se realizan en el ámbito de las ciencias de la computación (buscadores en la red, gestore sde bases de datos...). El primer objetivo es establecer un algoritmo de búsqueda que encuentre los datos eficientemente cuando éstos están totalmente ordenados. Esta tarea se puede realizar utilizando el algoritmo de búsqueda binaria. Pero si además queremos que la propia estructura de datos recuerde información del algoritmo podemos usar un **árbol de búsqueda binaria**, que es un árbol binario en el que vértice tiene dos hijos, uno el derecho y otro el izquierdo, distinguiéndose entre ambos cuál es cuál, y en el que cada vértice lleva asociada una etiqueta, que es uno de los datos.

La etiqueta (el dato) asociado a cada vértice es mayor (en el orden considerado) que las etiquetas de sus descendientes hacia la izquierda, y menor que las etiquetas de sus descendientes hacia la derecha.

El siguiente procedimiento recursivo se utiliza para formar un árbol de búsqueda binaria para una lista de datos. Se comienza con un árbol que contiene un único vértice, la raíz. El primer dato en la lista se asocia a dicho vértice. Para añadir un nuevo dato, primero comparamos dicho dato con las etiquetas de los vértices que ya están situados en el árbol, comenzando por la raíz y moviéndonos para realizar la siguiente comparación hacia la izquierda si el dato es menor que el dato con el que hemos realizado la comparación, y hacia la derecha si es mayor. Cuando el dato es menor que el del vértice con el que acabamos de compararle, y dicho vértice no tiene hijo izquierdo, entonces añadimos al árbol un nuevo vértice como hijo izquierdo de dicho dato. Análogamente, cuando el dato es mayor que el del vértice con el que acabamos de compararle, y dicho vértice no tiene hijo derecho, añadiremos al árbol un nuevo vértice como hijo derecho de dicho dato. Puede ser interesante finalmente hacer una operación de *equilibrar*, esto es, elegri una nueva raíz en el árbol para que todas las ramas tengan una longitud similar.

Ejercicio 124 *Utilizando el orden alfabético, construir un árbol de búsqueda binaria para las palabras Móstoles, Alcorcón, Vicálvaro, Pozuelo, Majadahonda, Pinto, Valdemoro, Madrid, Alcobendas, Boadilla del Monte, Las Rozas.*

La utilidad de los árboles de búsqueda binaria estriba en su poder de

localización de los datos así estructurados dentro del árbol. Para saber si cierto dato es la etiqueta o no de un vértice de un árbol de búsqueda binaria, intentaremos añadir dicho dato al árbol. Obrando de ese modo localizaremos el dato si forma parte de la lista, o añadiremos un nuevo vértice si el dato no forma parte de la lista.

Ejercicio 125 *¿Cuál es el número máximo de comparaciones que hay que realizar en un árbol de búsqueda binaria de altura h para determinar si un dato concreto es la etiqueta de uno de sus vértices? Construir un algoritmo que tenga como entrada un árbol de búsqueda binaria, su raíz, y un dato, y que determine si dicho dato es la etiqueta de uno de los vértices del árbol. Estudiar su complejidad.*

5.9.2 Árboles de decisión

Los árboles con raíz también se utilizan para modelizar problemas en los que una cadena de decisiones conduce a una solución. Un árbol con raíz en el que cada vértice interno corresponde a una decisión, con un subárbol colgando de él por cada una de las posibles alternativas (o salidas diferentes) se denomina **árbol de decisión**. Las posibles maneras de resolver el problema corresponden a los caminos que van desde las hojas hasta la raíz, puesto que cada hoja es una de las difeentes soluciones del problema. Veamos un ejemplo de aplicación de los árboles de decisión.

Ejemplo 5.9.14 *Supongamos que tenemos 8 monedas aparentemente iguales, pero tales que una de ellas pesa un poco menos que las otras 7. ¿Cuál es el menor número de veces que hay que utilizar una balanza para determinar la moneda que pesa menos? Establecer un algoritmo que nos permita localizar dicha moneda.*

Solución: *Hay tres posibilidades para cada pesada que realizamos con la balanza. Que el peso sea el mismo, que las monedas de la bandeja de la izquierda pesen más o que pesen más las de la bandeja derecha. Por consiguiente, el árbol de decisión es ternario (3-ario). Hay al menos 8 hojas en el árbol de decisión, puesto que hay 8 posibles salidas, y cada posible salida debe estar representada por al menos una hoja. El número de pesadas necesario para determinar la moneda que pesa menos es la altura del árbol de decisión. El resultado que recoge el último corolario visto nos permite garantizar que la altura del árbol de decisión es al menos* $\log_3(8)$. *De esta manera, al menos*

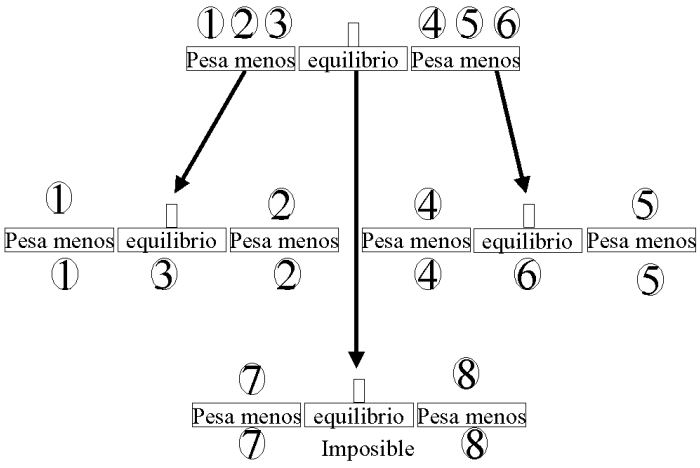


Figura 5.34:

dos pesadas son necesarias. Realmente, con dos pesadas podemos determinar la moneda que pesa menos, según se muestra en la figura 5.34.

Ejercicio 126 *Supongamos que tenemos* 4 *monedas aparentemente iguales, pero tales que una de ellas pesa diferente de las demás (un poco más o un poco menos). ¿Cuál es el menor número de veces que hay que utilizar una balanza para determinar la moneda que pesa menos? Establecer un algoritmo (mediante su árbol de decisión asociado) que nos permita localizar dicha moneda.*

Como señalábamos en el capítulo 2 podemos hablar de complejidad inherente a un problema como el orden de complejidad del mejor algoritmo que resuelve el problema. Los árboles de decisión sirven para calcular esta complejidad. Incidimos en que las cuestines sobre complejidad de problemas, además de ser interesantes en sí mismas, tienen importantes consecuencias sobre la seguridad informática, en el sentido de determinar si un sistema es atacable o no y por quién.

Proposición 5.9.15 *Cualquier algoritmo que resuelva el problema de ordenación de una lista* a_1, ..., a_n *tiene complejidad mayor que* $O(nl(n))$.

Demostración. Sea el árbol de decisión de cualquier algoritmo que resuelve el problema. Es un árbol m-ario con al menos $n!$ hojas pues cualquier posible ordenación de la lista debe ser una hoja de dicho árbol. El número de operaciones que se realizan, partiendo de la raíz, hasta llegar a una solución, esto es, a una hoja, es la altura h del árbol. Por tanto $h \geq log_m(n!)$. Es un ejercicio de Bases de Matemáticas demostrar, para terminar, que:

$$O(log_m(n!)) = O(nl(n)).$$

Sugerencia: demostrar que $n! \leq n^n$ y que $n! \geq n^{n/2}$.

Y podemos construir un algoritmo de ordenación, conocido en inglés como *merge-sort*, que tiene esta complejidad de manera que el problema de ordenar un lista resulta ser de complejidad $O(nl(n))$.

Teorema 5.9.16 *El problema de ordenar una lista de longitud n tiene complejidad $O(nL(n))$.*

La demostración de este teorema, en vistas de la proposición anterior, pasa por construir un algoritmo de ordenación con esta complejidad. La idea es la siguiente:

Idea. Partir sucesivamente la lista L que debe ordenarse en listas de longitud la mitad hasta llegar a listas con un solo elemento y finalmente *mezclarlas* como se señaló en la correspondiente sección del capítulo segundo. Usaremos los algoritmos sobre listas allí construidos así como los algoritmos $primeramitad(L)$ y $segundamitad(L)$ que dada la lista $L = a_1, \ldots, a_n$ construyen

$$primeramitad(L) = a_1, \ldots, a_{n/2}$$

y

$$segundamitad(L) = a_{1+n/2}, \ldots, a_n.$$

Estos dos últimos algoritmos los dejamos como ejercicio para el lector. Vamos a usar para este algoritmo listas donde posiblemente sus elementos son, a su vez, listas. Por ejemplo $L = L_1, L_2$ donde $L_1 = 3, 5, 7$ y $L_2 = 5, 7, 9$.

Algoritmo $merge - sort(L)$

Entrada: $M = M_1, \ldots, M_n$
For $i = 1$ to n $L_{0,i} := M_i$

$i := 1$, $j := 0$
while $longitud(L_j) > 1$
    while $i < longitud(L_j)$
        $L_{j+1,\frac{i+1}{2}} := mezclar(L_{j,i}, L_{j,i+1})$
        $i := i + 2$
    If $longitud(L_j)\ mod\ 2 = 1$ then
        $W := mezclar(L_{j+1,longitud(L_{j+1})}, L_{j,longitud(L_j)})$
        $L_{j+1} := Borrar(L_{j+1}, longitud(L_{j+1}))$
        $L_{j+1} :=$*Añadir*$(L_{j+1}, W, longitud(L_{j+1} + 1))$
    $j := j + 1$, $i := 1$
Salida: $L_{j,1}$

En efecto este algoritmo tiene complejidad $O(nL(n))$. Para llegar a construir listas unitarias a partir de una lista L de longitud n debemos hacer del orden de $log_2(n)$ particiones, luego este es el número de veces que se repite el bucle. Luego debemos mezclar las listas en cada caso, pero como se puede observar fácilmente el algoritmo $mezclar(L_1, L_2)$ necesita del orden de $longitud(L_1) + longitud(L_2)$ operaciones. De esta manera todas las mezclas correspondientes a un momento fijado de la partición necesitarán una cantidad lineal de operaciones. De este modo la complejidad total será $O(nl(n))$. El siguiente ejemplo explica convenientemente el funcionamiento del algoritmo y el número de operaciones que efectúa.

Ejemplo. Tomamos $L = 1, 8, 3, 5, 2, 1, 8, 13, 7$. Apliquemos el algoritmo:

Primeramente se construye la lista $L_0 = 1, 8, 3, 5, 2, 1, 8, 13, 7$ de longitud 9, cuyos elementos se denotan $L_{0,i}$.

Ahora $i := 1$, $j := 0$.

Como $longitud(L_0) = 9 > 1$ entramos en el primer bucle.

Como $1 < 9$ entonces entramos en el segundo bucle.

$L_{1,1} = mezclar(L_{0,1}, L_{0,2}) = 1, 8$ (En este momento la primera entrada de la lista L_1 es a su vez una lista)

$i := 3 < 9$

$L_{1,2} = mezclar(L_{0,3}, L_{0,4}) = 3, 5$

$i := 5 < 9$

$L_{1,3} = mezclar(L_{0,5}, L_{0,6}) = 1, 2$

$i := 7 < 9$

$L_{1,4} = mezclar(L_{0,7}, L_{0,8}) = 8, 13$

$i := 9 = 9$ lo que nos saca del bucle
Como $longitud(L_0) = 9$ es impar entonces
$W := mezclar(L_{1,4}, L_{0,9}) = 7, 8, 13$
$L_1 := (3, 5), (1, 2)$
$L_1 := (3, 5), (1, 2), (7, 8, 13)$, $longitud(L_1) = 3$
$j := 1$, $i := 1$
Como $longitud(L_1) = 3 > 1$ entramos de nuevo en el primer bucle
Como $1 < 3$ entonces entramos en el segundo bucle.
$L_{2,1} = mezclar(L_{1,1}, L_{1,2}) = 1, 2, 3, 5$
$i := 3 = 3$ lo que nos saca del bucle
Como $longitud(L_1) = 3$ es impar entonces
$W := mezclar(L_{2,1}, L_{1,3}) = 1, 2, 3, 5, 7, 8, 13$
vaciamos L_2
$L_2 := (1, 2, 3, 5, 7, 8, 13)$, $j := 1$
$longitud(L_2) = 1$ lo que nos saca del bucle
Salida: 1,2,3,5,7,8,13

Ejercicio 127 *Aplicar el algoritmo anterior a una lista de cinco elementos.*

5.9.3 Árboles generadores

Los árboles tienen propiedades de minimalidad, en el sentido de que son los grafos conexos con el menor número de aristas posibles. Modelan así redes de comunicaciones con un mínimo número de conexiones entre nodos. Esto justifica las siguientes definiciones.

Definición 5.9.17 *Un* **subgrafo generador** *de un grafo simple G es cualquier subgrafo de G que incluye todos los vértices de G (aunque no necesariamente todas sus aristas).*

Definición 5.9.18 *Un* **árbol generador** *(spanning tree) de un grafo simple G es un subgrafo generador de G que es un árbol.*

Observación 5.9.19 *Un grafo simple puede tener más de un árbol generador (ver figura 5.33).*

Los árboles generadores tienen su aplicación en redes de comunicación. Es claro que si queremos construir una red que conecte un conjunto de nodos,

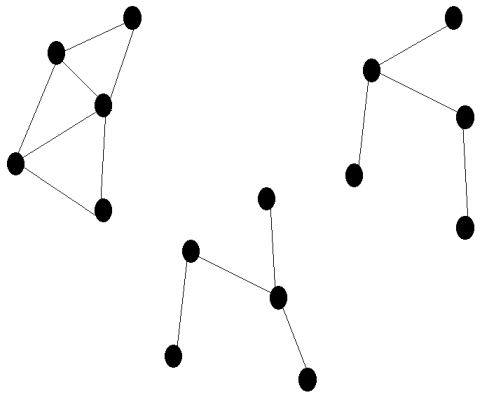


Figura 5.35:

digamos V, la red debe ser conexa para que cada par de nodos estén conectados. De entre los grafos conexos cuyo conjunto de vértices es V, los que *tienen el mínimo número de aristas* son los árboles y son por esta propiedad, minimales. Sin embargo, los árboles son muy poco resistentes a fallos porque, en un árbol, el camino que une dos vértices es único. Así el deterioro de una comunicación directa entre dos nodos desconecta algunos nodos entre sí.

La siguiente proposición caracteriza los grafos simples que tienen un árbol generador.

Proposición 5.9.20 *Un grafo simple G tiene un árbol generador si y solamente si G es conexo.*

Demostración. Si G tiene un subgrafo T que es un árbol y contiene todos sus vértices, entonces para cada par de vértices de G hay un camino en T, y por tanto en G, que los une. Esto muestra que G es conexo.

Si G es conexo, la manera de construir un árbol generador consiste en localizar un circuito en G y quitar una arista. Esta operación preserva la conexión y hace desaparecer un ciclo. Iterando esta operación desaparecerán todos los ciclos.

Hemos dado una manera de construir un árbol generador en el caso de los grafos que los tienen. No obstante, la localización de circuitos en un grafo no es una tarea sencilla. Por ello es conveniente, en lugar de emplear la estrategia anterior, emplear una estrategia de tipo **incremental**: partiendo de un vértice, ir añadiendo aristas junto con sus extremos sin que al añadir una nueva arista aparezca un circuito en el nuevo grafo.

La construcción de un árbol generador mediante un tipo de estrategia incremental puede hacerse básicamente de dos formas distintas:

- Empleando un método de **búsqueda en amplitud**: consiste en recorrer todos los vértices adyacentes a uno dado (elegido al azar) y añadir al árbol en construcción tanto las aristas como los vértices recorridos. Después repetir el proceso recorriendo los vértices adyacentes a los vértices adyacentes al primero. Así, repetir el proceso hasta que no queden vértices por incorporar al árbol.

- Empleando un método de **búsqueda en profundidad**: consiste en partir de un vértice y añadir uno de sus vértices adyacentes junto con su arista. A continuación, a partir de este nuevo vértice, añadir otro adyacente a este último junto con su arista. Así sucesivamente hasta que todos los vértices del grafo original estén incluidos en el árbol generador.

Debido a sus aplicaciones, los árboles generadores de los grafos etiquetados tienen un interés particular. Considérese el siguiente problema:

Problema. Una compañía planea construir una red de comunicaciones que conecte sus cinco centros de computación. Cualquier par de estos centros puede ser unido con un cable telefónico que tiene un coste diferente debido a las características de la conexión y a la topografía del terreno. La pregunta es: ¿cuáles son las conexiones que hay que establecer para garantizar que todos los centros están conectados y que el coste de construcción de la red es mínimo? Si consideramos el grafo etiquetado en el que la etiqueta (o peso) de cada arista es el coste de establecer la conexión entre los dos centros (vértices) que une, el problema se resuelve encontrando un árbol generador tal que la suma de las etiquetas de las aristas de dicho árbol sea mínima. A dicho árbol se le denomina **árbol generador mínimo.**

Definición 5.9.21 *Un* **árbol generador mínimo** *en un grafo etiquetado conexo es un árbol generador que satisface la propiedad de que la suma de las etiquetas de sus aristas es la más pequeña posible.*

Vamos a presentar dos algoritmos para construir árboles generadores mínimos. Ambos algoritmos son de tipo incremental pues tratan de crear un árbol generador añadiendo aristas y vértices. La estrategia opuesta consistiría en, a partir del grafo dado, ir eliminando las aristas de mayor peso teniendo cuidado de que dichas aristas no desconecten el grafo resultante. Ambos algoritmos hacen una elección óptima en cada uno de sus pasos, aunque es importante señalar que optimizar cada etapa de un algoritmo no garantiza que la solución final obtenida sea la óptima. En este caso se puede demostrar que ambos algoritmos producen soluciones globales óptimas.

En ambos algoritmos G es un grafo etiquetado conexo.

Algoritmo de Prim.

Entrada: $G = (V, E)$, $u \in V$
$V(T) := \{u\}$, $E(T) := \emptyset$
while $V(T) \neq V$
Añadir a $E(T)$ una de las aristas de menor peso incidente con un vértice de $V(T)$ (y sólo uno) y añadir su extremo a $V(T)$
Salida: $(V(T), E(T))$

Algoritmo de Kruskal

Entrada: $G = (V, E)$, $u \in V$.
$V(T) := \emptyset$, $E(T) := \emptyset$
while $V(T) \neq V$
Añadir a $E(T)$ la arista de menor peso siempre que no forme un circuito con las aristas de $E(T)$ y añadir sus extremos a $V(T)$.
Salida: $(V(T), E(T))$.

Podemos observar que, mientras que en el algoritmo de Prim el grafo $T = (V(T), E(T))$ es un árbol en todo momento de su construcción, en el de Kruskal sólo podemos asegurar que lo es cuando termina el algoritmo, ya que en cada paso no es necesariamente conexo.

Ejercicio 128 *Utilizar el algoritmo de Prim y el algoritmo de Kruskal para diseñar una red de comunicaciones de mínimo coste que conecte todos los ordenadores representados en el grafo de la figura 5.31.*

5.10 Otros aspectos de la teoría de grafos

Hay muchos otros conceptos, problemas y herramientas interesante relacionados con la teoría de grafos. En esta sección presentaremos someramente dos de ellos: la coloración de grafos y el concepto de grafo plano.

Definición 5.10.1 *Sea $G = (V, E)$ un grafo simple y $C = \{1, 2, ...m\}$ un conjunto de m colores. Una* **coloración** *con m colores del grafo G es una función $f : V \to C$ tal que si $u, v \in V$ y $\{u, v\} \in E$, entonces $f(u) \neq f(v)$.*

La coloración de grafos tiene aplicación, por ejemplo, a la confección de horarios: supongamos que en una universidad se desea realizar un horario para las distintas asignaturas optativas. Se debe tener cuidado de que aquéllas que puedan interesar al mismo tipo de alumnos se programen de manera que no coincidan sus horas. Para conocer el mínimo número de horas necesario para cumplir todos los requisitos podemos construir el siguiente grafo: a cada asignatura se le asigna un vértice, y dos vértices se unen por una arista si interesan al mismo tipo de alumnos. El número mínimo de colores necesarios para colorear los vértices del grafo es el número de horas buscado.

De esta manera se puede asignar un **número cromático** a un grafo, el mínimo número de colores necesarios para su coloración. El siguiente ejercicio caracteriza los grafos con número cromático 2.

Ejercicio 129 *Demostrar que un grafo simple es bipartido si y sólo si admite una coloración con dos colores.*

Ejercicio de ampliación. Construir un algoritmo que permita determinar si un grafo conexo es bipartido, obteniendo el conjunto V de vértices como la unión de dos conjuntos disjuntos V_1 y V_2. Idea: tomar como entrada una matriz de adyacencia A del grafo $G = (V, E)$, donde si n es el orden de la matríz, el conjunto V de vértices es $V = 1, ..., n$ con el orden natural. Partir de la situación inicial $V_1 = 1$ y $V_2 = \emptyset$. Ir añadiendo los vértices adyacentes

a vértices del conjunto V_1 al conjunto V_2 y recíprocamente, comprobando en cada adición si su intersección es vacía.

Otro concepto interesante es el de **grafo plano**:

Definición 5.10.2 *Sea $G = (V, E)$ un grafo. Se dice que G es* **plano** *si admite una representación gráfica en el plano de modo que las aristas únicamente se cortan en los vértices.*

No es difícil comprobar que el grafo completo K_4 es plano, aunque también podemos representar K_4 de manera que sus aristas no sólo se corten en los vértices. Se puede demostrar, aunque no es sencillo y se escapa a los objetivos del curso, la siguiente caracterización de los grafos planos

Teorema 5.10.3 *(de Kuratowski) Un grafo es plano si y solo si no contiene ningún subgrafo que sea isomorfo a una partición de K_5 o de $K_{3,3}$.*

En particular, cualquier grafo que contenga a K_5 o a $K_{3,3}$ como subgrafos no es un grafo plano.

Ejercicio 130 *Determinar si son planos los siguientes grafos: Q_3, $K_{2,2}$ y $K_{2,3}$.*

Obsérvese que saber si un grafo es o no plano tiene aplicaciones a la electrónica para saber si un circuito se puede construir sobre un panel o no, evitando las intersecciones entre las conexiones.

Y hay un famoso teorema que relaciona ambos conceptos:

Teorema 5.10.4 *(de los 4 colores) El número cromático de un grafo plano es menor o igual que 4.*

La historia de este teorema es bastante curiosa, pues a lo largo de la historia han ido apareciendo demostraciones falsas y contraejemplos incorrectos al teorema. Referimos al lector interesado a la sección 7.8 de [R].

5.11 Ejercicios

Ejercicio 131. Dibujar un grafo que represente las distintas aulas y dependencias (hall, etc...) del edificio en que se imparte clase, poniendo una arista entre cada par de dependencias que estén comunicadas.

Ejercicio 132. Dibuja los grafos K_7, W_5, $K_{1,6}$, $K_{4,4}$ y Q_4.

Ejercicio 133. Hallar una matriz de adyacencia y una matriz de incidencia de cada uno de los grafos del ejercicio anterior.

Ejercicio 134. Determinar si existe algún grafo simple con 7 vértices cuyos grados sean $3, 2, 2, 4, 3, 5, 4$. Determinar también si existe algún grafo simple con 5 vértices cuyos grados sean $3, 2, 2, 4, 3$.

Ejercicio 135. Hallar todos los subgrafos de $K_{1,6}$ y de W_3.

Ejercicio 136. Hallar los valores de n para los que los siguientes grafos admiten un circuito euleriano: K_n, C_n y W_n

Ejercicio 137. Dibujar los digrafos con lazos cuyas matrices de adyacencia son las siguientes:

$$\begin{pmatrix} 0 & 2 & 2 & 2 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 3 & 2 & 0 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 2 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 3 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Ejercicio 138. Diseñar un algoritmo cuya entrada sea la matriz de adyacencia de un grafo y que permita determinar el número de aristas de un grafo. Estudiar su complejidad.

Ejercicio 139. Diseñar un algoritmo cuya entrada sea la matriz de adyacencia de un grafo y que permita determinar si el grafo es o no euleriano. Estudiar su complejidad.

Ejercicio 140. Sea una red de 6 ordenadores numerados del 1 al 6 descrita por un grafo cuya matriz de adyacencias es la siguiente:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Sea $p = 0.1$ la probabilidad de que un cable que une dos ordenadores adyacentes se dañe.

i) Cuál es la probabilidad de que se estropeen al menos dos cables.

ii) Cuál es la probabilidad de que un mensaje que parte del ordenador numerado con un 1 no llegue al ordenador numerado con un 2, teniendo en cuenta la información sobre la probabilidad de que se dañe una conexión.

Ejercicio 141. Sea la siguiente matriz la matriz de adyacencias de un grafo simple $G = (V, A)$:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

i) Determinar el número de componentes conexas del grafo.

ii) Determinar si G es un árbol. Determinar si al añadir la arista $\{1, 5\}$ el nuevo grafo $(V, A \cup \{\{1, 5\}\})$ es un árbol.

iii) Determinar los grados de todos los vértices de G.

iv) Determinar si el grafo $(V, A \cup \{\{1, 5\}\})$ es Euleriano.

5.12 Ejercicios resueltos

Ejercicio 109. Las ruedas que se piden se pueden representar como un polígono cerrado de 3, 4, 5 y 6 lados respectivamente con un vértice adicional en el interior y radios desde este vértice a cada uno de los vértices del polígono (ver W_5 en la figura 5.40).

Ejercicio 110. Se puede representar como un cubo tridimensional.

Ejercicio 111. Escribimos $K_3 = (\{1,2,3\}, \{\{1,2\},\{2,3\},\{3,1\}\})$. Si fuera bipartido su conjunto de vértices V se podría escribir como unión disjunta de dos conjuntos no vacíos, digamos V_1 y V_2. Si fuera bipartido podemos suponer, sin perdida de generalidad, que $1 \in V_1$. Necesariamente los vértices 2 y 3, que son adyacentes con 1 deben estar en V_2 lo que da una contradicción ya que son adyacentes.

Sea

$$C_6 = (\{1,2,3,4,5,6\}, \{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,6\},\{6,1\}\}).$$

Tomamos $1 \in V_1$ entonces $2,6 \in V_2$. De este modo $3,1 \in V_1$. Así $V_1 = \{1,3,5\}$ y $V_2 = \{2,4,6\}$ cumplen las condiciones para que C_6 sea bipartido.

Ejercicio 112. Basta tomar 2 rectas paralelas, sobre una de ellas marcar 2 puntos y sobre la otra marcar 5 puntos (respectivamente 3 y 4) y representar todos los segmentos que unen una marca de una de las rectas con una marca de la otra (ver $K_{4,4}$ en la figura 5.42).

Como cada vértice de V_1 ($|V_1| = n$) ha de unirse con cada vértice de V_2 ($|V_2| = m$) el total de aristas es mn.

Ejercicio 113. Los grafos K_4 y W_3 tienen la propiedad de que cualquier vértice está conectado con todos los demás. Tienen el máximo de aristas posibles en un grafo de 4 vértices. Por tanto, son isomorfos.

El grafo $K_{1,3}$ tiene un vértice conectado con los otros 3 que están desconectados entre sí. No puede ser isomorfo a los otros dos grafos.

Ejercicio 114. Basta escribir $V = \{1,2,3\}$ y $V^* = \{1^*, 2^*, 3^*\}$ entonces:

$$K_{3,3} = (V \cup V^*, V \times V^*)$$

$$H = (\{1,2,3,1^*,2^*,3^*\}, \{\{1,2^*\},\{2,2^*\},\{3,1^*\},\{3,2^*\},\{3,3^*\}\})$$

Y de igual manera escribir el conjunto de vértices de los grafos y aristas de la figura, que son respectivamente:

$$G = (\{a,b,c,d,e\}, \{\{a,b\},\{c,b\},\{c,d\},\{d,e\},\{a,e\}\})$$

$$G' = (\{a, b, f\}, \{\{a, b\}, \{f, b\}, \{a, f\}\})$$

De modo que $G \cup G'$ es el grafo representado:

$$G \cup G' = (\{a, b, c, d, e, f\}, \{\{a, b\}, \{c, b\}, \{c, d\}, \{d, e\}, \{a, e\}, \{f, b\}, \{a, f\}\})$$

Ejercicio 115. El cubo Q_3 tiene como vértices, digamos V_3, las palabras de tres bites. El cubo Q_2, V_2, las palabras de dos bites. El cubo Q_1, V_1, las palabras de un bit. El producto $Q_1 \times Q_2$ tiene como vértices el producto $V_1 \times V_2$, esto es, sus vértices son pares de la forma (a_1, a_2a_3) donde $a_1, a_2, a_3 \in \{0, 1\}$. El isomorfismo de grafos se establece de la siguiente manera: $f : V_1 \times V_2 \to V_3$, de modo que $f((a_1, a_2a_3)) = a_1a_2a_3$. Y la relación de adyacencia en el producto se lee: (a_1, a_2a_3) y (b_1, b_2b_3) son adyacentes si y solamente si o bien $a_1 = b_1$ y a_2a_3 es adyacente con b_2b_3 en Q_2 o bien $a_2a_3 = b_2b_3$ y a_1 es adyacente con b_1 en Q_1. Esto es equivalente a decir que existe un único valor $i \in \{1, 2, 3\}$ de modo que $a_i \neq b_i$. Lo que garantiza que f es un isomorfismo de grafos.

Ejercicio 116. Sea $G = (V, E)$ y $G' = (V', E')$ con $|V| = n$, $|E| = m$, $|V'| = n'$ y $|E'| = m'$. Entonces, el conjunto de vértices de $G \times G'$ es $V \times V'$ de modo que, por la regla del cardinal del producto cartesiano, el número de vértices de $G \times G'$ es nn'. Sea (v, v') un vértice del grafo producto. Por definición de producto de grafos se verifica que: $gr((v, v')) = gr(v) + gr(v')$ donde $gr(v)$ es el grado de v en G y $gr(v')$ es el grado de v' en G'. De este modo:

$$\Sigma_{(v,v') \in V \times V'} gr((v, v')) = n' \Sigma_{v \in V} gr(v) + n \Sigma_{v \in V'} gr(v').$$

Se concluye aplicando el Teorema 5.1.14.

Ejercicio 116. Los de la figura 5.9 son isomorfos ya que ambos se pueden ver como la unión de dos ciclos C_5 que comparten 4 vértices.

Los de la figura 5.10 no son isomorfos, ya que el grafo G tiene exactamente dos subgrafos isomorfos a C_4 (uno interior y otro exterior) mientras que el H contiene 3 subgrafos isomorfos a C_4 (los dos del grafo G más el rombo a la izquierda).

Ejercicio 117. K_4 :

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

C_4 :

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

W_3: La rueda está formada por el ciclo C_3 y un vértice conectado con todos los del ciclo. Sea v_1 dicho vértice y v_2, v_3, v_4 los del ciclo. Entonces

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$K_{3,2}$: un cierto conjunto de tres vértices (v_1, v_2 y v_3) está conectado con todos los elementos de un conjunto de dos vértices (v_4 y v_5). Entonces

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Ejercicio 118. Establezcamos para el primer grafo la siguiente ordenación de vértices: *a,b,c,d,e*. Entonces

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

En el segundo grafo, ordenamos los vértices del siguiente modo: *a,b,c,d,e,f*. Entonces

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ejercicio 119. Hay:

1) $n^2 + 1$ asignaciones,

2) un bucle que se repite $n - 1$ veces en el que se hacen

- dos asignaciones,

- elevar una matriz a la potencia j, lo que equivale a multiplicarla por sí misma j veces. Multiplicar dos matrices $n \times n$ supone, por cada fila y columna que se multiplican, n productos y $n - 1$ sumas; es decir, un total de $n^2(2n - 1)$ operaciones. En conclusión, elevar la matriz a la potencia j supone $jn^2(2n - 1)$ operaciones.

- una suma de dos matrices $n \times n$, lo que supone n^2 sumas (de los elementos de ambas matrices).

- la suma $j + 1$.

En total se realizan $3 + n^2 + \sum_{j=1}^{n-1} jn^2(2n - 1) \in O(n^5)$.

3) La verificación de si los n^2 elementos de la matriz B son o no son cero. Total: n^2 operaciones.

El algoritmo tiene por tanto complejidad quíntica.

Ejercicio 120. Una ficha de dominó con dos valores (el 3 y el 5, por ejemplo) se puede ver como una arista entre dos vértices (v_3 y v_5 en el ejemplo). Juntar dicha ficha con otra en una jugada de dominó (por ejemplo, con la que tiene un 5 y un 1) equivale a considerar un camino en el grafo (por ejemplo v_3, v_5, v_1). Con este esquema, una serie de jugadas de dominó no es más que un camino en el grafo completo K_7. Dicho grafo es conexo, tiene 7 vértices y el grado de cada vértice es 6 (excluyendo las fichas dobles, que son lazos y no dan ningún problema), un número par. Podemos garantizar entonces la existencia de al menos un circuito euleriano. En otras palabras, tenemos garantizado que es posible encajar todas las piezas de dominó.

Ejercicio 121. De los cuatro vértices del grafo que representaría a los puentes, tres tienen grado tres y el otro tiene grado cinco. Por tanto, no puede haber ni circuitos ni caminos eulerianos.

Ejercicio 122. Si al grafo H le quito los dos vértices de grado 4 que tiene y sus aristas incidentes, nos quedan tres componentes conexas. Por tanto, el grafo no puede ser hamiltoniano.

Figura 5.36: Ejercicio 124

Ejercicio 123. En el caso de los vértices a y e, tenemos (a partir de los cálculos en los apuntes) que la longitud mínima de a a f es 5 y la de a a d es 6. Por tanto, la longitud mínima de a a g tiene que ser $min\{L(f)+1, L(d)+3\} = 6$.

En el caso de los vértices g y c tenemos que implementar el algoritmo de Dijkstra de nuevo.

Sea $T = \{g\}$. $L(f) = L(g) + 3 = 3$, $L(h) = L(g) + 4 = 4$. El mínimo de L ocurre para f. Por tanto,

$T = \{g, f\}$. $L(e) = L(f) + 1 = 4$, $L(i) = L(f) + 2 = 5$. El mínimo de L ocurre para e y h. Por tanto,

$T = \{g, f, e, h\}$. $L(a) = L(h) + 5 = 9, L(i) = L(h) + 4 = 8 > 5, L(d) = L(e) + 3 = 7$. El mínimo de L ocurre para i. Por tanto,

$T = \{g, f, e, h, i\}$. $L(d) = L(i) + 3 = 8 > 7, L(b) = L(i) + 1 = 6$. El mínimo de L ocurre para b. Por tanto,

$T = \{g, f, e, h, i, b\}$. $L(a) = L(b) + 2 = 8 < 9, L(c) = L(b) + 3 = 9$. El mínimo de L ocurre para d. Por tanto,

$T = \{g, f, e, h, i, b, d\}$. $L(c) = L(d) + 5 = 12 > 9$. El mínimo de L ocurre para a y vale 8. El valor de L para el último vértice que es c es 9.

Conclusión. El camino mínimo entre g y c tiene longitud 9 y recorre los vértices g, f, i, b, c.

Ejercicio 124. Ver la figura 5.36.

Ejercicio 125. El número máximo de comparaciones es el número máximo

de vértices que se pueden recorrer en un camino simple de la raíz a una hoja. Es decir, $h+1$, siendo h la altura del árbol.

Sea h la altura del árbol. Los datos se pueden ordenar mediante una relación que denotamos por $<$.

Entrada: El árbol, su raíz y un dato.
Salida="El dato no es etiqueta de ningún vértice"
lectura=raíz
While -no llegamos a una hoja- and Salida="El dato no es etiqueta de ningún vértice"
If lectura<dato then -nos movemos a la rama derecha del nivel inferior-
If lectura>dato then -nos movemos a la rama izquierda del nivel inferior-
If lectura=dato then Salida="el dato es etiqueta de un vértice"
Salida

En el peor de los casos, el dato no es una etiqueta en el árbol y el árbol binario de altura h tiene $n = 1 + 2 + 2^2 + ... + 2^h = 2^{h+1} - 1$ datos. Para localizar si un dato dado está en el árbol debemos hacer $h+1$ lecturas (el camino más largo que hay en el grafo tiene $h+1$ vértices). Por tanto, el número máximo de lecturas es $\log_2(n+1) \in O(\log n)$. En cada lectura, hacemos una comparación. La complejidad es, por tanto, logarítmica, esto es, $O(\log n)$.

Ejercicio 126. Tomamos dos monedas y las ponemos en sendos platillos de la balanza. Hay dos posibilidades: que pesen lo mismo o que no.

Si no pesan lo mismo, eso significa que una de las dos es la buscada. Tomamos una de las monedas (la que está a la derecha, por ejemplo) y una de las que no hemos pesado (ninguna de las cuáles puede ser la buscada). Las ponemos en sendos platillos. Si pesan lo mismo, eso significa que la moneda que estaba en la primera pesada en el platillo izquierdo es la buscada. Si pesan distinto es la que estaba en el platillo derecho la buscada.

Si en la primera pesada las monedas pesaban lo mismo, entonces tomo una de ellas y una de las que no hemos pesado. Si pesan lo mismo, eso significa que la moneda que no he pesado en ninguna de las dos pesadas es la buscada. Si no pesan lo mismo, eso implica que la moneda que he pesado la segunda vez y no la primera vez es la buscada.

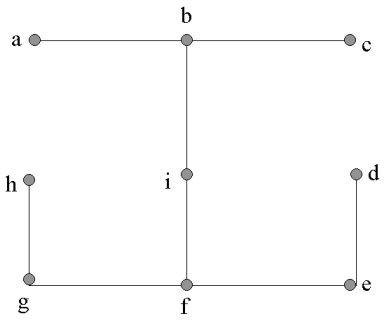


Figura 5.37: Ejercicio 128

En resumen, podemos localizar la moneda en dos pesadas. El árbol de decisión correspondiente es binario y tiene altura 2.

Ejercicio 127. Se deja al lector.

Ejercicio 128. Algoritmo de Prim. Partimos de a. $V(T) = \{a\}$, $E(T) = \emptyset$. Añadimos la arista de a a b y $V(T) = \{a, b\}$. Añadimos i y $V(T) = \{a, b, i\}$. Añadimos f y $V(T) = \{a, b, i, f\}$. Añadimos e y $V(T) = \{a, b, i, f, e\}$. Añadimos c, g y d y $V(T) = \{a, b, i, f, e, c, g, d\}$. Añadimos h y $V(T) = \{a, b, i, f, e, c, g, d, h\}$.

Algoritmo de Kruskal. Colocamos las aristas de longitud 1: $\{b, i\}, \{f, e\}$. Colocamos las aristas de longitud 2: $\{i, f\}, \{a, b\}$. Colocamos las aristas de longitud 3 con cuidado de no formar circuitos: $\{e, d\}, \{b, c\}, \{f, g\}$. Finalmente, colocamos una de las aristas de longitud 4: $\{g, h\}$.

Ejercicio 129. En primer lugar, si el grafo es bipartido, entonces el conjunto de vértices se puede poner como unión disjunta de dos subconjuntos, digamos V_1 y V_2. Podemos entonces colorear con colores distintos cada uno de los subconjuntos V_i $(i = 1, 2)$. Con esta coloración, dado que el grafo es bipartido, no hay nunca dos vértices del mismo color unidos. En sentido opuesto, si el grafo puede ser coloreado sólo con dos colores, podemos dividir

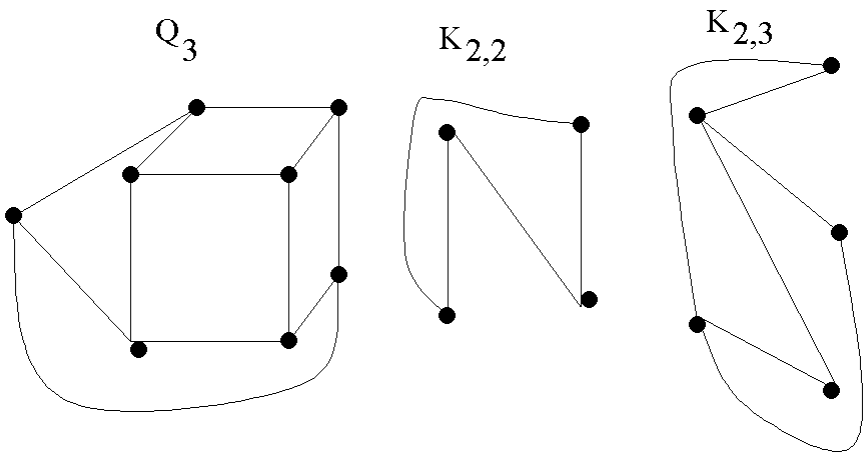


Figura 5.38: Ejercicio 130

los vértices del grafo en dos subconjuntos disjuntos: los vértices de un color y los del otro. Los vértices de un color sólo entán enlazados con los del otro. Es decir, el grafo es bipartido.

Ejercicio 130. Los tres grafos son planos como se puede ver en la figura 5.38.

Ejercicio 131. Se deja al lector.

Ejercicio 132. Ver figuras correspondientes.

Ejercicio 133. Cualquier matriz de adyacencias de K_7 tiene el siguiente aspecto:

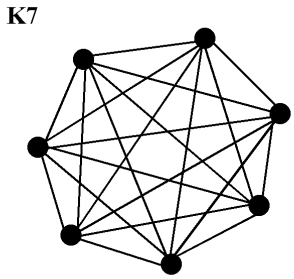


Figura 5.39: Una representación de K_7

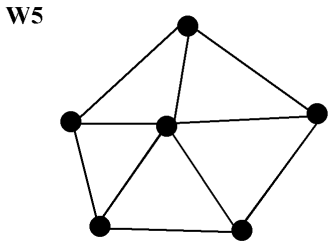


Figura 5.40: Una representación de W_5

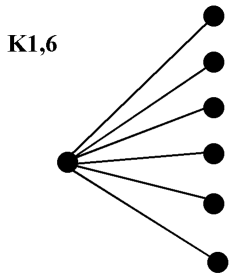


Figura 5.41: Una representación de $K_{1,6}$

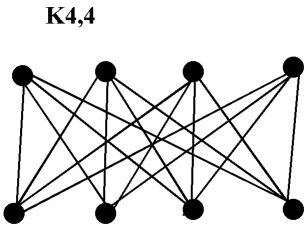


Figura 5.42: Una representación de $K_{4,4}$

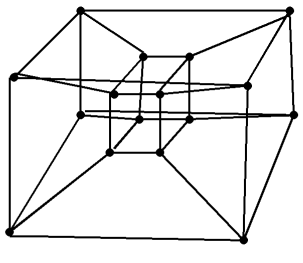


Figura 5.43: Una representación de Q_4

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

El grado de cada uno de los 7 vértices es 6, por tanto la suma de los grados es 42, de modo que K_7 tiene 21 aristas. Si tomo los vértices numerados $\{1, 2, 3, 4, 5, 6, 7\}$ podemos escribir las aristas como:

$$\{\{1,2\},\{1,3\},\{1,4\},\{1,5\},\{1,6\},\{1,7\},$$

$$\{2,3\},\{2,4\},\{2,5\},\{2,6\},\{2,7\},$$

$$\{3,4\},\{3,5\},\{3,6\},\{3,7\},\{4,5\},\{4,6\},\{4,7\},$$

$$\{5,6\},\{5,7\},\{6,7\}\}$$

Por tanto una matriz de incidencias que sigue este orden es:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

El grafo W_5 tiene 6 vértices, digamos $\{1, 2, 3, 4, 5, 6\}$ y las siguientes aristas:

$$\{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,1\},\{1,6\},\{2,6\},\{3,6\},\{4,6\},\{5,6\}\}$$

Siguiendo este orden de vértices tenemos las siguientes matrices de adyacencias e incidencias respectivamente:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

El grafo $K_{1,6}$ tiene 7 vértices, digamos, $\{1, 2, 3, 4, 5, 6, 7\}$ y tiene las aristas:

$$\{\{1,2\},\{1,3\},\{1,4\},\{1,5\},\{1,6\},\{1,7\}\}$$

Así ordenados los vértices y las aristas tenemos su matriz de adyacencias y

su matriz de incidencias:

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

El grafo $K_{4,4}$ tiene 8 vértices que pueden escribirse como $\{1, 2, 3, 4, 1', 2', 3', 4'\}$ y 16 aristas que pueden escribirse como: $A \times A'$ donde $A = \{1, 2, 3, 4\}$ y $A' = \{1', 2', 3', 4'\}$. Si ordenamos los vértices así tenemos la siguiente matriz de adyacencias:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Y si ordenamos las aristas en orden lexicográfico su matriz de incidencias será:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

El cubo Q_4 tiene 16 aristas que corresponden a las palabras de 4 bites:

$$\{0000, 0001, 0010, 0100, 1000, 0011, 0101, 1001,$$

$$1010, 1100, 0110, 0111, 1011, 1101, 1110, 1111\}$$

Cada uno de los vértices tiene grado 4 por lo tanto la suma de todos los grados da 64. Entonces Q_4 tiene 32 aristas. La matriz de incidencias la dejamos al lector. La matriz de adyacencias según este orden es:

$$\begin{pmatrix}
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0
\end{pmatrix}$$

Ejercicio 134. El primero de los casos no es posible pues la suma de los grados es 23 que es un número impar, en contradicción con el hecho de que la suma de los grados de los vértices es 2 veces el número de aristas. El segundo es posible, por ejemplo:

$$(\{a, b, c, d, e\}, \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \{c, e\}, \{d, e\}\})$$

donde $gr(a) = 4$, $gr(b) = 2$, $gr(c) = 3$, $gr(d) = 2$ y $gr(e) = 3$.

Ejercicio 135. El grafo $K_{1,6} = (V, E)$ tiene 7 vértices y 6 aristas:

$$(\{1, 2, 3, 4, 5, 6, 1'\}, \{\{1, 1'\}, \{2, 1'\}, \{3, 1'\}, \{4, 1'\}, \{5, 1'\}, \{6, 1'\}\}).$$

El grafo $W_3 = (V', E')$ tiene 4 vértices y 6 aristas:

$$(\{1,2,3,4\}, \{\{1,2\},\{2,3\},\{3,1\},\{1,4\},\{2,4\},\{3,4\}\}).$$

En ambos casos se tienen como subgrafos los grafos cuyo conjunto de aristas es vacío. Es decir, de la forma $(A, \emptyset)$ donde A es un subconjunto cualquiera de V (respectivamente de V'). Hay 2^7 y 2^4 grafos de este tipo respectivamente.

Si tomamos dos vértices, los nuevos grafos no seleccionados en el párrafo anterior se construyen tomando el par de vértices y la arista que los une. Entonces hay tantos como aristas, esto es, 6 en cada caso.

Si tomamos tres vértices y no son de los del primer párrafo entonces pueden tener 1 ó dos aristas. Si tienen 2 aristas, estas necesariamente deben tener un extremo en común. Para construir un grafo con 3 vértices y una arista se toma una arista, sus extremos y otro vértice cualquiera distinto de sus extremos. En el primer caso hay 6*5=30 de estos grafos, en el segundo 12. Para construir un grafo con dos aristas y 3 vértices, hay que tomar dos aristas que compartan un extremo y sus tres vértices. En el primer caso cada par de aristas comparten el extremo $1'$ por tanto hay $\binom{6}{2}$ de estos grafos. En el segundo caso, como cada vértice tiene grado 3 el número de grafos de este tipo que comparten un vértice fijado es 3. De este modo hay 12 grafos de este tipo.

Si tomamos 4 vértices, en el segundo de los casos estamos tomando todos los vértices, de modo que como conjunto de aristas se puede tomar cualquier subconjunto del conjunto de sus aristas. Así hay $2^6 - 1$ (el vacío ya lo hemos considerado) grafos de este tipo.

Con cuatro vértices en el primero de los casos son del siguiente tipo. Si no contienen al $1'$ ya están considerados pues el conjunto de aristas es vacío. Si contienen al $1'$, entonces podemos tomar una, dos o tres de las aristas que contienen. Así de este tipo hay $7 * \binom{6}{3}$ grafos de este tipo.

Del mismo modo para 5 vértices. Deben contener al $1'$ y se pueden tomar 1,2,3 ó 4 de las aristas que tienen. Hay entonces $15 * \binom{6}{4}$.

Con 6 vértices, deben contener al $1'$ y hay que tomar de 1 a 5 de las aristas que contienen, hay $(2^5 - 1) * 6$.

Con 7 vértices hay que considerar cualquier subconjunto del conjunto de aristas (salvo el vacío que ya está considerado). Entonces hay $2^6 - 1$ grafos de este tipo.

Ejercicio 136. Según el teorema de Euler un grafo admite un circuito

euleriano si todos sus vértices están en la misma componente conexa y los grados de sus vértices son todos números pares. En el caso del grafo completo K_n cada vértice tiene grado $n-1$ entonces será euleriano si y solamente si n impar. En el caso del ciclo C_n el grado de cada vértice es 2 por lo que siempre admite un circuito euleriano. En el caso de W_n hay vértices de grado 3 por lo que no admite un circuito euleriano en ningún caso. Nótese que todos estos grafos son conexos.

Ejercicio 137. Basta escribir tantos puntos como filas (o columnas) tenga la matriz numerándolos según el lugar de su respectiva fila. Entre cada par de puntos i y j se representan tantas aristas (segmentos que unen el punto numerado con la i con el numerado con la j) como indique la matriz de adyacencias A en su término A_{ij}.

Ejercicio 138. Se considera la matriz como una lista con dos subíndices. Se usa el teorema por el que la suma de los grados de los vértices es 2 veces el número de aristas. Como el grado de un vértice es el número de vértices adyacentes con él, entonces el grado del vértice i-ésimo es la suma de los elementos de la fila i-ésima de la matriz de adyacencias del grafo.

Entrada: a_{11}, ..., a_{nn}
$S := 0$
For $i = 1$ to n
    For $j = 1$ to n
        $S := S + a_{ij}$
Salida: $S/2$.

Si el tamaño de la entrada es n, el número de vértices del grafo, la complejidad es cuadrática pues el bucle interior se repite n veces cada vez que i toma un cierto valor.

Ejercicio 139. Primero hay que comprobar que todas las aristas del grafo están en la misma componente conexa (ver práctica de Maple). Una vez hecho esto vamos computando los grados de cada vértice y viendo si es par o no.

Entrada: a_{11}, ..., a_{nn}
$S := 2$
For $i = 1$ to n while $Smod2 = 0$
    $S := 0$

For $j = 1$ to n
$S := S + a_{ij}$
If $Smod2 = 0$ then R :=Es euleriano else R :=No es euleriano.
Salida: R.

Considerando n el tamaño de la entrada y sin considerar el problema de la conexión, el algoritmo es de complejidad cuadrática, pues hay dos bucles anidados que se realizan n veces cada uno y en cada repetición efectúan una cantidad constante de operaciones.

Ejercicio 140. i) Se trata de una red con 6 ordenadores (vértices) y 6 cables (aristas). La probabilidad de que se dañe un cable es 0.1. Es entonces un experimento de Bernoulli con probabilidad de éxito 0.1. La probabilidad de obtener al menos dos éxitos es:

$$\Sigma_{k=2}^{6}\binom{6}{k}(0.1)^k(0.9)^{6-k}.$$

ii) La única manera de que la información no llegue ocurre cuando se dañan a la vez el cable $\{1,2\}$ y al menos uno de los cables del conjunto:

$$\{\{1,5\},\{5,4\},\{4,3\},\{3,2\}\}.$$

Por tanto, así considerados, son sucesos independientes de probabilidades respectivas 0.1 y

$$\Sigma_{k=1}^{4}\binom{4}{k}(0.1)^k(0.9)^{4-k}.$$

El resultado pedido es el producto de ambas probabilidades.

Ejercicio 141. El grafo G tiene dos componentes conexas, la que forman los vértices $\{1,2,3,4\}$ (y sus correspondientes aristas) y la que forman los vértices restantes (y sus correspondientes aristas). De este modo no es conexo. No puede ser un árbol. Si añadimos la arista $\{1,5\}$ el nuevo grafo es conexo, pero contiene un ciclo (formado por los vértices 2, 3 y 4) por lo que sigue sin ser un árbol. Tomando los vértices en el orden que indica la matriz, la sucesión de sus grados no es otra cosa que la suma de los términos de cada fila, así: $gr(1) = 1$, $gr(2) = 3$, $gr(3) = 2$, $gr(4) = 2$, $gr(5) = 1$, $gr(6) = 2$, $gr(7) = 2$, $gr(8) = 1$. Al añadir la arista $\{1,5\}$ el grado de los vértices 1 y 5 aumenta en una unidad. Pero, por ejemplo, el grado del vértice 8 sigue siendo impar, por lo que el grafo no puede ser euleriano.

Capítulo 6

Relaciones y estructuras inducidas

En este capítulo retomaremos, tras el conocimiento adquirido de la teoría de grafos, el concepto de relación, iniciado en el capítulo tercero. Como ya señalamos allí, las relaciones de equivalencia y las relaciones de orden aparecen en ámbitos y ramas muy diferentes de las matemáticas y son fundamentales tanto para su desarrollo como en sus aplicaciones. La definición de número racional, la ordenación usual que se establece en el conjunto de los números naturales (donde cada conjunto tiene mínimo) y en el conjunto de los números reales (donde cada conjunto acotado tiene supremo e ínfimo), la aritmética modular... son conceptos fundamentales en los que está involucrada la noción de relación. También las funciones son un caso especial de relaciones y su estudio es la materia central del Análisis Matemático, de las conexiones de las matemáticas con la física... Por otro lado, las relaciones también tienen su aplicación en la informática, por ejemplo en las bases de datos relacionales y en problemas relativos a clasificación.

Se pretende que al finalizar el capítulo el alumno:

- Conozca las propiedades de una relación de equivalencia y de una relación de orden y sepa discernir si una relación dada es o no de uno de estos tipos.
- Sepa representar una relación mediante un grafo dirigido y analizar propiedades de la relación mediante la matriz de adyacencias del digrafo asociado.

- Pueda calcular la clausura (reflexiva, simétrica) transitiva de una relación.

- Sea capaz de extender un orden parcial para convertirlo en un orden total que contenga las relaciones del orden parcial.

- Conozca las propiedades características de un retículo y un álgebra de Boole y sepa reconocer estas estructuras en algunos ejemplos significativos.

6.1 Compendio de definiciones y propiedades básicas

Aunque son definiciones ya presentadas en secciones anteriores vamos a recordar, en este primer párrafo, los conceptos que necesitamos.

Definición 6.1.1 *Sean A y B dos conjuntos, una* **relación entre A y B** *es un subconjunto R del producto cartesiano $A \times B$. En el caso particular en que A es igual a B hablaremos de una relación en A.*

Se dice que $a \in A$ **está relacionado con** *$b \in B$ (y se denota aRb) si $(a, b) \in R$.*

Definición 6.1.2 *Sean A y B conjuntos y $R \subset A \times B$ una relación entre A y B, se define* **dominio de R** *al conjunto de elementos $a \in A$ tales que existe un elemento $b \in B$ y $(a, b) \in R$.*

Se define el **rango de** *R al conjunto de elementos $b \in B$ para los que existe un elemento $a \in A$ de modo que $(a, b) \in R$.*

Ejemplos 6.1.3 *i) Sea A un conjunto definimos la relación* **identidad de** *A como*

$$Id_A = \{(x, y) \in A \times A : x = y\}.$$

ii) Sean A y B conjuntos y $R \subset A \times B$ una relación entre A y B, denominamos **relación inversa de** *R, y la denotamos por R^{-1}, a la relación*

$$R^{-1} = \{(x, y) \in B \times A : (y, x) \in R\}$$

iii) Si A, B y C son conjuntos y $R \subset A \times B$ y $S \subset B \times C$ son relaciones, denominamos **relación compuesta de R y S**, *y la denotamos por $S \circ R$, a la relación entre A y C definida como sigue*

$$S \circ R = \{(x,z) \in A \times C : \exists y \in B : \ (x,y) \in R \wedge (y,z) \in S\}.$$

Por ejemplo, si $R = \{(a,b),(a,c),(d,a)\}$ y $S = \{(a,d),(c,e)\}$, resulta que $S \circ R = \{(a,e),(d,d)\}$, $R \circ S = \{(a,a)\}$, $R^{-1} = \{(b,a),(c,a),(a,d)\}$, $S^{-1} = \{(d,a),(e,c)\}$ y $R^{-1} \circ S^{-1} = \{(d,d),(e,a)\}$.

Ejercicio 142 *Siendo A, B, C, y D conjuntos, y $R \subset A \times B$, $S \subset B \times C$ y $T \subset C \times D$, comprobar que se satisfacen las siguientes propiedades:*

i) $R \circ Id_A = R$.

ii) $Id_B \circ R = R$.

iii) $Id_A^{-1} = Id_A$.

iv) $R \circ (S \circ T) = (R \circ S) \circ T$.

v) $(S \circ R)^{-1} = R^{-1} \circ S^{-1}$.

Definición 6.1.4 *Sea A un conjunto y $R \subset A \times A$ una relación en A.*

Se dice que R es **reflexiva** *si $(a,a) \in R$ para cada $a \in A$.*

Se dice que R es **simétrica** *si para cada $(a,b) \in R$ se tiene que $(b,a) \in R$.*

Se dice que R es **antisimétrica** *si $(a,b) \in R$ y $(b,a) \in R$ implica que $a = b$.*

Se dice que R es **transitiva** *si para cada $(a,b) \in R$ y $(b,c) \in R$ se tiene que $(a,c) \in R$.*

Ejercicio 143 *Sean A un conjunto y $R \subset A \times A$ una relación en A, comprobar que se satisfacen las siguientes propiedades:*

i) R es reflexiva si y sólo si $Id_A \subset R$.

ii) R es simétrica si y sólo si $R^{-1} = R$.

iii) R es antisimétrica si y sólo si $R \cap R^{-1} \subset Id_A$.

iv) R es transitiva si y sólo si $R \circ R \subset R$.

Ejemplo 6.1.5 *Dados dos conjuntos A y B, una aplicación de A en B es una relación $R \subset A \times B$ tal que: el dominio de R es A y para cada elemento $a \in A$ existe un único elemento $b \in B$ de modo que $(a,b) \in R$.*

6.2 Relaciones de orden

Definición 6.2.1 *Sea A un conjunto y $R \subset A \times A$ una relación en A. La relación R es una* **relación de orden** *si verifica las propiedades reflexiva, antisimétrica y transitiva. Dos elementos $a, b \in A$ se dicen* **comparables** *si o bien (a, b) o bien (b, a) pertenece a R. En el caso particular en que todo par de elementos $a, b \in A$ son comparables se dice que R es una* **relación de orden total**. *Si existen elementos $a, b \in A$ no comparables, entonces diremos que es una* **relación de orden parcial**.

A un par (A, R) formado por un conjunto y una relación de orden parcial le llamaremos **conjunto parcialmente ordenado**. *Si la relación es de orden total diremos que el par (A, R) es un* **conjunto totalmente ordenado**.

Observación 6.2.2 *Utilizaremos los símbolos $\leq$ (ó $\geq$) para denotar las relaciones de orden. Podemos observar que si R es una relación de orden (a la que denotamos por $\leq$) entonces la relación R^{-1} (inversa de R) es también una relación de orden que denotaremos como $\geq$.*

Ejemplos 6.2.3 *i) Como ya señalamos en el capítulo tercero, los números naturales admiten una relación de orden total que viene definida por: dados $a, b \in \mathbb{N}$ se dice que $b \geq a$ si o bien $b = a$ o bien existe un número natural c tal que $b = a + c$.*

ii) También vimos que los números enteros admiten una relación de orden total que es una extensión natural de la relación definida en los números naturales: $a \leq b$ si y solamente si $b - a \in \mathbb{N}^+ \cup \{0\}$.

iii) Los números reales admiten una relación de orden total definida por la existencia de la semirrecta real positiva $\mathbb{R}^+$, (un subconjunto de $\mathbb{R}$ con las propiedades de que para cada $a \in \mathbb{R}$ no nulo o bien a o bien $-a$ pertenece a $\mathbb{R}^+$; si $a, b \in \mathbb{R}^+$ entonces $a + b \in \mathbb{R}^+$, $ab \in \mathbb{R}^+$ y $0 \notin \mathbb{R}^+$) de modo que $a \leq b$ si y solamente si $b - a \in \mathbb{R}^+ \cup \{0\}$.

6.3 Relaciones de equivalencia

Definición 6.3.1 *Sea A un conjunto y $R \subset A \times A$ una relación en A. La relación R es una* **relación de equivalencia** *si verifica las propiedades reflexiva, simétrica y transitiva.*

Ejemplos 6.3.2 *i) Sea el conjunto* $F = \{r/s : r, s \in \mathbb{Z} \quad s \neq 0\}$. *La relación que se define como* $r/s \sim u/v$ *si y solamente si* $rv = su$ *es una relación de equivalencia en* F.

ii) En el conjunto de los números enteros hemos definido la relación de equivalencia módulo p: $a \equiv b \bmod p$ *si y solamente si* $a - b$ *es un múltiplo de* p.

Observación 6.3.3 *Utilizaremos el símbolo* $\sim$ *para representar una relación de equivalencia.*

Definición 6.3.4 *Sea* A *un conjunto y* $\sim$ *una relación de equivalencia en* A. *Para cada elemento* $a \in A$ *se define la* **clase de equivalencia de** a, *y se denota por* $C(a)$ *o por* $\overline{a}$, *como el conjunto de todos aquellos elementos de* A *que se relacionan con* a:

$$\overline{a} = \{b \in A : b \sim a\}.$$

Ejercicio 144 *Con la notación de la definición anterior demostrar que:*

i) para cada $a \in A$ *se tiene que* $\overline{a} \neq \emptyset$;
ii) para cada $a, b \in A$ *de modo que* $a \sim b$ *se tiene que* $\overline{a} = \overline{b}$;
iii) si $\overline{a} \neq \overline{b}$ *entonces* $\overline{a} \cap \overline{b} = \emptyset$;
iv) $A = \cup_{a \in A} \overline{a}$.

Esto permite definir el siguiente concepto.

Definición 6.3.5 *Sean* A *un conjunto y* $\sim$ *una relación de equivalencia en* A. *Se define el* **conjunto cociente** *de* A *por dicha relación de equivalencia como el conjunto de las clases de equivalencia:*

$$A/_{\sim} = \{\overline{a} : a \in A\}.$$

Ejemplos 6.3.6 *i) Como ya vimos, la relación de congruencia módulo* p *es una relación de equivalencia en* $\mathbb{Z}$ *y el conjunto cociente es:*

$$\mathbb{Z}_p = \{\overline{0}, \overline{1}, ..., \overline{p-1}\}.$$

ii) El conjunto cociente del conjunto F *del ejemplo iv) del párrafo anterior por la relación de equivalencia allí definida es el conjunto* $\mathbb{Q}$ *de los números racionales.*

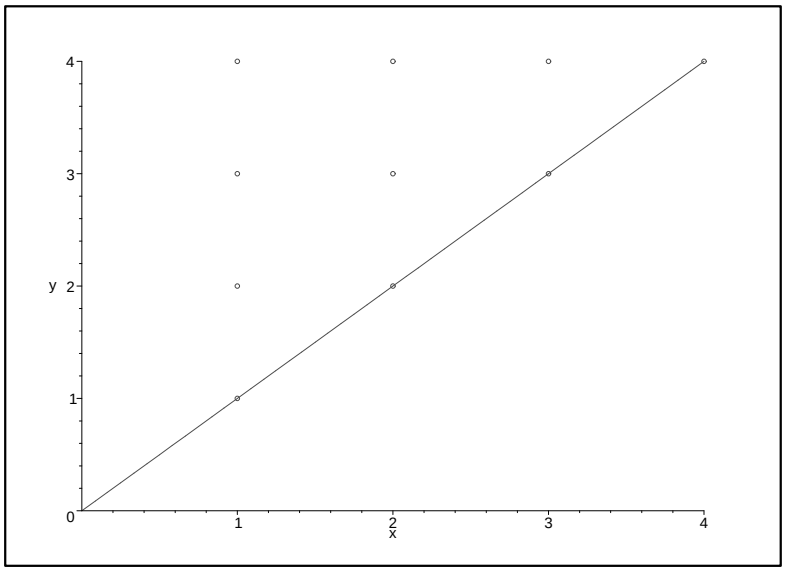


Figura 6.1: Tabla 1

Ejercicio 145 *Determinar si las siguientes relaciones son de orden o de equivalencia. En el caso de que sean de equivalencia, determinar el conjunto cociente.*

i) Sea A_5 el conjunto de palabras de 5 letras hechas con el alfabeto español (29 letras). Sea la relación definida por comenzar por la misma letra, esto es, dados dos elementos $a, b \in A_5$, aRb si y solamente si ambas palabras comienzan por la misma letra. ii) Sea $A = \{a, b, c, d\}$ y

$$R = \{(a,a), (b,b), (c,c), (d,d), (a,b), (a,c), (b,c)\}.$$

iii) Sea $\mathbb{Z} - \{0\}$ y la relación aRb si y solamente si el signo de a es igual al signo de b.

iv) Sea el conjunto $\mathbb{R}$ y la relación aRb si y solamente si $a - b \in \mathbb{Z}$.

Vamos a centrarnos en relaciones **definidas en conjuntos finitos**. Analizaremos distintas formas de representarlas mediante tablas o grafos. De ahora en adelante el conjunto A donde se define la relación es siempre un conjunto finito.

6.4 Representación de relaciones

6.4.1 Mediante tablas

Una relación R en un conjunto A es un subconjunto del producto cartesiano $A \times A$. Tomamos dos ejes coordenados donde en cada uno de ellos se están representando los elementos de A. La relación R se representa mediante una nube de puntos del plano coordenado. Esta nube de puntos permite verificar de forma sencilla algunas de las propiedades de la relación.

La relación es reflexiva si y solamente si toda la diagonal o equivalentemente la relación identidad

$$\Delta = \{(a, a) : a \in A\} = Id_A$$

está contenida en la gráfica.

La relación es simétrica si la gráfica es simétrica con respecto a dicha diagonal.

La relación es antisimétrica si no existen pares de puntos de la gráfica simétricos con respecto a la diagonal y fuera de ella.

Ejemplos 6.4.1 *i) Sea la relación de orden usual sobre el conjunto* $\mathbb{N}_4$*, es decir,*

$$R = \{(1,1),(1,2),(1,3),(1,4),(2,2),(2,3),(2,4),(3,3),(3,4),(4,4)\}.$$

Su tabla es la representada en la Figura 6.1.
Y sobre ella se ve claramente que es reflexiva y antisimétrica.
ii) Sea en el conjunto $\mathbb{N}_4$ *la relación siguiente:*

$$S = \{(1,1),(1,2),(2,1),(1,4),(4,1),(2,3),(3,2),(3,3),(3,4),(4,3)\}.$$

Su tabla es la representada en la Figura 6.2.
Y sobre ella se ve claramente que es simétrica y que no es reflexiva.

6.4.2 Mediante grafos dirigidos

Una relación también puede ser representada por un grafo dirigido que llamaremos **grafo dirigido asociado a la relación**. Este multidigrafo que representa la relación puede tener lazos (elementos de la forma (a, a)) y, como máximo, contiene dos aristas que conectan dos vértices a y b (las aristas (a, b) y (b, a)).

Definición 6.4.2 *Sea A un conjunto finito y $R \subset A \times A$ una relación en A, llamaremos* **grafo dirigido asociado a la relación** *al multidigrafo $G = (A, R)$.*

Ejemplos 6.4.3 *i) En el ejemplo i) anterior el grafo dirigido asociado es:*

$$(\mathbb{N}_4, \{(1,1),(1,2),(1,3),(1,4),(2,2),(2,3),(2,4),(3,3),(3,4),(4,4)\}).$$

ii) En el ejemplo ii) el grafo dirigido asociado es:

$$(\mathbb{N}_4, \{(1,1),(1,2),(2,1),(1,4),(4,1),(2,3),(3,2),(3,3),(3,4),(4,3)\}).$$

Sobre el grafo la propiedad reflexiva consiste en la existencia de un lazo sobre cada vértice.

La propiedad simétrica consiste en que cada par de vértices adyacentes presentan dos aristas (una en cada sentido) que los conectan.

La propiedad antisimétrica consiste en que cada par de vértices adyacentes distintos sólo presentan una arista que los conecta.

La propiedad transitiva consiste en que cada dos vértices unidos por un camino dirigido de longitud dos son necesariamente adyacentes.

En el ejemplo i) se ve que es reflexiva (un lazo sobre cada vértice) y antisimétrica (no hay dos vértices conectados por dos aristas).

En el ejemplo ii) se ve que es simétrica (dos aristas entre cada par de vértices conectados) y no es reflexiva (por ejemplo el vértice 2 no es adyacente con él mismo).

Ejercicio 146 *Sean los siguientes conjuntos y relaciones definidas sobre ellos. Representarlas mediante tablas y grafos dirigidos y verificar qué propiedades satisfacen.*

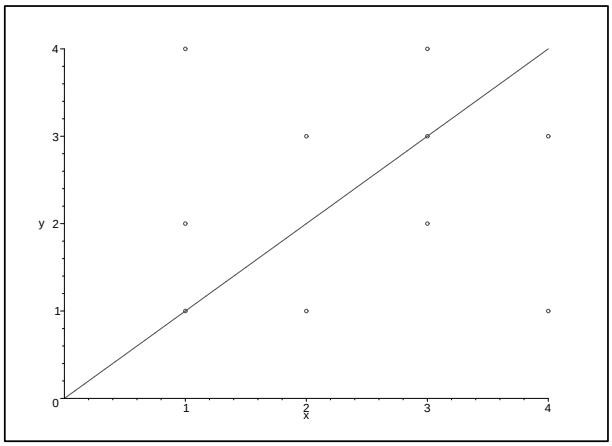


Figura 6.2: Tabla 2

i) Sea el conjunto $\mathbb{N}_5$ *y la relación* R *definida como* aRb *si y solamente si* $a + b$ *es un múltiplo de 7.*

ii) Sea el conjunto $\mathbb{N}_{10}$ *y la relación de divisibilidad, es decir,* aRb *si y solamente si* a *divide a* b*.*

iii) Sea el conjunto $A = \{\emptyset, \mathbb{N}_1, \mathbb{N}_2, \mathbb{N}_3, \mathbb{N}_4, \mathbb{N}_5\}$ *y la relación en* A *dada por* aRb *si y solamente si* $a \subset b$*.*

iv) Sea el conjunto $B = \{a, b, d, c\}$ *y* $R = \{(b, b),\ (c, c),\ (d, d),\ (a, c),\ (c, d),\ (a, d)\}$*.*

6.4.3 Mediante matrices

Podemos representar las relaciones usando las matrices de adyacencias asociadas a los grafos dirigidos que las representan.

Definición 6.4.4 *Dada una relación* R *en un conjunto* A *se define una* **matriz asociada a la relación** M *como una matriz de adyacencias del grafo dirigido asociado a la relación.*

Ejemplo 6.4.5 *Sea en el conjunto* $\mathbb{N}_5$ *la relación de orden* $\leq$ *habitual. La matriz de adyacencias del grafo dirigido, eligiendo el orden en los vértices dado por* $1, 2, 3, 4, 5$ *es una matriz triangular con la parte superior formada por unos y la parte inferior por ceros:*

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Ejercicio 147 *Computar las matrices de adyacencias de los grafos del ejercicio 146.*

Debemos recordar que la matriz de adyacencias M de un grafo dirigido tiene un uno en la entrada (i, j) $(m_{ij} = 1)$ si y solamente si $(v_i, v_j) \in R$, en caso contrario tiene un 0.

De este modo, una relación R sobre un conjunto A es reflexiva si y solamente si la diagonal de la matriz asociada M está formada por unos, esto es $m_{ii} = 1$ para cada $i = 1, ..., |A|$.

La relación es simétrica si y solamente si la matriz es simétrica, porque entonces la equivalencia $(v_i, v_j) \in R$ *si y solamente si* $(v_j, v_i) \in R$ se puede reescribir como $m_{ij} = 1$ *si y solamente si* $m_{ji} = 1$.

La relación es antisimétrica si para cada $i \neq j$ tales que $m_{ij} = 1$ se tiene que $m_{ji} = 0$.

Si la relación es transitiva se verifica el siguiente teorema:

Teorema 6.4.6 *Sea* A *un conjunto finito,* R *una relación en* A *y* M *una matriz de adyacencias del grafo dirigido asociado a la relación. Llamemos* m_{ij} *a las entradas de* M *y* $m_{ij}^{(2)}$ *(el dos es un superíndice, no un cuadrado) a las entradas de* M^2*. Entonces* R *es transitiva si y solamente si* $m_{ij}^{(2)} \neq 0$ *implica* $m_{ij} \neq 0$.

Demostración. Comprobemos primero que si $m_{ij}^{(2)} \neq 0$ implica $m_{ij} \neq 0$ entonces R es transitiva. Para que R sea transitiva debe verificar que cada vez que $(v_i, v_j), (v_j, v_k) \in R$ necesariamente $(v_i, v_k) \in R$.

Si $i = j$ o $j = k$ no hay nada que comprobar. En caso contrario, existe un camino de longitud dos entre v_i y v_k por lo que $m_{ik}^{(2)} \neq 0$. Por hipótesis se tiene $m_{ik} \neq 0$, con lo que $(v_i, v_k) \in R$, como queríamos demostrar.

Por otro lado (la implicación en el otro sentido) si R es transitiva y $m_{ij}^{(2)} \neq 0$ entonces existe un camino de longitud 2 entre v_i y v_j. Por tanto existirá un vértice v_k de modo que $(v_i, v_k) \in R$ y $(v_k, v_j) \in R$. Por la transitividad de R se tiene que $(v_i, v_j) \in R$, es decir $m_{ij} \neq 0$, como queríamos demostrar.

Esto quiere decir que para comprobar la transitividad de una relación basta ver que M^2 no tiene entradas no nulas nuevas con respecto a M.

Ejemplo 6.4.7 *Sea el conjunto* $A = \{a, b, c, d\}$ *y la relación*

$$R = \{(a, a), (b, b), (a, b), (b, a)\}.$$

La relación es transitiva porque $M^2 = 2M$.

Ejercicio 148 *Comprobar si las relaciones del ejercicio 146 son transitivas mediante esta caracterización de la transitividad.*

6.5 Clausuras de una relación

En esta sección presentaremos algoritmos para resolver el problema siguiente:

Problema: dada una relación R en un conjunto A, encontrar la mínima relación (en el sentido del subconjunto más pequeño de $A \times A$) que contiene a R y que verifica una cierta propiedad, como, por ejemplo, la propiedad reflexiva (o la simétrica o la transitiva).

Es un problema que presenta distintas aplicaciones, sobre todo la clausura transitiva, por ejemplo en el campo de las comunicaciones.

Ejemplo 6.5.1 *Tenemos una red de ordenadores numerados del 1 al 5 y tenemos cables de comunicación del 1 al 2, del 2 al 3, del 1 al 4 y del 2 al 5. Aunque el 1 y el 3 no están directamente comunicados entre sí, se pueden comunicar a partir del 2. Nos interesaría conocer los ordenadores que están conectados entre sí.*

De la teoría de grafos podemos extraer una solución sin más que preguntarnos por las componentes conexas del grafo cuyos vértices son los ordenadores y sus aristas las comunicaciones directas.

En el contexto de las relaciones podemos definir una relación en $\mathbb{N}_5$ de modo que $(a, b) \in R$ si y solamente si el ordenador con número a está conectado directamente con el que tiene número b (o $a = b$ o hay un cable directo entre a y b). Esta relación es claramente reflexiva y simétrica por definición. El problema de encontrar todas las parejas de ordenadores conectados (directa o indirectamente), es equivalente al cómputo de la clausura transitiva, es decir, el mínimo conjunto de parejas C que contiene a R y que hace que C sea transitiva.

Definición 6.5.2 *Sea A un conjunto finito y $R \subset A \times A$ una relación en A. Se define la* **clausura transitiva** *(resp. simétrica, resp. reflexiva) al mínimo conjunto $C_t \subset A \times A$ (resp. C_s, resp. C_r) tal que $R \subset C_t$ (resp. $R \subset C_s$, resp. $R \subset C_r$) y la relación que define C_t es transitiva (resp. simétrica, resp. reflexiva).*

El cómputo de la clausura reflexiva es simple. Se trata de unir a R el conjunto $\Delta = \{(a, a) : a \in A\} = Id_A$. Es decir,

$$C_r = R \cup \Delta.$$

La matriz del grafo asociado a C_r es simplemente cambiar los ceros de la diagonal de la matriz M de R por unos.

Ejemplo 6.5.3 *Sea el conjunto $\mathbb{N}_{10}$ y sobre él la relación de menor estricto $<$. Esta relación no es reflexiva porque no se verifica $a < a$. Se verifica que la clausura reflexiva de la relación $<$ es la relación $\leq$.*

Ejercicio 149 *Dadas las relaciones del ejercicio 146 que no son reflexivas computar su clausura reflexiva.*

La clausura simétrica de una relación se computa uniendo a R el conjunto $R^{-1} = \{(b, a) : (a, b) \in R\}$, es decir:

$$C_s = R \cup R^{-1}.$$

La matriz del grafo asociado a C_s se obtiene tomando la matriz del grafo asociado a R, digamos M, sumando esta matriz con su traspuesta M^t, esto es $N = M + M^t$ y construyendo una matriz que tenga un uno donde N tiene un elemento no nulo y un cero donde N tiene un cero.

Ejemplo 6.5.4 *Sea el conjunto* $\mathbb{N}_{10}$ *y sobre él la relación de orden* $\leq$. *La clausura simétrica de dicha relación es* $C_s = \mathbb{N}_{10} \times \mathbb{N}_{10}$. *En efecto, por definición* $C_s \subset \mathbb{N}_{10} \times \mathbb{N}_{10}$. *Para demostrar el otro contenido, sea cualquier elemento* $(a, b) \in \mathbb{N}_{10} \times \mathbb{N}_{10}$. *Si* $a \leq b$ *entonces* $(a, b) \in R \subset C_s$, *en caso contrario* $b < a$ *y por tanto* $(b, a) \in R$ *lo que significa* $(a, b) \in S \subset C_s$, *lo que concluye la demostración.*

Ejercicio 150 *Dadas las relaciones del ejercicio 146 que no son simétricas computar su clausura simétrica.*

Abordamos finalmente el cómputo de la clausura transitiva.

Lema 6.5.5 *Sea* A *un conjunto,* R *una relación sobre* A *y* G *el grafo dirigido asociado a la relación. Sean* a *y* b *dos elementos de* A *tales que existe un camino* $(a, v_1), (v_1, v_2), ...(v_m, b)$ *en* G *que une* a *con* b. *Entonces* (a, b) *pertenece a la clausura transitiva,* C_t, *de* R.

Demostración. Lo demostramos por inducción sobre la longitud del camino.

Si la longitud del camino que une a y b es uno entonces $(a, b) \in R \subset C_t$. Si la longitud es dos, como $(a, v_1), (v_1, b) \in R$ entonces $(a, b) \in C_t$.

Si la longitud es m entonces, por hipótesis de inducción, $(a, v_{m-1}) \in C_t$ y por tanto, como $(v_{m-1}, b) \in R$, se tiene que $(a, b) \in C_t$.

Según este lema, la clasusura transitiva debe contener todos los pares (a, b) de vértices tales que existe un camino que une a con b.

Lema 6.5.6 *Sea* A *un conjunto y* R *una relación en* A *de modo que el grafo dirigido* G *asociado a la relación verifica que si dos vértices* a *y* b *están unidos por un camino entonces* $(a, b) \in R$. *En estas condiciones la relación* R *es transitiva.*

Demostración. Si tenemos que $(a, b), (b, c) \in R$ entonces a y c están unidos por un camino de longitud 2 y por tanto $(a, c) \in R$.

Los dos lemas previos muestran que la clausura transitiva de una relación R en un conjunto A es entonces el conjunto de pares (a, b) de modo que existe un camino en el grafo asociado G que une a con b.

Recordamos ahora una observación que ya comentamos en el capítulo anterior (observación 5.6.13), consecuencia directa del Principio del Palomar:

Observación 6.5.7 *Sea $G = (V, E)$ un grafo dirigido y a, b dos vértices de G. Si existe un camino que une a con b entonces existe un camino de longitud menor o igual que $|V|$ que une a con b. Más aún, si $a \neq b$ y existe un camino que une a con b, se puede encontrar un camino de longitud menor o igual que $|V| - 1$.*

Esta observación permite construir un algoritmo para calcular la clausura transitiva.

Algoritmo para calcular la clausura transitiva

Sea A un conjunto, R una relación en A, $G = (A, R)$ el grafo dirigido asociado a R y M una de sus matrices de adyacencias. El siguiente algoritmo computa la clausura transitiva de la relación R.

Introduzcamos primero el siguiente algoritmo auxiliar:

Matriz de unos y ceros
Entrada: M (una matriz $n \times n$)
For $i = 1$ to n
    For $j = 1$ to n
        If $m_{ij} \neq 0$ then $Matrizdeunosyceros_{ij} := 1$ else
        $Matrizdeunosyceros_{ij} := 0$
Salida: $Matrizdeunosyceros$ (una matriz $n \times n$).

Este algoritmo construye una matriz de unos y ceros a partir de una matriz cualquiera. La matriz construida tiene un uno donde la inicial tenía una entrada no nula y un cero donde tenía un cero. Con esto ya podemos construir el siguiente algoritmo para calcular la clausura transitiva de una relación.

Entrada: M (una matriz $n \times n$)
For $i = 2$ to n
    $M := M + M^i$
Salida: $C_t := Matrizdeunosyceros(M)$.

Observemos que la complejidad de este algoritmo es cuártica. El producto de dos matrices $n \times n$ necesita $2n^3$ operaciones. En efecto al multiplicar una fila por una columna se realizan n productos y n sumas ($2n$ operaciones)

y esto se hace n^2 veces. En nuestro algoritmo se efectúan $n-1$ potencias de M. Suponemos que nuestro sistema de cálculo con matrices hace las potencias como producto de dos matrices (la potencia anterior multiplicada por M). Así tenemos $2n^3(n-1)$ operaciones. La última línea de pseudocódigo de nuestro algoritmo llama al algoritmo $Matrizdeunosyceros$ que tiene complejidad cuadrática. Podemos entonces concluir que el algoritmo es de orden $O(n^4)$.

Ejercicio 151 *Dadas las relaciones del ejercicio 146 que no son transitivas computar su clausura transitiva.*

Ejercicio 152 *Entendiendo una matriz como una lista de números con dos subíndices (el primero la fila y el segundo la columna) escribir un algoritmo que multiplica matrices de tamaño $n \times n$.*

Terminamos esta sección presentando otro algoritmo para computar la clausura transitiva de una relación conocido como **Algoritmo de Warshall** que tiene complejidad cúbica.

Algoritmo de Warshall para el cómputo de la clausura transitiva

Sea como siempre A un conjunto finito ($|A| = n$), R una relación en A, $G = (A, R)$ el grafo dirigido asociado a R y M una de sus matrices de adyacencias. Nuestro objetivo es calcular la matriz asociada a C_t, que sabemos cómo es por el algoritmo anterior. Conviene señalar que M se obtiene mediante una ordenación de los vértices de G, es decir $A = \{v_1, ..., v_n\}$.

Recordemos esta definición.

Definición 6.5.8 *Sea un camino $(x_0, x_1), (x_1, x_2), ..., (x_{n-1}, x_n)$ en un grafo dirigido G. A los vértices $x_1, ..., x_{n-1}$ se les llama* **vértices interiores del camino**.

Ya podemos describir el algoritmo de Warshall. Se trata de construir $n+1$ matrices $n \times n$ que llamaremos W_k ($k \in \{0, 1, ..., n\}$) de modo que $W_0 = M$ y W_n sea la matriz de adyacencias de C_t asociada a la ordenación de A que define M.

Llamamos $w_{ij}^{(k)}$ a las correspondientes entradas de W_k. La definición de W_k es como sigue: $w_{ij}^{(k)} = 1$ si existe un camino que une v_i con v_j de modo que el conjunto de sus vértices interiores esté contenido en el conjunto $\{v_1, ..., v_k\}$.

En el caso $k = 0$ se interpreta como $w_{ij}^{(0)} = 1$ si existe un camino sin vértices interiores que une v_i con v_j, por tanto $W_0 = M$.

En el caso $k = n$ tenemos que $w_{ij}^{(n)} = 1$ si existe un camino que une v_i con v_j (porque el conjunto de posibles vértices interiores es A). Por tanto se tiene que W_n es la matriz de adyacencias buscada.

Ejemplo 6.5.9 *Sea la relación siguiente:*

$$A = \{a, b, c, d, e, f\}$$

$$R = \{(a,a), (a,b), (c,c), (c,d), (c,f), (f,f), (f,e), (f,a)\}.$$

Elegimos el orden para los vértices a, b, c, d, e, f y por tanto la matriz W_1 tiene entradas no nulas (que valen 1) si hay un camino que une los respectivos vértices cuyo único posible vértice interior es a. Entonces:

$$W_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Para analizar la complejidad del algoritmo debemos aportar algun método para construir las matrices W_k. La siguiente observación obvia es la que permite construir W_k a partir de la matriz anterior W_{k-1}.

Observación 6.5.10 *Los caminos sin vértices interiores repetidos que unen los vértices v_i y v_j y cuyos vértices interiores pertenecen al conjunto $\{v_1, ..., v_k\}$ son de dos tipos:*

o bien caminos cuyos vértices interiores son del conjunto $\{v_1, ..., v_{k-1}\}$;

o bien caminos que tienen a v_k como vértice interior, es decir, que se pueden ver como la concatenación de un camino C_1 de v_i a v_k y un camino C_2 de v_k a v_j, donde los vértices interiores de C_1 y C_2 pertenecen al conjunto $\{v_1, ..., v_{k-1}\}$.

Por tanto se tiene que

$$w_{ij}^k = 1 \quad \Leftrightarrow (w_{ij}^{k-1} = 1) \vee (w_{ik}^{k-1} = 1 = w_{kj}^{k-1}).$$

Esto nos permite calcular la complejidad del algoritmo. Para construir W_k a partir de W_{k-1} hay que construir sus n^2 entradas. Para cada entrada, usando la observación anterior, hay que hacer (como máximo) 3 comparaciones y una asignación. Por tanto cada construcción de W_k es un algoritmo de complejidad cuadrática ($4n^2$). Como hay que hacer n construcciones de este tipo, el algoritmo de Warshall tiene **complejidad cúbica**, es decir, del orden $O(n^3)$. Por tanto mejora la complejidad del anterior algoritmo presentado.

Ejercicio 153 *Dadas las relaciones del ejercicio 146 que no son transitivas computar su clausura transitiva mediante el Algoritmo de Warshall.*

6.6 Conjuntos parcialmente ordenados

Como señalamos en párrafos anteriores, un conjunto parcialmente ordenado es un par $(A, \leq)$ formado por un conjunto A y una relación de orden parcial $\leq$. Sea G el grafo dirigido asociado a la relación $\leq$. Como dicha relación $\leq$ es, por definición, reflexiva, antisimétrica y transitiva, se puede simplificar un poco la representación de G. Es lo que se llama diagrama de Hasse de un conjunto parcialmente ordenado (de un *poset* usando las siglas inglesas de *partially ordered set*). Si la relación de orden es total la construcción del diagrama de Hasse tiene también sentido (aunque saldrá siempre una recta vertical).

En general, la expresión *conjunto ordenado* servirá para referirnos a un par $(A, \leq)$ formado por un conjunto y una relación de orden que puede ser parcial o total.

6.6.1 Diagramas de Hasse

En las condiciones del párrafo anterior partimos de G.

Como sabemos que la relación $\leq$ es reflexiva podemos borrar todos los lazos, ya que necesariamente $a \leq a$ para cada $a \in A$.

Como sabemos que la relación $\leq$ es transitiva podemos borrar las aristas que son consecuencia de esta propiedad. Por ejemplo, si $a \leq b$ y $b \leq c$ necesariamente $a \leq c$, entonces los pares (a, b), (b, c) y (a, c) son aristas del grafo y podemos borrar la arista (a, c).

Finalmente, en lugar de dirigir las aristas del grafo, elegimos una disposición vertical de manera que si un elemento a está debajo de otro b y ambos están unidos por un camino ascendente en el grafo simple final entonces $a \leq b$.

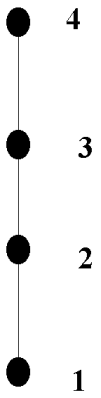


Figura 6.3: Diagrama de Hasse

Ejemplos 6.6.1 *i) Sea $A = \{1, 2, 3, 4\}$ y el orden $\leq$ habitual. Partimos del grafo de la relación. Borramos los lazos y las aristas que son consecuencia de la propiedad transitiva, que son $(1, 3)$, $(1, 4)$ y $(2, 4)$. Por tanto el diagrama de Hasse es una recta vertical y está dibujado en la Figura 6.3.*

ii) Sea el conjunto $B = \{a, b\}$ y definimos la relación $\subseteq$ en el conjunto de las partes de B, denotado $P(B)$. Se puede comprobar que es una relación de orden parcial. El diagrama de Hasse asociado a la relación queda dibujado en la Figura 6.4.

Ejercicio 154 *Dado el conjunto $A = \{a, b, c, d, e\}$ y la relación*

$$R = \{(a, a), (a, c), (c, e), (c, d), (b, d)\}$$

Dibujar el diagrama de Hasse de la relación de orden que viene definida por la clausura reflexiva y transitiva de R.

6.6.2 Elementos característicos de un conjunto ordenado

En esta sección presentaremos algunos elementos interesantes que aparecen en el ámbito de los conjuntos ordenados. Sea $(A, \leq)$ un conjunto ordenado:

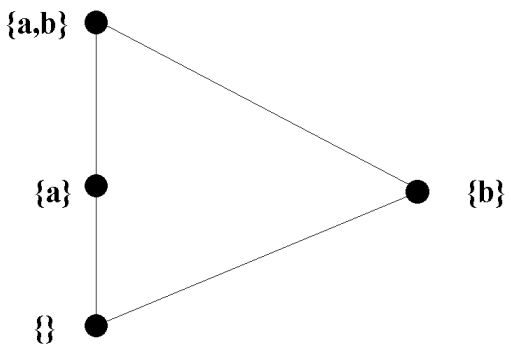


Figura 6.4: Diagrama de Hasse

Definición 6.6.2 *Se dice que un elemento $\alpha \in A$ es un elemento* ***maximal*** *si se verifica que $\forall x \in A\ (\alpha \leq x \Rightarrow \alpha = x)$.*

Se dice que un elemento $\alpha \in A$ es un elemento ***minimal*** *si se verifica que $\forall x \in A\ (x \leq \alpha \Rightarrow x = \alpha)$.*

En los diagramas de Hasse los elementos maximales son aquellos que son extremo superior de un camino ascendente que no se puede prolongar; recíprocamente los elementos minimales son extremos inferiores de caminos descendentes que no se pueden prolongar.

Ejemplo 6.6.3 *En el ejemplo anterior i) el 1 es minimal y el 4 es maximal.*

En el ejemplo anterior ii) el vacío es un elemento minimal y el conjunto B es un elemento maximal.

Observación 6.6.4 *El conjunto de los elementos maximales (resp. minimales) no es necesariamente unitario, como se puede ver en el siguiente ejemplo.*

Ejemplo 6.6.5 *Un conjunto parcialmente ordenado $(\{1,2,3,4,5\}, \leq)$ cuyo diagrama de Hasse es el de la Figura 6.5 tiene dos elementos maximales, 3 y 5, y dos minimales, 1 y 4.*

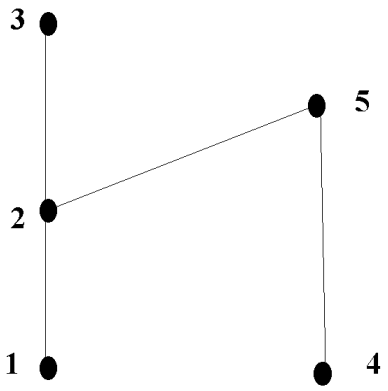


Figura 6.5: Diagrama de Hasse

Definición 6.6.6 *Un elemento $M \in A$ es un* **máximo** *si $a \leq M$ para cada $a \in A$.*

Un elemento $m \in A$ es un **mínimo** *si $m \leq a$ para cada $a \in A$.*

Ejemplo 6.6.7 *En el ejemplo i) 1 es un mínimo y 4 es un máximo.*

En el ejemplo ii) el conjunto vacío es un mínimo (ya que el vacío es un subconjunto de cada conjunto) y el total B es un máximo (ya que cada subconjunto de B está, por definición, contenido en B).

En el diagrama de Hasse el máximo está caracterizado por ser el extremo superior de cada camino ascendente. Del mismo modo, el mínimo está caracterizado por ser el extremo inferior de cada camino descendente.

Observación 6.6.8 *El máximo y el mínimo no necesariamente existen. Si un conjunto parcialmente ordenado tiene máximo (resp. mínimo) entonces es el único elemento maximal (resp. minimal). En efecto, si M es máximo y M' maximal entonces, por ser M máximo, $M' \leq M$. Y por ser M' maximal $M = M'$. De igual manera se razona con el mínimo.*

Ejemplo 6.6.9 *El ejemplo del párrafo anterior, representado en la Figura 6.5, con dos elementos maximales y dos minimales, no tiene máximo ni mí-*

nimo. Por ejemplo, 1 no es un mínimo porque no se verifica que $1 \leq 4$*, de hecho 1 y 4 no son comparables.*

Definición 6.6.10 *Sea* $(A, \leq)$ *un conjunto parcialmente ordenado y* $B \subset A$*.*

Se dice que $a \in A$ *es una* **cota superior** *de* B *si* $b \leq a$ *para cada* $b \in B$*.*

Se dice que $a \in A$ *es una* **cota inferior** *de* B *si* $a \leq b$ *para cada* $b \in B$*.*

Se dice que $B \subset A$ *está* **acotado superiormente** *si tiene cotas superiores, y que está* **acotado inferiormente** *si tiene cotas inferiores. Si tiene cotas superiores e inferiores hablaremos de conjunto* **acotado**.

Ejemplos 6.6.11 *i) Si consideramos el conjunto de los números naturales con la relación de orden usual, el* $1 \in \mathbb{N}$ *es el mínimo de* $\mathbb{N}$ *(*$min(\mathbb{N}) = 1$*). Por otra parte, el conjunto* $\mathbb{N}$ *no está acotado superiormente. Denotando por* $\mathbb{P}$ *al conjunto de los números pares, el 1 y el 2 son cotas inferiores de* $\mathbb{P}$*. El conjunto* $\mathbb{P}$ *tampoco está acotado superiormente, y se verifica que* $min(\mathbb{P}) = 2$*.*

ii) Si consideramos el conjunto de los números reales con relación de orden usual, y el conjunto

$$(0, \infty) = \{x \in \mathbb{R} : 0 < x\}$$

se verifica que 0 es una cota inferior de $(0, \infty)$*, (-1 y -2 también son cotas inferiores de* $(0, \infty)$*), pero el conjunto* $(0, \infty)$ *no tiene mínimo.*

Definición 6.6.12 *Sea* $(A, \leq)$ *un conjunto parcialmente ordenado y* $B \subset A$ *un conjunto acotado superiormente. Se define el* **supremo de** B*, si existe, como el único elemento* $sup(B) \in A$ *que satisface las dos propiedades siguientes:*

i) $sup(B)$ *es una cota superior de* B*,*

ii) $sup(B) \leq c$ *para cualquier* $c \in A$ *cota superior de* B*.*

Esto es, el supremo de B *es el mínimo del conjunto de las cotas superiores de* B*.*

Definición 6.6.13 *Sea* $(A, \leq)$ *un conjunto parcialmente ordenado y* $B \subset A$ *un conjunto acotado inferiormente. Se define el* **ínfimo de** B*, si existe, como el único elemento* $inf(B) \in A$ *que satisface las dos propiedades siguientes:*

i) $inf(B)$ *es una cota inferior de* B*,*

ii) $inf(B) \geq c$ *para cualquier* $c \in A$ *cota inferior de* B*.*

Esto es, el ínfimo de B *es el máximo del conjunto de las cotas inferiores de* B*.*

Ejercicio 155 *Sea el conjunto* $A = \{a, b, c, d, e, f\}$ *y la relación de orden dada por la clausura reflexiva y transitiva de:*

$$R = \{(f, a), (a, c), (c, e), (c, d), (f, b), (b, d)\}.$$

Calcular los elementos minimales, los maximales, el máximo y el mínimo si existen.

Tomando $B = \{a, b, c, d\} \subset A$ *calcular el conjunto de cotas superiores de* B*, el de cotas inferiores, y el supremo y el ínfimo, si existen.*

El conjunto de los números reales con el orden usual $(\mathbb{R}, \leq)$ es un conjunto totalmente ordenado de singular importancia sobre el que se edifica el Análisis Matemático. Por otra parte, en el capítulo 3, hemos usado las propiedades de ordenación en el conjunto de los números enteros $(\mathbb{Z}, \leq)$ para el teorema de la división entera. Ambos conjuntos son totalmente ordenados y no finitos. En la siguiente sección vamos a presentar algunas aplicaciones de los conjuntos finitos totalmente ordenados y un algoritmo para, dado un conjunto parcialmente ordenado, construir una relación de orden total que contenga la relación de orden parcial.

6.6.3 Inmersión de un orden parcial en un orden total

Supongamos que tenemos un conjunto de tareas que realizar de modo que hay una serie de prioridades, estableciéndose una relación de orden parcial sobre el conjunto de tareas. Por ejemplo un conjunto de asignaturas (que no pueden cursarse simultáneamente) que aprobar para conseguir una licenciatura con una serie de asignaturas que han de cursarse después de aprobarse unas asignaturas previas. Nos interesará establecer un orden total para realizarse las tareas, por ejemplo, una manera de cursar las asignaturas sabiendo en qué orden han de realizarse. Veamos un ejemplo muy simple:

Ejemplo 6.6.14 *Sea un conjunto de tres asignaturas Matemáticas I, Matemáticas II y Física para realizar en 3 años. Se tiene que cursar primero Matemáticas I y luego Matemáticas II. Nada se dice sobre cuando cursar la Física. Formalmente tenemos un conjunto (usando las siglas)*

$$C = \{MI, MII, F\}$$

con una relación de orden parcial

$$MI \leq MI, MI \leq MII, MII \leq MII, F \leq F.$$

Las distintas maneras de cursar estas asignaturas corresponden a los distintos órdenes totales de que se puede dotar a C respetando el orden parcial.

Estos órdenes totales son:

$$MI \leq MII \leq F \qquad MI \leq F \leq MII \qquad F \leq MI \leq MII.$$

Esto es, de las posibles 6 permutaciones del conjunto de las tres asignaturas, elegimos una de las tres en que MI precede a MII.

Por tanto abordamos este problema general, que se puede aplicar a la planificación de tareas:

Problema. Dado un conjunto parcialmente ordenado (A, R) encontrar una relación de orden total T de modo que $R \subset T$.

Como se ha visto en el ejemplo la relación T no es única. La aplicación a la planificación de tareas consiste en diseñar una forma de realizar todas las tareas teniendo en cuenta las prioridades que se conozcan, esto es, tareas que deben preceder necesariamente a otras.

Lema 6.6.15 *Cada conjunto finito parcialmente ordenado (A, R) tiene un elemento minimal.*

Demostración. Sea un elemento $x_1 \in A$. Si x_1 es minimal el lema queda demostrado, en caso contrario existe $x_2 \leq x_1$ con $x_2 \neq x_1$. Si x_2 es minimal queda demostrado el lema, si no existe $x_3 \leq x_2$ y $x_3 \neq x_2$. Como el conjunto es finito necesariamente este proceso debe terminar aportando un elemento minimal.

Este lema permite construir un algoritmo para la construcción del orden total $T \supseteq R$.

Partimos de A y elegimos un elemento minimal a_1, que existe por el lema, aunque no es necesariamente único (elección de una tarea para realizarla la primera de entre las tareas más prioritarias). Ahora el conjunto $A - \{a_1\}$ es un conjunto parcialmente ordenado con el orden que induce R sobre él. Repetimos el proceso para elegir un elemento minimal a_2 (que por el lema anterior necesariamente existe) y así sucesivamente hasta agotar los elementos de A.

El orden total T es

$$a_1 \leq a_2 \leq a_3 \leq ... \leq a_n$$

Ejemplo 6.6.16 *Sea el conjunto* $A = \{1, 2, 3, 4, 5\}$ *y la relación de orden parcial en* A,

$$R = \{(1,1), (2,2), (3,3), (4,4), (5,5), (1,2), (1,3), (2,3), (1,4), (1,5), (4,5)\}.$$

Elegimos un elemento minimal, que en el primer caso es necesariamente $x_1 = 1$. *Después en el conjunto* $A - \{1\}$ *tenemos la relación de orden inducida*

$$R_1 = \{(2,2), (3,3), (4,4), (5,5), (2,3), (4,5)\}.$$

Elegimos un elemento minimal entre los dos posibles candidatos que son el 2 ó el 4. Tomemos $x_2 = 2$. *En el conjunto* $A - \{1, 2\}$ *tenemos un orden inducido:*

$$R_1 = \{(3,3), (4,4), (5,5), (4,5)\}.$$

Tomamos $x_3 = 3$ *(también podíamos elegir el 4). Continuando el proceso* $x_4 = 4$ *y* $x_5 = 5$. *Por tanto el orden total es:*

$$1 \leq 2 \leq 3 \leq 4 \leq 5.$$

Obsérvese que podríamos haber elegido de otra manera los x_i para obtener un orden total distinto, también compatible con el orden parcial con el que comenzamos.

Ejercicio 156 *Se tiene que montar una cadena de montaje de una pieza según el siguiente esquema de tareas.*

Tareas: Ensamblar, atornillar, pintar, empaquetar, limpiar, registrar.

Prioridades: Ensamblar antes de atornillar, atornillar antes de pintar, pintar antes de empaquetar, limpiar antes de pintar.

Diseñar un posible planificación de las tareas.

6.7 Retículos y Álgebras de Boole

Sea $(A, \leq)$ un conjunto ordenado.

Definición 6.7.1 *Diremos que* $(A, \leq)$ *es un retículo si para cualesquiera elementos* $x, y \in A$ *el conjunto* $\{x, y\} \subset A$ *tiene supremo e ínfimo.*

Observación 6.7.2 *Si $(A, \leq)$ es un conjunto ordenado, utilizaremos la siguiente notación para designar al supremo y al ínfimo del conjunto {x,y}:*

$$sup\{x, y\} = x \vee y.$$

$$inf\{x, y\} = x \wedge y.$$

Obsérvese que, al emplear esta notación, de forma implícita estamos definiendo dos operaciones en el conjunto A. El supremo $sup\{x, y\}$ es el resultado de operar x con y (en ese orden) mediante la operación $\vee$. Del mismo modo el ínfimo $inf\{x, y\}$ es el resultado de operar x con y (en ese orden) mediante la operación $\wedge$.

Ejemplos 6.7.3 *i) Si X es un conjunto, y $\mathcal{P}(X)$ es el conjunto formado por todos los subconjuntos de X, el par $(\mathcal{P}(X), \subseteq)$ es un retículo, de tal forma que si $A, B \in X$, $A \vee B = A \cup B$ y $A \wedge B = A \cap B$.*

ii) Cualquier conjunto totalmente ordenado $(A, \leq)$ es un retículo.

iii) Si consideramos el conjunto de los números naturales con la relación de orden dada por

$$a \leq b \Leftrightarrow \exists p \in \mathbb{N} \ : \ a \cdot p = b,$$

resulta que $(\mathbb{N}, \leq)$ es un retículo, y para cualesquiera $a, b \in \mathbb{N}$

$$a \vee b = mcm(a, b), \ a \wedge b = mcd(a, b),$$

donde mcm es el mínimo común múltiplo y mcd es el máximo común divisor.

iv) Si R es la relación de orden del ejemplo anterior, y consideramos el conjunto D_{12} formado por los números naturales que son divisores de 12, el par $(D_{12}, R \cap (D_{12} \times D_{12}))$ es un retículo.

Sea $(A, \leq)$ un retículo. Para cualesquiera $x, y \in A$ se satisfacen las siguientes propiedades (la primera y la segunda se siguen directamente de las definiciones de $x \vee y = sup\{x, y\}$ y $x \wedge y = inf\{x, y\}$):

Propiedad 1. $x \leq (x \vee y)$, $y \leq (x \vee y)$, $(x \wedge y) \leq x$ y $(x \wedge y) \leq y$.

Propiedad 2. Si $x \leq z$ y $y \leq z$, entonces $(x \vee y) \leq z$. Análogamente, si $z \leq x$ y $z \leq y$, entonces $z \leq (x \wedge y)$.

Propiedad 3. (Conmutatividad) $(x \vee y) = (y \vee x)$ y $(x \wedge y) = (y \wedge x)$.

Demostración. Demostraremos la primera de las dos propiedades (la otra se demuestra de manera análoga): $(x \vee y) = sup\{x, y\} = sup\{y, x\} = (y \vee x)$.

Propiedad 4. (Asociatividad) $(x \vee y) \vee z = x \vee (y \vee z)$ y $(x \wedge y) \wedge z = x \wedge (y \wedge z)$

Demostración. Demostraremos la primera de las dos propiedades (la otra se demuestra de manera análoga). Comprobaremos, en primer lugar, que $(x \vee y) \vee z \leq x \vee (y \vee z)$ y dejaremos como ejercicio el verificar la otra desigualdad, es decir, que $x \vee (y \vee z) \leq (x \vee y) \vee z)$. Por la propiedad 1 $x \leq x \vee (y \vee z)$ y $y \leq (y \vee z) \leq x \vee (y \vee z)$. Por la propiedad 2 $(x \vee y) \leq x \vee (y \vee z)$. Por otra parte $z \leq (y \vee z) \leq x \vee (y \vee z)$ y, por lo tanto, $(x \vee y) \vee z \leq x \vee (y \vee z)$.

Propiedad 5. (Propiedad de Absorción) $(x \vee y) \wedge x = x$ y $(x \wedge y) \vee x = x$.

Demostración. Demostraremos la primera de las igualdades (la otra se demuestra análogamente). Por la propiedad 1 y la refexividad de la relación $x \leq (x \vee y)$ y $x \leq x$. Por tanto $x \leq (x \vee y) \wedge x$. Por otra parte, para cualquier z, $(z \wedge x) \leq x$, luego $((x \vee y) \wedge x) \leq x$. De las dos desigualdades y de la propiedad antisimétrica se sigue que $(x \vee y) \wedge x = x$.

En resumen, hemos probado que si tenemos un conjunto ordenado $(A, \leq)$ con estructura de retículo, podemos considerar sobre el conjunto A dos operaciones, que denotamos por $\vee$ y $\wedge$, que satisfacen las propiedades conmutativa, asociativa y de absorción. También, como ahora veremos, es posible hacer la construcción al revés, i.e., partiendo de un conjunto A con dos operaciones, que denotamos por $\vee$ y $\wedge$, que satisfagan las propiedades conmutativa, asociativa y de absorción, es posible definir sobre el conjunto A una relación de orden $\leq$ de tal manera que $(A, \leq)$ es un retículo. Por ello, y de forma completamente equivalente, se puede adoptar como definición de retículo, cualquiera de las siguientes:

- Un conjunto ordenado $(A, \leq)$ tal que para cualesquiera $x, y \in A$ el conjunto $\{x, y\}$ tiene supremo e ínfimo.

- Un conjunto A con dos operaciones, denotadas por $\vee$ y $\wedge$, que satisfacen las propiedades conmutativa, asociativa y de absorción.

Es importante señalar que en la literatura sobre el tema es usual encontrar la propiedad de *idempotencia*, a saber, que para cualquier $x \in A$ se verifiquen las condiciones $x \vee x = x$ y $x \wedge x = x$, como un axioma exigible a la estructura de retículo. Sin embargo, si $\vee$ y $\wedge$ satisfacen las propiedades conmutativa, asociativa y de absorción, necesariamente satisfacen también la condición

de idempotencia. En efecto, dado cualquier elemento $x \in A$, tomamos la expresión

$$E := x \vee (x \wedge (x \vee x)).$$

Llamando $(y = x \vee x)$ y aplicando la propiedad de absorción, obtenemos que E es igual a x. Por otra parte, si en E aplicamos la propiedad de absorción a $x \wedge (x \vee x)$ obtenemos que dicha expresión es igual a $x \vee x$. En consecuencia, $x \vee x = x$. (Análogamente se prueba la otra propiedad de idempotencia).

Veamos que efectivamente es posible definir una relación de orden *reticular* a partir de dos operaciones que satisfagan las propiedades conmutativa, asociativa y de absorción.

En lo que sigue consideraremos un conjunto A y dos operaciones $\vee$ y $\wedge$ definidas sobre A que satisfacen las propiedades conmutativa, asociativa y de absorción. Definimos la siguiente relación en A:

$$x \leq y \;\Leftrightarrow\; x \vee y = y.$$

Antes de demostrar que $(A, \leq)$ es un retículo, podemos caracterizar la relación $\leq$ de otro modo:

Lema 6.7.4 *Para cada $x, y \in A$ se tiene que*

$$x \leq y \Leftrightarrow x \wedge y = x.$$

Demostración. Será suficiente con demostrar que para cualesquiera $x, y \in A$, $x \vee y = y$ equivale a $x \wedge y = x$. Supongamos que $x \vee y = y$. En ese caso $x \wedge (x \vee y) = x \wedge y$, es decir, $x = x \wedge y$. Recíprocamente, si $x \wedge y = x$, tendremos que $(x \wedge y) \vee y = x \vee y$, es decir, $y = x \vee y$.

Comprobemos ahora que $(A, \leq)$ es un retículo:

1. La relación $\leq$ satisface la propiedad **reflexiva**. En efecto, para cada $x \in A$ se tiene que $x \vee x = x$, es decir, $\forall x \in A,\ x \leq x$.

2. La relación $\leq$ satisface la propiedad **simétrica**. Si $x \leq y$, y $y \leq x$, tendremos simultáneamente que $x \vee y = y$, y que $y \vee x = x$, y, por la propiedad conmutativa de $\vee$, obtenemos que $y = x$.

3. La relación $\leq$ satisface la propiedad **transitiva**. Supongamos que $x \leq y$, e $y \leq z$. En ese caso $x \vee y = y$, y $y \vee z = z$. Ahora bien, a partir de esas dos

igualdades obtenemos que: $x \vee z = x \vee (y \vee z) = (x \vee y) \vee z) = y \vee z = z$, es decir, $x \leq z$.

Las propiedades 1, 2 y 3 muestran que $(A, \leq)$ es un conjunto ordenado.

4. Para cada par de elementos $x, y \in A$ se tiene que $sup\{x, y\} = x \vee y$. (La propiedad relacionada con la existencia del ínfimo del conjunto $\{x, y\}$ se demuestra análogamente). Puesto que, por la propiedad de absorción, $x \wedge (x \vee y) = x$, y $y \wedge (x \vee y) = y$, tendremos que $x \leq (x \vee y)$, y $y \leq (x \vee y)$, es decir, $x \vee y$ es una cota superior de $\{x, y\}$. Veamos que es la más pequeña: Supongamos que $z \in A$ es tal que $x \leq z$ y $y \leq z$. En ese caso, $x \vee z = z$, y $y \vee z = z$. Luego $(x \vee z) \vee (y \vee z) = (z \vee z) = z$. Pero, por las propiedades conmutativa y asociativa de $\vee$ (y por la propiedad de idempotencia, consecuencia de ambas), tenemos que $(x \vee z) \vee (y \vee z) = (x \vee y) \vee (z \vee z) = (x \vee y) \vee z$. En definitiva, $(x \vee y) \vee z = z$, de donde $(x \vee y) \leq z$.

Por consiguiente, debido a la equivalencia demostrada, podremos utilizar como definición de retículo cualquiera de las siguientes:

- Un conjunto ordenado $(A, \leq)$ tal que para cualesquiera $x, y \in A$ el conjunto $\{x, y\}$ tiene supremo e ínfimo. En este caso podremos definir las operaciones $\vee$ y $\wedge$ en el retículo a partir de la relación $\leq$.

- Un conjunto A con dos operaciones, denotadas por $\vee$ y $\wedge$, que satisfacen las propiedades conmutativa, asociativa y de absorción. En este caso, podremos definir la relación de orden $\leq$ a partir de $\vee$ o equivalentemente a partir de $\wedge$.

Definición 6.7.5 *Se dice que un retículo $(A, \leq)$ es distributivo si para cualesquiera $x, y, z \in A$ se verifica que $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.*

Si un retículo $(A, \leq)$ es distributivo, entonces también se satisface la distributividad de la operación $\wedge$ respecto de la operación $\vee$, a saber, que $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ pues, si $x, y, z \in A$, por la propiedad distributiva, $(x \wedge y) \vee (x \wedge z) = ((x \wedge y) \vee x) \wedge ((x \wedge y) \vee z) =$ (por las propiedades de absorción y distributiva) $= x \wedge (z \vee (x \wedge y)) =$ (propiedad distributiva) $= x \wedge ((z \vee x) \wedge (z \vee y)) =$ (propiedad asociativa) $= (x \wedge (z \vee x)) \wedge (z \vee y) =$ (propiedad de absorción) $= x \wedge (z \vee y) = x \wedge (y \vee z)$.

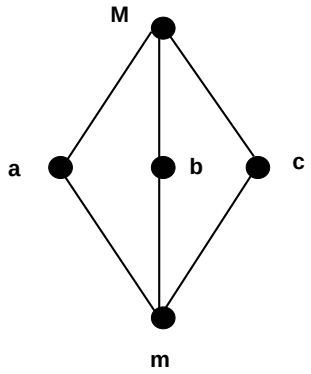


Figura 6.6: Un retículo no distributivo

Observación 6.7.6 *No todos los retículos son distributivos. Por ejemplo, el conjunto ordenado representado en el diagrama de Hasse de 6.6 es un retículo que no satisface la propiedad distributiva, pues*

$$(a \wedge b) \vee c = m \vee c = c$$

y, sin embargo,

$$(a \vee b) \wedge (b \vee c) = M \wedge M = M.$$

Observación 6.7.7 *Si un retículo $(A, \leq)$ tiene máximo, denotaremos a dicho elemento por 1, y si tiene mínimo, por 0.*

Definición 6.7.8 *Sea $(A, \leq)$ un retículo con máximo 1 y mínimo 0 y $x \in A$; se dice que $x' \in A$ es **complementario** de x si $x \vee x' = 1$ y $x \wedge x' = 0$.*

El retículo de la figura 6.6 nos muestra que un elemento puede tener varios complementarios, pues tanto b como c son complementarios de a, ya que $a \vee b = a \vee c = 1$ y $a \wedge b = a \wedge c = 0$. Sin embargo, si un retículo es distributivo, entonces cada elemento puede tener, a lo sumo, un complementario, como muestra la siguiente proposición:

Proposición 6.7.9 *Sea $(A, \leq)$ un retículo distributivo con máximo 1 y mínimo 0, y sea $x \in A$. Si x' y x'' son complementarios de x, necesariamente $x' = x''$.*

Demostración. Por hipótesis $x \vee x' = 1$, $x \wedge x' = 0$, $x \vee x'' = 1$ y $x \wedge x'' = 0$. Veamos que $x' \leq x''$: $x' = x' \wedge 1 = x' \wedge (x \vee x'') = (x' \wedge x) \vee (x' \wedge x'') = 0 \vee (x' \wedge x'') = (x' \wedge x'')$. Es decir, $x' \leq x''$. Análogamente se demuestra que $x'' \leq x'$, con lo que $x' = x''$.

Definición 6.7.10 *Un retículo $(A, \leq)$ es* **complementario** *si tiene máximo y mínimo y todo elemento $x \in A$ tiene al menos un complementario. Denominaremos* **álgebra de Boole** *a cada retículo distributivo y complementario.*

Ejemplos 6.7.11 *i) Si X es un conjunto, y $\mathcal{P}(X)$ es el conjunto formado por todos los subconjuntos de X, el par $(\mathcal{P}(X), \subseteq)$ es un álgebra de Boole, en la que $X = 1$, $\varnothing = 0$. Recordamos que si $A, B \in X$, $A \vee B = A \cup B$ y $A \wedge B = A \cap B$. En lo sucesivo diremos que este álgebra de Boole es el álgebra de Boole de las partes del conjunto X.*

ii) El conjunto $(\mathbb{B}, \leq)$ donde $\mathbb{B} = \{0, 1\}$ y $\leq = \{(0,0), (0,1), (1,1)\}$ es un álgebra de Boole.

iii) Siendo n cualquier número natural, el conjunto $(\mathbb{B}^n, \leq)$ formado por todas las n-tuplas (o sucesiones finitas) de "ceros" y "unos", junto con la relación de orden $(x_1, ..., x_n) \leq (y_1, ..., y_n)$ si y sólo si en todas las posiciones en las que $(x_1, ..., x_n)$ tiene un 1 se verifica que $(y_1, ..., y_n)$ también lo tiene, es un álgebra de Boole. Nótese que el elemento máximo es $(1, 1, ..., 1)$ y el mínimo $(0, 0, ..., 0)$.

iv) El conjunto de los números naturales con la relación de orden dada por la relación de divisibilidad

$$a \leq b \Leftrightarrow \exists p \in \mathbb{N} \ : \ ap = b$$

constituye un retículo distributivo. Este retículo no es un álgebra de Boole, puesto que no tiene máximo. Recuérdese que en este retículo, siendo $a, b \in \mathbb{N}$

$$a \vee b = mcm\{a, b\}$$

$$a \wedge b = mcd\{a, b\}.$$

v) Si R es la relación de orden del ejemplo anterior, y consideramos el conjunto D_{12} formado por los números naturales que son divisores de 12, el par $(D_{12}, R \cap (D_{12} \times D_{12}))$ no es un álgebra de Boole . Se puede comprobar que este retículo es distributivo, que tiene máximo y mínimo, pero que, por ejemplo, 2 no tiene complementario.

vi) Si R es la relación de orden considerada en el ejemplo iv) y consideramos el conjunto D_6 formado por los números naturales que son divisores de 6, el par $(D_6, R \cap (D_6 \times D_6))$ es un álgebra de Boole. Se propone como ejercicio dibujar su diagrama de Hasse.

vii) El conjunto formado por aquellos subconjuntos A de $\mathbb{N}$ tales que o bien A es finito, o bien su complementario $\mathbb{N} - A$ es finito, constituye un álgebra de Boole respecto del orden usual dado por la relación $\subseteq$. En este álgebra de Boole, como se señaló antes, $A \vee B = A \cup B$ y $A \wedge B = A \cap B$.

Teorema de Representación

En el contexto usual de las estructuras algebraicas aparece una noción de isomorfismo de álgebras de Boole.

Definición 6.7.12 *Dos álgebras de Boole $(A, \vee, \wedge)$ y $(A', \vee', \wedge')$ son* **isomorfas** *si existe $f : A \to A'$ biyectiva de manera que para cada $a, b \in A$ se tiene que $f(a \vee b) = f(a) \vee' f(b)$, $f(a \wedge b) = f(a) \wedge' f(b)$ y la imagen por f del complementario de a es el complementario de $f(a)$.*

Dos álgebras de Boole isomorfas son, en cuanto a su estructura de álgebra de Boole, idénticas, aunque tengan elementos diferentes. El siguiente teorema nos permite caracterizar todas las álgebras de Boole finitas:

Teorema 6.7.13 *(Teorema de representación) Cualquier álgebra de Boole* finita *es isomorfa al álgebra de Boole $\mathcal{P}(X)$ de las partes de un conjunto finito X.*

La idea de la demostración es la siguiente. Sea A un álgebra de Boole finita. Consideramos el conjunto A_{atom} formado por los *átomos de A*, es decir, aquellos elementos $a \in A$ caracterizados por la siguiente propiedad:

$$x \leq a \Rightarrow \ (x = 0 \vee x = a).$$

Se puede demostrar que A es isomorfa al álgebra $\mathcal{P}(A_{atom})$ de las partes del conjunto A_{atom}.

Como consecuencias inmediatas de este teorema podemos establecer las siguientes:

- Todas las álgebras de Boole finitas tienen 2^n elementos, siendo n el número de sus átomos.
- Dos álgebras de Boole finitas con el mismo número de átomos, o de elementos, son isomorfas.

Para finalizar, vamos a establecer un principio que permitirá obviar la demostración de muchas propiedades.

Principio de dualidad

Casi todas las propiedades relacionadas con los retículos y las álgebras de Boole tienen una versión para la operación $\vee$ y otra para la operación $\wedge$. Esto no es por casualidad sino consecuencia de la propia estructura de álgebra de Boole. De hecho, está relacionado con un resultado que hemos visto en la primera sección de este capítulo, a saber, que si $(A, \leq)$ es un conjunto ordenado, entonces $(A, \geq)$ también es un conjunto ordenado, donde

$$\geq = (\leq)^{-1}.$$

Esto es, la relación de inversa de una relación de orden dada, también es una relación de orden. Además, no resulta difícil comprobar que, si en $(A, \leq)$ tenemos que $x \vee y = M$ y $x \wedge y = m$, en el conjunto ordenado con la relación de orden inversa, $(A, \geq)$ tendremos que $x \vee y = m$ y $x \wedge y = M$. Por ello, a partir de cualquier enunciado válido en un retículo o en un álgebra de Boole se puede obtener un nuevo enunciado válido sin más que sustituir en el mismo cada uno de los siguientes símbolos por el que aparece a su derecha:

$$\begin{array}{ccc} \leq & \text{por} & \geq \\ \vee & \text{por} & \wedge \\ \wedge & \text{por} & \vee \\ 1 & \text{por} & 0 \\ 0 & \text{por} & 1 \end{array}$$

La expresión obtenida al realizar las sustituciones anteriores es denominada expresión dual de la expresión dada.

Así por ejemplo, si $(A, \leq)$ es un retículo o un álgebra de Boole, y $x, y \in A$ tendremos que

- De la propiedad $x \vee x = x$, obtendríamos directamente, por dualidad, que $x \wedge x = x$.
- De la propiedad $x \leq (x \vee y)$ obtendríamos directamente, por dualidad, que $x \geq (x \wedge y)$ o, equivalentemente, $(x \wedge y) \leq x$.
- De la propiedad $(x \vee y) = (y \vee x)$ obtendríamos directamente, por dualidad, que $(x \wedge y) = (y \wedge x)$.
- De la propiedad $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ obtendríamos directamente, por dualidad, que $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.
- De la propiedad $x \vee 1 = 1$ obtendríamos directamente, por dualidad, que $x \wedge 0 = 0$.

6.8 Ejercicios

Ejercicio 157. Estudiar si las siguientes relaciones son relaciones de orden o de equivalencia:

i) Sea el conjunto $\mathbb{N}_7$ y la relación aRb si $a + b < 9$.

ii) Sea el conjunto R de las rectas del plano y la relación de paralelismo, esto es, rRr' si r y r' son paralelas.

iii) Sea el conjunto $A = \{a, b, c, d, 1, 2, 3\}$. En él definimos una relación por la cual dados $m, n \in A$ mRn si o bien m y n son ambos números o bien m y n son ambos letras.

iv) Sea el conjunto $B = \{1, 2, 3, 4, 7, 8, 90\}$. En él definimos la siguiente relación S, aSb si $3a > b$.

Ejercicio 158. Representar mediante digrafos y sus matrices las relaciones del ejercicio anterior definidas sobre conjuntos finitos.

Ejercicio 159. En las relaciones del ejercicio 157 definidas sobre conjuntos finitos que no sean reflexivas (respectivamente simétricas, respectivamente transitivas) calcular su clausura reflexiva (respectivamente simétrica, respectivamente transitiva).

Ejercicio 160. Tomar la clausura reflexiva y transitiva de la relación definida en el conjunto

$$\{1, 2, 3, 4, 5, 6\}$$

como:

$$R = \{(1,2),(1,4),(3,6),(5,4)\}.$$

Verificar que es una relación de orden. Representar su diagrama de Hasse. Calcular elementos maximales, minimales y máximos y mínimos si los hubiera.

Ejercicio 161. Describir el conjunto cociente de 157.ii).

Ejercicio 162. Sea el conjunto $A = \{\alpha, \beta, \gamma, \omega\}$. Tomamos la siguiente relación en él:

$$R = \{(\alpha,\alpha),(\alpha,\beta),(\alpha,\gamma),(\gamma,\omega)\}.$$

Construir la clausura reflexiva y transitiva de R y verificar que es una relación de orden. Representar su diagrama de Hasse y construir una relación de orden total que contenga a R.

Ejercicio 163. Demostrar que en cualquier álgebra de Boole se verifican las leyes de *De Morgan*: $\forall x, y \in A$, $(x \wedge y)' = x' \vee y'$ y $(x \vee y)' = x' \wedge y'$, donde x' denota el complementario de x.

6.9 Ejercicios resueltos

Ejercicio 142.

i) Sea $(a,b) \in R$. Como $(b,b) \in Id_A$ entonces $(a,b) \in R \circ Id_A$. Esto prueba que $R \subset R \circ Id_A$. Si $(a,b) \in R \circ Id_A$ entonces por definición existe $c \in A$ tal que $(a,c) \in R$ y $(b,c) \in Id_A$. Por definición de la identidad $b = c$ y se tiene el otro contenido.

ii) Se razona como en i).

iii) Como cada elemento de Id_A es de la forma (a,a), con $a \in A$ entonces la relación inversa no tiene ningún elemento nuevo.

iv) Sea $(a,d) \in R \circ (S \circ T)$. Existe entonces $b \in B$ tal que $(a,b) \in R$ y $(b,d) \in S \circ T$. De la segunda relación de pertenenecia se deduce que existe $c \in C$ tal que $(b,c) \in S$ y $(c,d) \in T$. Por tanto, de la existencia de b, se deduce que $(a,c) \in R \circ S$. Se concluye la inclusión de izquieda a derecha del hecho que $(c,d) \in T$. El otro contenido se razona análogamente.

v) Observemos que $(c,a) \in (S \circ R)^{-1}$ si y solamente si existe $b \in B$ y $(a,b) \in R$ y $(b,c) \in T$. Esto es equivalente a decir $(b,a) \in R^{-1}$ y $(c,b) \in T^{-1}$. De esta manera $(c,a) \in R^{-1} \circ T^{-1}$.

Ejercicio 143. i) La relación R es reflexiva si y solamente si $(a,a) \in R$ para cada $a \in A$. Por tanto i) es exactamente la definición de la propiedad reflexiva.

ii) R es simétrica si y solamente si $(b,a) \in R$ para cada $(a,b) \in R$.

iii) R es antisimétrica si y solamente si $(a,b),(b,a) \in R$ implica $a = b$, equivalentemente $(a,b) \in R \cap R^{-1}$ implica $(a,b) \in Id_A$.

iv) R es transitiva si y solamente si $(a,b),(b,c) \in R$ implica $(a,c) \in R$. Pero $(a,b),(b,c) \in R$ es equivalente a $(a,c) \in R \circ R$.

Ejercicio 144.

i) Como R es una relación de equivalencia entonces es reflexiva, de modo que para cada $a \in A$ se tiene que aRa y de este modo $a \in \overline{a}$.

ii) Sea $c \in \overline{a}$ entonces cRa y aRb. Por la propiedad transitiva cRb y así $c \in \overline{b}$. De igual manera se prueba el otro contenido.

iii) Supongamos que $\overline{a} \cap \overline{b} \neq \emptyset$ entonces existe $c \in A$ de modo que aRc y bRc lo que garantiza por la propiedad simétrica y por la propiedad transitiva que aRb y por ii) que $\overline{a} = \overline{b}$ lo que da una contradicción.

iv) Sea $c \in A$ entonces $c \in \overline{c}$ por lo que se tiene la igualdad requerida.

Ejercicio 145.

i) Es una relación de equivalencia. En efecto, es reflexiva porque toda palabra p empieza por la misma letra que la propia palabra p. Es simétrica porque si pRq entonces la primera letra de la palabra p es igual que la primera letra de la palabra q, por tanto qRp, pues sus primeras letras son iguales. De igual modo si la primera letra de la palabra p es igual que la primera letra de la palabra q y la primera letra de la palabra q es igual que la primera letra de la palabra r, entonces la primera letra de la palabra p es igual que la primera letra de la palabra r, por lo que se tiene la propiedad transitiva. No es una relación de orden porque hay palabras distintas como *alto* y *alba* que verifican $altoRalba$ y $albaRalto$, luego no es antisimétrica.

El conjunto cociente es el conjunto de letras pues se puede identificar cada clase de equivalencia con la inicial de cualquiera de las palabras que la forman.

ii) Es reflexiva pues $\{(a,a),(b,b),(c,c),(d,d)\} \subset R$. No es simétrica pues $(a,b) \in R$ pero $(b,a) \notin R$. Es antisimétrica pues nunca se tienen $m \neq l$ tal que $(m,l) \in R$ y $(l,m) \in R$. Es transitiva pues el único caso relevante es $(a,b),(b,c) \in R$ y efectivamente $(a,c) \in R$. Por tanto es una relación de

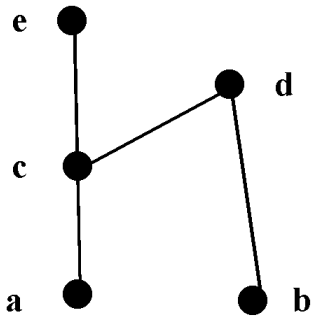


Figura 6.7: Ejercicio 145.ii). Diagrama de Hasse

orden (parcial) pero no de equivalencia. Su diagrama de Hasse está en la Figura 6.7.

iii) Como todo número entero no nulo n tiene el mismo signo que n se tiene la propiedad reflexiva. Si a, b enteros no nulos verifican que $signo(a) = signo(b)$ entonces $signo(b) = signo(a)$ con lo que la relación es simétrica. También es transitiva pues se tiene que si $signo(a) = signo(b)$ y $signo(b) = signo(c)$ entonces $signo(a) = signo(c)$. Por tanto es una relación de equi/-valencia y no es de orden porque no es antisimétrica (hay números distintos con el mismo signo).

El conjunto cociente tiene dos elementos, digamos $signo+$ y $signo-$, que son, respectivamente, la clase de equivalencia de los positivos y la de los negativos.

iv) Es reflexiva pues para cada número real a se tiene que $a-a = 0 \in \mathbb{Z}$. Es simétrica ya que si $a-b \in \mathbb{Z}$ entonces $b-a \in \mathbb{Z}$. Es transitiva pues si $a-b \in \mathbb{Z}$ y $b-c \in \mathbb{Z}$ entonces la suma es un número entero: $a-b+b-c = a-c \in \mathbb{Z}$. Por tanto es una relación de equivalencia que no es de orden.

Se tiene que $\overline{a} = \{a+k : k \in \mathbb{Z}\}$.

Se puede tomar en cada clase de equivalencia un representante entre 0 y uno (podíamos decir, su parte decimal) por lo que el conjunto cociente se puede identificar con el intervalo $[0, 1)$.

Ejercicio 146.

i) $\mathbb{N}_5 = \{1, 2, 3, 4, 5\}$. La relación es el siguiente conjunto:

$$R = \{(2, 5), (5, 2), (3, 4), (4, 3)\}.$$

La tabla asociada consiste en representar los puntos de R en un plano coordenado.

El digrafo asociado es:

$$(\mathbb{N}_5, \{(2, 5), (5, 2), (3, 4), (4, 3)\}).$$

ii) En este conjunto la relación es:

$$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (1, 10),$$

$$(2, 2), (2, 4), (2, 6), (2, 8), (2, 10), (3, 3), (3, 6), (3, 9), (4, 4), (4, 8), (5, 5), (5, 10)$$

$$(6, 6), (7, 7), (8, 8), (9, 9), (10, 10)\}$$

La tabla asociada consiste en representar los puntos de R en un plano coordenado.

El digrafo asociado es:

$$(\mathbb{N}_{10}, R).$$

iii) La relación es:

$$R = \{(\emptyset, \emptyset), (\emptyset, \mathbb{N}_1), (\emptyset, \mathbb{N}_2), (\emptyset, \mathbb{N}_3), (\emptyset, \mathbb{N}_4), (\emptyset, \mathbb{N}_5), (\mathbb{N}_1, \mathbb{N}_1),$$

$$(\mathbb{N}_1, \mathbb{N}_2), (\mathbb{N}_1, \mathbb{N}_3), (\mathbb{N}_1, \mathbb{N}_4), (\mathbb{N}_1, \mathbb{N}_5),$$

$$(\mathbb{N}_2, \mathbb{N}_2), (\mathbb{N}_2, \mathbb{N}_3), (\mathbb{N}_2, \mathbb{N}_4), (\mathbb{N}_2, \mathbb{N}_5), (\mathbb{N}_3, \mathbb{N}_3), (\mathbb{N}_3, \mathbb{N}_4),$$

$$(\mathbb{N}_3, \mathbb{N}_5), (\mathbb{N}_4, \mathbb{N}_4), (\mathbb{N}_4, \mathbb{N}_5), (\mathbb{N}_5, \mathbb{N}_5)\}$$

La tabla asociada consiste en representar los puntos de R en un plano coordenado, tomando como coordenadas los subíndices (el 0 para el conjunto vacío). El digrafo asociado es:

$$(A, R).$$

iv) La tabla asociada consiste en representar los puntos de R en un plano coordenado, tomando por ejemplo $a = 1$, $b = 2$, $d = 3$ y $c = 4$. El digrafo asociado es:

$$(B, R).$$

Ejercicio 147.

i) Tomamos $\mathbb{N}_5$ ordenado de la forma usual.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

ii)Tomamos $\mathbb{N}_{10}$ ordenado de la forma usual.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

iii) Tomamos A ordenado de la forma en que está escrito.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

iv) Tomamos B ordenado de forma alfabética.

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Ejercicio 148. Podemos hacerlo automáticamente mediante un procedimiento de Maple (ver la Práctica 5 de Maple en la página web referida) que

dada una matriz M calcula M^2. Con M^2 construye una matriz M^{2*} que tiene un 1 en la entrada de subíndice ij si la correspondiente entrada en M^2 es no nula y un 0 en caso contrario. Y toma la diferencia $M^{2*} - M$ para comprobar si tiene entradas positivas (es decir si hay alguna entrada no nula nueva).

En cualquier caso i) no es transitiva porque $2R5$ y $5R2$ pero $2 + 2 = 4$ no es un múltiplo de 7. La relación ii) es transitiva porque si a divide a b entonces $b = aK$ con K un número entero. Del mismo modo si b divide a c entonces $c = bK'$ con K' un número entero. Por tanto $c = aKK'$, de modo que a divide a c. La relación iii) es transitiva pues lo es la inclusión. La relación iv) es transitiva.

Ejercicio 149. La relación i) no es reflexiva, para convertirla en reflexiva hay que unir a R el conjunto

$$\{(2,2),(3,3),(4,4),(5,5)\}.$$

Las relaciones ii) y iii) son reflexivas.

La relación iv) no es reflexiva, hay que añadirle el elemento (a,a)

Ejercicio 150. La relación i) es simétrica.

La relación ii) no es simétrica, hay que unirle el conjunto:

$$\{(2,1),(3,1),(4,1),(5,1),(6,1),(7,1),(8,1),(9,1),(10,1),$$

$$(4,2),(6,2),(8,2),(10,2),(6,3),(9,3),(8,4),(10,5)\}.$$

La relación iii) no es simétrica, su clausura simétrica es $A \times A$.

La relación iv) no es simétrica, hay que unirle los elementos (c,a), (d,c) y (a,d).

Ejercicio 151. Para calcular la clausura transitiva de la única relación que no es transitiva, la i), hay que unir los elementos

$$(2,2),(5,5),(3,3),(4,4).$$

Ejercicio 152. Consideramos una matriz como una lista con dos subíndices. Si la entrada es:

$$a_{11}, a_{12}, ..., a_{1n}, ..., b_{n1}, ..., b_{nn}.$$

La correspondiente salida se define como una matriz de entradas c_{ij} donde:

$$c_{ij} = a_{i1}b_{1j} + ... + a_{in}b_{nj}.$$

De este modo el algoritmo debe ser:

Entrada: $a_{11}, a_{12}, ..., a_{1n}, ..., b_{n1}, ..., b_{nn}$.
For $i = 1$ to n
    For $j = 1$ to n
        $c_{ij} := 0$
        For $l = 1$ to n
        $c_{ij} := c_{ij} + a_{il}b_{lj}$
Salida: $c_{11}, ..., c_{nn}$.

Ejercicio 153. La única relación no transitiva es la de i). Las matriz W_0 es exactamente la matriz del grafo dirigido, esto es,

$$W_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

La matriz W_1 tiene un uno donde W_0 tenga un uno y además se verifica que $w^1_{ij} = 1$ si $w^0_{i1} = 1 = w^0_{1j}$. Como en nuestro caso la primera fila de W_0 es toda nula entonces

$$W_1 = W_0.$$

La matriz W_2 tiene un uno donde W_1 tenga un uno y además se verifica que $w^2_{ij} = 1$ si $w^1_{i2} = 1 = w^1_{2j}$. Esta condición se verifica sólo en $w^1_{52} = 1 = w^1_{25}$, por tanto se tiene $w^2_{55} = 1$.

$$W_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

La matriz W_3 tiene un uno donde W_2 tenga un uno y además se verifica que $w^3_{ij} = 1$ si $w^2_{i3} = 1 = w^2_{3j}$. Esta condición se verifica sólo en $w^2_{43} = 1 = w^2_{34}$,

por tanto se tiene $w^3_{44} = 1$.

$$W_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

La matriz W_4 tiene un uno donde W_3 tenga un uno y además se verifica que $w^4_{ij} = 1$ si $w^3_{i4} = 1 = w^3_{4j}$. Esta condición se verifica sólo en $w^3_{34} = 1 = w^3_{43}$, por tanto se tiene $w^4_{33} = 1$.

$$W_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

La matriz W_5 tiene un uno donde W_4 tenga un uno y además se verifica que $w^5_{ij} = 1$ si $w^4_{i5} = 1 = w^4_{5j}$. Esta condición se verifica sólo en $w^4_{25} = 1 = w^4_{52}$, por tanto se tiene $w^5_{22} = 1$.

$$W_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Se observa como se han introducido 4 unos en la diagonal correspondientes a los elementos $(2,2),(3,3),(4,4),(5,5)$ que son los elementos que hay que unir a R para tener la transitividad.

Ejercicio 154. Para construir la clausura reflexiva hemos de añadir los elementos

$$(b,b),(c,c),(d,d)(e,e).$$

De este modo:

$$C_r = \{(a,a),(b,b),(c,c),(d,d),(e,e),(a,c),(c,e),(c,d),(b,d)\}.$$

Para computar la clausura transitiva observamos que $(a,c),(c,e) \in C_r$ por tanto debemos unir (a,e). De igual modo $(a,c),(c,d) \in C_r$ por tanto debemos

unir (a, d). Es una comprobación demostrar la siguiente igualdad, siendo C_t la clausura transitiva de C_r:

$$C_t = \{(a,a),(b,b),(c,c),(d,d),(e,e),(a,c),(c,e),(a,e),(c,d),(a,d),(b,d)\}.$$

Ejercicio 155. Para calcular la clausura reflexiva hay que unir todos los miembros de la diagonal. Como $(f,a),(a,c) \in R$ entonces $(f,c) \in C_t$. Ahora $(f,c),(c,e) \in C_t$ implica $(f,e) \in C_t$. Como $(a,c),(c,e) \in R$ entonces $(a,e) \in C_t$. Como $(a,c),(c,d) \in R$ entonces $(a,d) \in C_t$. Como $(f,c),(c,d) \in R$ entonces $(f,d) \in C_t$. Finalmente $(f,b),(b,d) \in R$ entonces $(f,d) \in C_t$. De este modo:

$$C_t = \{(a,a),(a,c),(a,d),(a,e),(b,b),(b,d),(c,c),(c,d),(c,e),$$

$$(d,d),(e,e),(f,a),(f,b),(f,c),(f,d),(f,e),(f,f)\}.$$

En efecto, la matriz que lo representa es:

$$A = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Y se tiene que A^2 no tiene entradas no nulas nuevas, luego es en efecto una relación transitiva.

$$A^2 = \begin{pmatrix} 1 & 0 & 2 & 3 & 3 & 0 \\ 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 2 & 3 & 5 & 4 & 1 \end{pmatrix}.$$

Viendo su diagrama de Hasse (ver Figura 6.8) se tiene que hay dos elementos maximales que son d y e y por tanto no hay máximo. Hay un elemento minimal f que es además el mínimo. El elemento d es cota superior de B y es además la única, por tanto es el supremo. La única cota inferior es f que por tanto es el ínfimo de B.

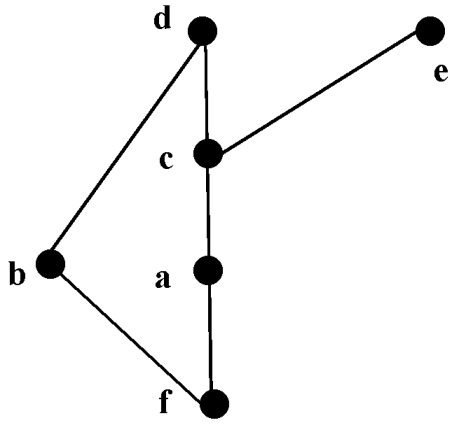


Figura 6.8: Ejercicio 155. Diagrama de Hasse

Ejercicio 156. Representamos cada una de las tareas por su inicial (ensamblar con la E y empaquetar con la e):

$$Tareas = \{E, a, p, e, l, r\}$$

Las prioridades son:

$$E \leq a, a \leq p, p \leq e, l \leq p.$$

Y haciendo su clausura reflexiva y transitiva dan un orden parcial:

$$\{(E,a),(a,p),(E,p),(p,e),(E,e),(a,e),(l,p),(l,e)$$

$$(E,E),(a,a),(p,p),(e,e),(l,l),(r,r)\}.$$

El orden total que completa este orden parcial (ver diagrama de Hasse en la Figura 6.9) se va construyendo de la siguiente manera. Se elige un elemento minimal, por ejemplo el E. Tomamos el conjunto $Tareas - \{E\}$ y el orden que se induce:

$$\{(a,p),(p,e),(a,e),(l,p),(l,e)$$

$$(a,a),(p,p),(e,e),(l,l),(r,r)\}.$$

Ahora elegimos un elemento minimal a y el conjunto resultante de quitar el elemento a, con el orden inducido. Así sucesivamente obtenemos, por ejemplo

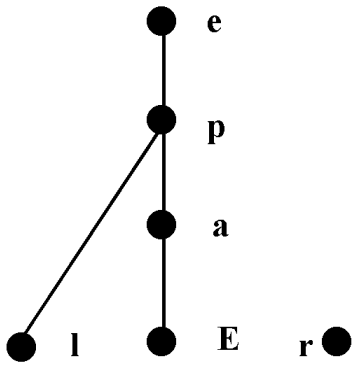


Figura 6.9: Ejercicio 156. Diagrama de Hasse

(la elección del elemento minimal no es única así que hay varias maneras de hacerlo):

$$E \leq a \leq l \leq p \leq e \leq r.$$

Ejercicio 157.

i) La relación definida en i) no es reflexiva puesto que, por ejemplo, $7+7 = 14$ no verifica ser menor que 9. Por tanto no puede ser relación de orden ni de equivalencia. Verifica la propiedad simétrica ya que $a + b = b + a$. No es transitiva ya que, por ejemplo, $7R1$ y $1R7$ y sin embargo $7 + 7 > 9$. De hecho la relación es:

$$R = \{(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(2,1),(2,2),(2,3),(2,4),$$

$$(2,5),(2,6),(3,1),(3,2),(3,3),(3,4),(3,5),(4,1),(4,2),(4,3),(4,4),$$

$$(5,1),(5,2),(5,3),(6,1),(6,2),(7,1)\}$$

ii) La relación de paralelismo es reflexiva (toda recta es paralela a sí misma) simétrica (si r es paralela a r' entonces r' es paralela a r) y transitiva (si r es paralela a r' y r' lo es a r'' entonces r es paralela a r''). De este modo es una relación de equivalencia.

Podemos tomar como representante en cada clase de equivalencia la recta que pasa por el origen, y entonces el conjunto cociente se puede identificar con la semicircunferencia unidad, viendo cada recta como el único punto de corte con dicha semicircunferencia. Esto es el suconjunto del plano real definido como $\{(x,y):\ x^2+y^2=1,\ x\geq 0\}-\{(-1,0)\}$.

iii) La relación es reflexiva ya que la cualidad de ser número o letra la comparte cada elemento de A consigo mismo. Es simétrica porque si $m,n\in A$ son ambos números (respectivamente ambos letras) entonces la misma propiedad la tienen $n,m\in A$. De igual manera es transitiva. Por tanto es una relación de equivalencia. El conjunto cociente A/R es un conjunto de dos elementos, la clase de los números y la clase de las letras.

iv) La relación es reflexiva porque $3a>a$ para cada número natural, en particular para cada elemento de B. No es antisimétrica ya que, por ejemplo, $1S2$ ya que $3>2$ y $2S1$ ya que $6>1$. No es simétrica ya que, por ejemplo, $90S1$ pero $3\times 1<90$. Por tanto no es de equivalencia ni de orden. La relación es:

$$S=\{(1,1),(2,2),(3,3),(4,4),(7,7),(8,8),(90,90),(1,2),(2,1),(2,3),$$

$$(2,4),(3,1),(3,2),(3,3),(3,4),(3,7),(4,1),(4,2),(4,3),(4,7),(4,8),(4,90),$$

$$(7,1),(7,2),(7,3),(7,4),(7,8),(7,90),(8,1),(8,2),(8,3),$$

$$(8,4),(8,7),(90,1),(90,2),(90,3),(90,4),(90,7),(90,8),(90,90)\}.$$

Ejercicio 158.

i) El digrafo es $(\mathbb{N}_7,R)$ cuya matriz es

$$\begin{pmatrix}1&1&1&1&1&1&1\\1&1&1&1&1&1&0\\1&1&1&1&1&0&0\\1&1&1&1&0&0&0\\1&1&1&0&0&0&0\\1&1&0&0&0&0&0\\1&0&0&0&0&0&0\end{pmatrix}$$

ii) Sean $A'=\{a,b,c,d\}$ y $B'=\{1,2,3\}$ entonces el digrafo asociado a la relación es: $(A,(A'\times A')\cup(B\times B'))$. Su matriz (con el orden en el que esta

escrito A) es:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

iv) El digrafo asociado es (B, S) y una matriz de adyacencias con el orden en que está escrito B es:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Ejercicio 159. La única relación que no es reflexiva es la i) y su clausura reflexiva es añadir $(5,5)$, $(6,6)$ y $(7,7)$.

La única relación que no es simétrica es la iv). Como la matriz de su digrafo tiene todas las entradas de la forma $a_{ij} = 1$ cuando $i \leq j$ entonces su clausura simétrica es $B \times B$.

Tomando la matriz del digrafo de i) se observa que su cuadrado tiene todas las entradas no nulas de modo que su clausura transitiva es $\mathbb{N}_7 \times \mathbb{N}_7$.

La clausura transitiva de iv) consiste en añadir los elementos

$$(1,3), (1,4), (1,7), (1,8), (2,4), (2,7), (2,8), (3,8).$$

Ejercicio 160. La relación es antisimétrica, transitiva y no reflexiva. Su clausura reflexiva consiste en unir toda la diagonal.

Mirando el diagrama de Hasse (ver la Figura 6.10) se observa que 1, 3 y 5 son minimales y 2, 4 y 6 maximales. No hay máximo ni mínimo.

Ejercicio 161. De entre todas las rectas paralelas (en la misma clase de equivalencia) elegimos la que pasa por el origen. Por tanto podemos identificar cada clase de equivalencia con el ángulo que forma dicha recta con el eje de abscisas, esto es, con el intervalo $[0, 180) \subset \mathbb{R}$.

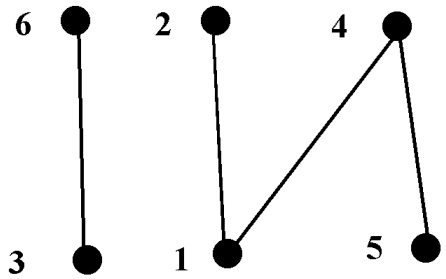


Figura 6.10: Ejercicio 160. Diagrama de Hasse

Ejercicio 162. La matrix de R es:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

La matriz de su clausura reflexiva es:

$$C_r = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Como

$$C_r^2 = \begin{pmatrix} 1 & 2 & 2 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

se tiene que la matriz de la clausura transitiva es:

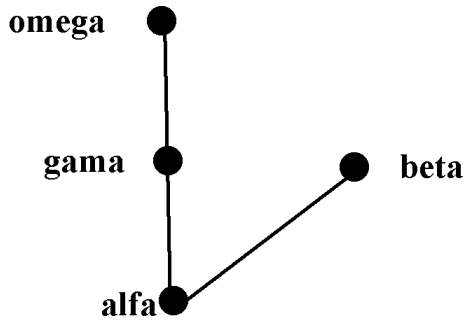


Figura 6.11: Ejercicio 162. Diagrama de Hasse

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

pues esta matriz al cuadrado no tiene entradas no nulas nuevas.

Una relación de orden total que la completa podría ser:

$$\alpha \leq \beta \leq \gamma \leq \omega.$$

Ejercicio 163. Sean $x, y \in A$. Para comprobar que $(x \vee y)' = x' \wedge y'$, es suficiente con verificar que $(x \vee y) \vee (x' \wedge y') = 1$ y que $(x \vee y) \wedge (x' \wedge y') = 0$. Veámoslo: $(x \vee y) \vee (x' \wedge y') = ((x \vee y) \vee x') \wedge ((x \vee y) \vee y') = (x \vee x') \vee y) \wedge (x \vee 1) = ((1 \vee y) \wedge 1)) = 1 \wedge 1 = 1$. De forma similar, $(x \vee y) \wedge (x' \wedge y') = (x \wedge (x' \wedge y') \vee (y \wedge (x' \wedge y') = (0 \wedge y') \vee ((y \wedge y') \wedge x') = 0 \vee (0 \wedge x') = 0 \vee 0 = 0$. La otra propiedad (ley de De Morgan) se obtiene directamente por dualidad.

Capítulo 7

Controles y exámenes resueltos

En esta sección coleccionamos controles y exámenes resueltos de (algunos) cursos anteriores, como un instrumento útil para el estudio de la asignatura. Algunos ejercicios han sido extraídos de diferentes libros. Es de señalar el texto [GLP] recogido en la bibliografía.

LADE+ITIG. Matemática Discreta. Control: 21-12-2005

La duración del examen es de una hora.

Consta de tres ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (3 puntos) Demostrar por inducción que la siguiente desigualdad es cierta para cada número natural n:

$$(2n)! < 2^{2n}(n!)^2.$$

Ejercicio 2. (3 puntos) Un juego de lotería consiste en marcar diez números entre el 1 y el 70. La combinación ganadora está formada por 20 números. Hay premio si se aciertan 10, 9, 8 o ninguno de estos 20 números.

(i) (1 punto) Determinar cuántas posibles apuestas hay.
(ii) (1 punto) Determinar la probabilidad de acertar 10 números.
(iii) (1 punto) Determinar la probabilidad de obtener premio.

Ejercicio 3. (4 puntos) Dados n, p dos números enteros mayores que 1 construir un algoritmo que determine si existe el inverso de n para el producto módulo p y, si existe, lo calcule. Recordamos que este inverso es un número entero m tal que $n \times m \equiv 1 \ mod \ p$. Podemos llamar, sin necesidad de construirlo, a un algoritmo $mcd(x, y)$ que calcula el máximo común divisor de dos enteros cualesquiera x e y.

LADE+ITIG. Matemática Discreta. Control Resuelto: 21-12-2005

La duración del examen es de una hora.

Consta de tres ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (3 puntos) Demostrar por inducción que la siguiente desigualdad es cierta para cada número natural n:

$$(2n)! < 2^{2n}(n!)^2.$$

Base de inducción: Para $n = 1$ se tiene que la desigualdad es verdadera

$2! = 2 < 2^2(1)^2 = 4$.

Paso de inducción: Como la primera parte de la desigualdad en el caso $n+1$ es

$$(2(n+1))! = (2n+2)! = (2n+2)(2n+1)(2n)!,$$

entonces por hipótesis de inducción se tiene que

$$(2(n+1))! < (2n+2)(2n+1)2^{2n}(n!)^2.$$

La segunda parte de la desigualdad en el caso $n+1$ es

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

De este modo basta comprobar, usando la desigualdad obtenida de la hipótesis de inducción:

$$(2n+2)(2n+1) < 2^2(n+1)^2.$$

Esta desigualdad es equivalente a

$$4n^2 + 6n + 2 < 4n^2 + 8n + 4$$

y por tanto verdadera.

Ejercicio 2. (3 puntos) Un juego de lotería consiste en marcar diez números entre el 1 y el 70. La combinación ganadora está formada por 20 números. Hay premio si se aciertan 10, 9, 8 o ninguno de estos 20 números.

(i) (1 punto) Determinar cuántas posibles apuestas hay.

Hay que elegir 10 números entre setenta posibles, esto es,

$$\binom{70}{10}.$$

(ii) (1 punto) Determinar la probabilidad de acertar 10 números.

Hay que elegir los 10 números entre los 20 que salen:

$$\binom{20}{10}/\binom{70}{10}.$$

(iii) (1 punto) Determinar la probabilidad de obtener premio.

Hay que sumar la probabilidad de acertar 10, 9, 8 y 0. Esto será:

$$(\binom{20}{10}+\binom{20}{9}\times 50+\binom{20}{8}\times\binom{50}{2}+\binom{50}{10})/\binom{70}{10}.$$

Ejercicio 3. (4 puntos) Dados n, p dos números enteros mayores que 1 construir un algoritmo que determine si existe el inverso de n para el producto módulo p y, si existe, lo calcule. Recordamos que este inverso es un número entero m tal que $n \times m \equiv 1 \ mod \ p$. Podemos llamar, sin necesidad de construirlo, a un algoritmo $mcd(x, y)$ que calcula el máximo común divisor de dos enteros cualesquiera x e y. Sabemos que el inverso de n módulo p existe si y solamente si $mcd(n, p) = 1$. Esta será la primera línea de pseudocódigo. Si existe, entrará en un bucle que va comprobando si el producto da uno hasta encontrar el inverso.

Entrada: n, p
$I := mcd(n, p)$
If $I \neq 1$ then $R :=$*n no tiene inverso módulo p*
else
$R := 2$
while $R \times n \ mod \ p \neq 1$
    $R := R + 1$
Salida: R

LADE+ITIG. Matemática Discreta. Examen Final: 2-2-2006

La duración del examen es de tres horas.

Consta de 6 ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (1 punto) Dados dos números reales a y r, $r \neq 1$ demostrar por inducción para cada número natural n:

$$a + ar + ar^2 + \cdots + ar^n = \frac{a(r^{n+1} - 1)}{r - 1}.$$

Ejercicio 2. (1 punto) Formaliza y determina la validez del siguiente razonamiento:
Hipótesis:

Si estudio, entonces no suspenderé matemáticas.

Si no juego al baloncesto, estudio.

Suspendo matemáticas.

Tesis:

Juego al baloncesto.

Ejercicio 3. (2 puntos)

(i) (1 punto) Determinar de cuántas maneras se pueden repartir 12 camisetas iguales entre 5 personas.

(ii) (1 punto) La misma pregunta que en (i) pero no permitiendo que ninguna persona reciba (estrictamente) más de siete camisetas.

Ejercicio 4. (1 punto) Sea $G = (V, E)$ un grafo simple conexo, de modo que $|V| = n$.

(i) (0.5 puntos) Determinar, en función de n, el número máximo y el número mínimo de aristas de G.

(ii) (0.5 puntos) Determinar cuál es el número mínimo de aristas que puede tener si es euleriano.

Ejercicio 5. (2 puntos) Sean los tres grafos simples siguientes

$$G_1 = (V_1 = \{a, b, c\}, E_1 = \{\{a, b\}, \{b, c\}, \{a, c\}\}),$$

$$G_2 = (V_2 = \{1, 2\}, E_2 = \{\{1, 2\}\}),$$

$$G_3 = (V_3 = \{\alpha, \beta, \gamma, \epsilon\}, E_3 = \{\{\alpha, \beta\}, \{\alpha, \gamma\}, \{\gamma, \epsilon\}\}).$$

(i) (1 punto) Construir el grafo $G = (G_1 \times G_2) \cup G_3$. Determinar los grados de sus vértices. Construir una matriz de adyacencias de G.

(ii) (0.5 punto) Determinar si el grafo G es conexo, o, en su defecto, cuántas componentes conexas tiene. Determinar si es Euleriano, determinar si es Hamiltoniano.

(iii) (0.5 punto) Determinar cuántos caminos simples de longitud 3 tienen su extremo inicial y su extremo final en el vértice $(a, 1)$.

Ejercicio 6. (3 puntos) Sea el siguiente sistema de congruencias lineales:

$$x \equiv 5 \; mod \; 7$$

$$x \equiv 3 \; mod \; 9$$

$$x \equiv 4 \; mod \; 5$$

(i) (0.5 puntos) Determinar el conjunto de soluciones del sistema.

(ii) (2 puntos) Construir un algoritmo que tenga como entrada dos números naturales distintos a, b y que determine cuántas soluciones del sistema anterior (el del apartado (i)) son mayores o iguales que el mínimo de $\{a, b\}$ y menores o iguales que el máximo de $\{a, b\}$.

(iii) (0.5) Definir el tamaño de la entrada en (ii) y estudiar la complejidad del algoritmo.

LADE+ITIG. Matemática Discreta. Examen Final resuelto: 2-2-2006

La duración del examen es de tres horas.

Consta de 6 ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (1 punto) Dados dos números reales a y r, $r \neq 1$ demostrar por inducción para cada número natural n:

$$a + ar + ar^2 + \cdots + ar^n = \frac{a(r^{n+1} - 1)}{r - 1}.$$

Para $n = 1$ se tiene que el primer término de la igualdad en cuestión es

$$a + ar = a(r + 1).$$

Por otro lado, el segundo término es:

$$\frac{a(r^2 - 1)}{r - 1} = \frac{a(r + 1)(r - 1)}{r - 1} = a(r + 1),$$

por lo que la base de la inducción es verdadera. Como

$$a + ar + ar^2 + \cdots + ar^{n+1} = (a + ar + ar^2 + \cdots + ar^n) + ar^{n+1},$$

entonces, por hipótesis de inducción, la anterior suma es igual a

$$\frac{a(r^{n+1} - 1)}{r - 1} + ar^{n+1}.$$

Resulta una comprobación verificar que esto es igual a la fórmula buscada, esto es:

$$\frac{a(r^{n+2} - 1)}{r - 1}.$$

Ejercicio 2. (1 punto) Formaliza y determina la validez del siguiente razonamiento:
Hipótesis:

Si estudio, entonces no suspenderé matemáticas.

Si no juego al baloncesto, estudio.

Suspendo matemáticas.

Tesis:

Juego al baloncesto.

Las hipótesis son:

$H_1 := (e \Rightarrow \neg s)$

$H_2 := (\neg b \Rightarrow e)$

$H_3 := s$

La tesis es $T := b$. Donde e es *estudio*, s es *suspendo matemáticas*, b es *juego al baloncesto*.

Debemos verificar que la forma proposicional

$$(H_1 \wedge H_2 \wedge H_3) \Rightarrow T$$

es una tauotlogía.

Si las hipótesis son falsas no hay nada que comprobar, entonces podemos suponer $H_1 : V$, $H_2 : V$, $H_3 : V$. Como $H_3 : V$ entonces $s : V$ y para que H_1 sea V debe ser $e : F$. Para que H_2 sea V debe ser $b : V$ lo que hace que $T : V$, siendo, de este modo, una tautología y el razonamiento válido.

Ejercicio 3. (2 puntos)

(i) (1 punto) Determinar de cuántas maneras se pueden repartir 12 camisetas iguales entre 5 personas.

(ii) (1 punto) La misma pregunta que en (i) pero no permitiendo que ninguna persona reciba más (estrictamente) de siete camisetas.

(i) $CR_{5,12} = \binom{16}{12}$.

(ii) Asignamos 8 camisetas a una persona. Repartimos las 4 restantes para obtener el la cantidad de repartos que asignan más de 7 camisetas a una persona fijada. Como hay cinco personas, debe ir multiplicado por 5, de modo que la cantidad buscada es

$$CR_{5,12} - 5CR_{5,4}.$$

Ejercicio 4. (1 punto) Sea $G = (V, E)$ un grafo simple conexo, de modo que $|V| = n$.

(i) (0.5 puntos) Determinar, en función de n, el número máximo y el número mínimo de aristas de G.

(ii) (0.5 puntos) Determinar cuál es el número mínimo de aristas que puede tener si es euleriano.

(i) El grafo conexo mínimo es un árbol de modo que $|V| \geq n-1$. El grafo con más aristas será K_n. De modo que

$$n-1 \leq |V| \leq n(n-1)/2.$$

(ii) Como todos los vértices han de ser de grado par, entonces, el grado mínimo es dos y por tanto $|V| = n$. El ciclo C_n es un ejemplo donde se alcanza este mínimo.

Ejercicio 5. (2 puntos) Sean los tres grafos simples siguientes

$$G_1 = (V_1 = \{a, b, c\}, E_1 = \{\{a, b\}, \{b, c\}, \{a, c\}\}),$$

$$G_2 = (V_2 = \{1, 2\}, E_2 = \{\{1, 2\}\}),$$

$$G_3 = (V_3 = \{\alpha, \beta, \gamma, \epsilon\}, E_3 = \{\{\alpha, \beta\}, \{\alpha, \gamma\}, \{\gamma, \epsilon\}\}).$$

(i) (1 punto) Construir el grafo $G = (G_1 \times G_2) \cup G_3$. Determinar los grados de sus vértices. Construir una matriz de adyacencias de G.

(ii) (0.5 punto) Determinar si el grafo G es conexo, o, en su defecto, cuántas componentes conexas tiene. Determinar si es Euleriano, determinar si es Hamiltoniano.

(iii) (0.5 punto) Determinar cuántos caminos simples de longitud 3 tienen su extremo inicial y su extremo final en el vértice $(a, 1)$.

(i) El grafo es $G = (V, E)$ donde

$$V = \{(a,1), (b,1), (c,1), (a,2), (b,2), (c,2), \alpha, \beta, \gamma, \epsilon\}$$

y

$$E = \{\{(a,1),(b,1)\}, \{(b,1),(c,1)\}, \{(a,1),(c,1)\}, \{(a,2),(b,2)\}, \{(b,2),(c,2)\},$$

$$\{(a,2),(c,2)\}, \{(a,1),(a,2)\}, \{(b,1),(b,2)\}, \{(c,1),(c,2)\}\}.$$

Los grados verifican: $gr(v) = 3$ para cada $v \in V_1 \times V_2$ $gr(\alpha) = gr(\gamma) = 2$, $gr(\beta) = gr(\epsilon) = 1$.

Para el orden escogido en los vértices, una matriz puede ser:

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

(ii) El grafo es no conexo pues tiene dos componentes conexas, ya que la unión es disjunta y cada uno de los miembros de la unión es conexo. Al no ser conexo no puede ser Hamiltoniano. Al tener dos componentes conexas, cada una de ellas con aristas no puede ser Euleriano.

(iii) Como el extremo inicial y final del camino es el mismo, no puede haber caminos no simples de esa longitud. De modo que cada camino de longitud 3 es, en efecto, simple. Hay dos caminos simples de longitud 3, que son:

$$(a,1),(b,1),(c,1),(a,1)\ y (a,1),(c,1),(b,1),(a,1).$$

Y no puede haber más pues en la componente conexa en que está $(a,1)$, que es un producto, si el camino sube a la altura definida por la segunda coordenada igual a 2, no puede en dos pasos, volver al vértice $(a,1)$.

Ejercicio 6. (3 puntos) Sea el siguiente sistema de congruencias lineales:

$$x \equiv 5\ mod\ 7$$

$$x \equiv 3\ mod\ 9$$

$$x \equiv 4\ mod\ 5$$

(i) (0.5 puntos) Determinar el conjunto de soluciones del sistema.

(ii) (2 puntos) Construir un algoritmo que tenga como entrada dos números naturales distintos a, b que determine cuántas soluciones del sistema anterior sean mayores o iguales que el mínimo de $\{a,b\}$ y menores o iguales que el máximo de $\{a,b\}$.

(iii) (0.5) Definir el tamaño de la entrada en (ii) y estudiar la complejidad del algoritmo.

(i) Se puede aplicar el teorema Chino de los restos ya que 5 y 7 son números primos y 9 no es un múltiplo suyo. Entonces el conjunto de soluciones es:

$$\{264 + 7 \times 5 \times 9K : K \in \mathbb{Z}\}.$$

(ii) Podemos hacer el siguiente algoritmo

Entrada: a, b
If $a < b$ then $m := a$, $M := b$ else $m := b$, $M := a$
$i := 0$
$s := 0$
while $264 + 7 \times 5 \times 9i \leq M$
    if $264 + 7 \times 5 \times 9i \geq m$ then $s := s + 1$
    $i := i + 1$
Salida: s

(iii) Definimos M como el tamaño de la entrada y entonces el algoritmo es de complejidad lineal.

LADE+ITIG. Matemática Discreta. Examen Final: 5-9-2006

La duración del examen es de tres horas.

Consta de 4 ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (20 puntos) Sea $K_n = (V, E)$ el grafo simple completo de n vértices.

(i) (4 puntos) Determinar $|V|$, el grado de cada vértice y, usando la fórmula que relaciona el grado de los vértices con el número de aristas, determinar $|E|$.

(ii) (4 puntos) Demuestra la fórmula obtenida para $|E|$ por inducción en el número de vértices (sin usar la fórmula que relaciona el grado de los vértices con el número de aristas).

(iii) (4 puntos) Determinar para qué valores de n el grafo K_n es euleriano. Y para cuáles es hamiltoniano.

(iv) (4 puntos) Determinar cuántos subgrafos isomorfos a C_3 (el ciclo de tres elementos) tiene el grafo K_n. La misma pregunta cambiando C_3 por C_4.

(v) (4 puntos) Determinar, fijada una arista $e \in E$, cuántos subgrafos de K_n isomorfos a C_3 contienen la arista e.

Ejercicio 2. (20 puntos)

(i) (8 puntos) Demostrar que para todo número natural n, $n^2 + 1$ no es divisible por 19.

(ii) (12 puntos) Determinar la probabilidad de que al elegir tres números naturales menores o iguales que 100 (posiblemente repetidos)

(a) sean los tres múltiplos de 17,

(b) al menos uno de los tres elegidos sea congruente con 5 módulo 17,

(c) sean soluciones del sistema de congruencias:

$x \equiv 2 \bmod 3$

$x \equiv 5 \bmod 7$

$x \equiv 4 \bmod 2$

Ejercicio 3. (20 puntos) Sea $G = (V, E)$ un grafo simple (no dirigido) dado como:

$V = v_1, \ldots, v_n$ es una lista de números enteros no repetidos y

$E = (e_{11}, e_{12}), \ldots, (e_{m1}, e_{m2})$ es una lista de pares ordenados (listas de dos elementos) de números enteros de V.

Construir los siguientes algoritmos y estudiar su complejidad:

(4 puntos) *vértice.* Tiene como entrada G y un número entero v. Determina si v es un vértice de G.

(4 puntos) *arista.* Tiene como entrada G y un par ordenado $e := v_1, v_2$. Determina si e es una arista de G.

(4 puntos) *adyacentes.* Tiene como entrada G y un número entero v. Determina todas las aristas que son adyacentes con v.

(4 puntos) *grado.* Tiene como entrada G y un número entero v. Determina el grado de v.

(4 puntos) *euleriano.* Tiene como entrada G (conexo) y determina si G tiene o no un camino euleriano.

Ejercicio 4. (20 puntos)

(i) (10 puntos) En el conjunto de las formas proposicionales definimos la siguiente relación: $\mathcal{P}$ se relaciona con $\mathcal{Q}$ si y solamente si $\mathcal{P} \vee \mathcal{Q}$ es una tautología. Determinar si esta relación es reflexiva, simétrica, antisimétrica y transitiva.

(ii) (10 puntos) En el conjunto $A = \{(p \vee \neg p), (q \Rightarrow p), p, (q \wedge \neg q)\}$ se define la relación siguiente: dados $\mathcal{P}, \mathcal{Q} \in \mathcal{A}$ se tiene que $\mathcal{P}$ se relaciona con $\mathcal{Q}$ si y solamente si $\mathcal{P} \Leftarrow \mathcal{Q}$ es una tautología. Construye el grafo dirigido asociado a esta relación. Computa la clausura reflexiva, la clausura simétrica y la clausura transitiva de la relación.

LADE+ITIG. Matemática Discreta. Examen final resuelto: 5-9-2006

La duración del examen es de tres horas.

Consta de 4 ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (20 puntos) Sea $K_n = (V, E)$ el grafo simple completo de n vértices.

(i) (4 puntos) Determinar $|V|$, el grado de cada vértice y, usando la fórmula que relaciona el grado de los vértices con el número de aristas, determinar $|E|$.

Por definición $V = \{1, \ldots, n\}$ de modo que $|V| = n$. Como contiene todas las aristas posibles entonces cada vértice se une con todos los demás. Esto quiere decir que $gr(v) = n - 1$ para cada $v \in V$. Entonces $\Sigma_{i=1}^{n} gr(v) = n(n-1) = 2|E|$. De modo que

$$E = \frac{n(n-1)}{2}.$$

(ii) (4 puntos) Demuestra la fórmula obtenida para $|E|$ por inducción en el número de vértices (sin usar la fórmula que relaciona el grado de los vértices con el número de aristas).

Para $n = 1$ se tiene que $|E| = 0 = 1(1-1)$, por tanto la fórmula se verifica.

Si $|V| = n + 1$ entonces $K_{n+1} = (V, E)$ es la unión de K_n y $G = (\{1, \ldots, n, n+1\}, \{\{n+1, i\} : i = 1, \ldots, n\})$. De este modo, como queríamos demostrar, se tiene

$$|E| = \frac{n(n-1)}{2} + n = n(\frac{n-1}{2} + 1) = \frac{n(n+1)}{2}.$$

(iii) (4 puntos) Determinar para qué valores de n el grafo K_n es euleriano. Y para cuáles es hamiltoniano.

Como el grafo K_n es conexo, el teorema de Euler caracteriza los grafos eulerianos por la paridad del grado de sus vértices. De este modo si n es impar cada vértice tiene grado par y así K_n es euleriano. El mismo razonamiento asegura que si n es par el grafo no es euleriano.

Los grafos K_1 y K_2 no son hamiltonianos. Si $n > 2$ entonces el circuito $1, \ldots, n, 1$ no pasa dos veces por el mismo vértice y los recorre todos siendo así el grafo hamiltoniano.

(iv) (4 puntos) Determinar cuántos subgrafos isomorfos a C_3 (el ciclo de tres elementos) tiene el grafo K_n. La misma pregunta cambiando C_3 por C_4.

Basta tomar tres vértices (respectivamente 4), por tanto $\binom{n}{3}$ (respectivamente $\binom{n}{4}$).

(v) (4 puntos) Determinar, fijada una arista $e \in E$, cuántos subgrafos de K_n isomorfos a C_3 contienen la arista e.

Tantos como elegir el tercer vértice entre los $n-2$ vértices restantes, por tanto $n-2$.

Ejercicio 2. (20 puntos)

(i) (8 puntos) Demostrar que para todo número natural n, n^2+1 no es divisible por 19.

Si $n^2 + 1 \equiv 0 \bmod 19$ entonces $n^2 \equiv -1 \equiv 18 \bmod 19$. Pero observamos que, módulo 19:

$$\begin{array}{cccccccc} 0^2 \equiv 0, & 1^2 \equiv 1, & 2^2 \equiv 4, & 3^2 \equiv 9, & 4^2 \equiv 16, & 5^2 \equiv 6, & 6^2 \equiv 17, \\ 7^2 \equiv 11, & 8^2 \equiv 7 & 9^2 \equiv 5 & 10^2 \equiv 5 & 11^2 \equiv 7 & 12^2 \equiv 11 & \\ 13^2 \equiv 17, & 14^2 \equiv 6 & 15^2 \equiv 16 & 16^2 \equiv 9 & 17^2 \equiv 4 & 18^2 \equiv 1 & \end{array}$$

Así nunca n^2 es congruente con 18 módulo 19 y el enunciado del ejercicio es verdadero.

(ii) (12 puntos) Determinar la probabilidad de que al elegir tres números naturales menores o iguales que 100 (posiblemente repetidos)

(a) sean los tres múltiplos de 17,

Los múltiplos de 17 menores que 100 son 17, 34, 51, 68 y 85. Por tanto para que los tres sean múltiplos de 17 los tres deben elegirse de este conjunto

de 5 elementos. Así la probabilidad buscada es

$$\frac{5^3}{100^3}.$$

(b) al menos uno de los tres elegidos sea congruente con 5 módulo 17,

Los menores de 100 que son congruentes con 5 módulo 17 son 5, 22, 39, 56, 73 y 90. Entonces el suceso complementario al que se nos pide es que los tres números se escojan entre los 94 restantes, así la probabilidad buscada es:

$$1 - \frac{94^3}{100^3}.$$

(c) sean soluciones del sistema de congruencias:
$x \equiv 2 \bmod 3$
$x \equiv 5 \bmod 7$
$x \equiv 4 \bmod 2$

Las soluciones de este sistema, calculadas por el teorema Chino de los restos, son de la forma $26 + 42 \times k$ con $k \in \mathbb{Z}$. Por tanto sólo dos números menores que 100 son soluciones del sistema. La probabilidad buscada es por tanto

$$\frac{2^3}{100^3}.$$

Ejercicio 3. (20 puntos) Sea $G = (V, E)$ un grafo simple (no dirigido) dado como:

$V = v_1, \ldots, v_n$ es una lista de números enteros no repetidos y

$E = (e_{11}, e_{12}), \ldots, (e_{m1}, e_{m2})$ es una lista de pares ordenados (listas de dos elementos) de números enteros de V.

Construir los siguientes algoritmos y estudiar su complejidad:

(4 puntos) *vértice.* Tiene como entrada G y un número entero v. Determina si v es un vértice de G.

vértice
Entrada: $V = v_1, \ldots, v_n$; $E = (e_{11}, e_{12}), \ldots, (e_{m1}, e_{m2})$; v
$i := 1$
while $v \neq v_i$ do
    $i := i + 1$
if $i = n + 1$ then $r := 0$ else $r := 1$
Salida: r (0 si $v \notin V$, 1 si $v \in V$).

Un algoritmo de complejidad lineal $\mathcal{O}(n)$.

(4 puntos) *arista.* Tiene como entrada G y un par ordenado $e := v_1, v_2$ de número entero v. Determina si e es una arista de G.

Como los grafos son no dirigidos entonces las aristas, aunque se introducen ordenadamente, son en realidad subconjuntos de dos elementos. Necesitamos el siguiente algoritmo adicional que dice si dos pares ordenados tienen los mismos elementos o no

compara aristas
Entrada: $e_{11}, e_{12}, f_{21}, f_{22}$
if $e_{11} = f_{21}$ and $e_{12} = f_{22}$ or
if $e_{11} = f_{22}$ and $e_{12} = f_{21}$ then
$r := 1$ else $r := 0$
Salida: r

arista
Entrada: $V = v_1, \ldots, v_n$; $E = (e_{11}, e_{12}), \ldots, (e_{m1}, e_{m2})$; $e = (e_1, e_2)$
$i := 1$
while *compara aristas*$(e_{i1}, e_{i2}), e = 0$ do
    $i := i + 1$
if $i = n + 1$ then $r := 0$ else $r := 1$
Salida: r (0 si e no es arista y 1 si lo es)

Un algoritmo de complejidad lineal $\mathcal{O}(m)$

(4 puntos) *adyacentes.* Tiene como entrada G y un número entero v. Determina todas las aristas que son adyacentes con v.

adyacentes
Entrada: $V = v_1, \ldots, v_n$; $E = (e_{11}, e_{12}), \ldots, (e_{m1}, e_{m2})$; v
$j := 0$
for $i = 1$ to m do
    if $v = e_{i1}$ or $v = e_{i2}$ then
    $j := j + 1$ $adyacentes_j := (e_{i1}, e_{i2})$
if $j = 0$ then $r := 0$ else $r :=$ *adyacentes*
Salida: r (0 si v no es extremo de ninguna arista y la lista de aristas adyacentes en caso contrario.)

Es un algoritmo lineal $\mathcal{O}(m)$.

(4 puntos) *grado.* Tiene como entrada G y un número entero v. Determina el grado de v.

En el algoritmo anterior la j como salida da el grado.

(4 puntos) *euleriano.* Tiene como entrada G (conexo) y determina si G tiene o no un camino euleriano.

camino euleriano
Entrada: $V = v_1, \ldots, v_n$; $E = (e_{11}, e_{12}), \ldots, (e_{m1}, e_{m2})$
$j := 0$
for $i = 1$ to n do
    if $grado(v)$ mod 2=1 then $j := j + 1$
If $j \leq 2$ then $r := 1$ else $r := 0$
Salida: r (1 si hay tal camino, 0 si no lo hay)

Complejidad: $\mathcal{O}(nm)$.

Ejercicio 4. (20 puntos)
(i) (10 puntos) En el conjunto de las formas proposicionales que se pueden construir con las variables proposicionales p, q, r, s definimos la siguiente relación: $\mathcal{P}$ se relaciona con $\mathcal{Q}$ si y solamente si $\mathcal{P} \vee \mathcal{Q}$ es una tautología. Determinar si esta relación es reflexiva, simétrica, antisimétrica y transitiva.

No es reflexiva: tomamos $\mathcal{P} := p$, esto es, una variable proposicional. Como $p \vee p$ no es una tautología la relación no es reflexiva.

Es simétrica: porque $\mathcal{P} \vee \mathcal{Q}$ es lógicamente equivalente a $\mathcal{Q} \vee \mathcal{P}$.

No es antisimétrica: p se relaciona con $\neg p$, $\neg p$ se relaciona con p y son formas proposicionales diferentes.

No es transitiva: Sea $\mathcal{P} = p$, $\mathcal{Q} = (q \vee \neg q)$ y $\mathcal{R} = r$. Como $\mathcal{Q}$ es una tautología se tiene que $\mathcal{P} \vee \mathcal{Q}$ y $\mathcal{Q} \vee \mathcal{R}$ son tautologías. De este modo $\mathcal{P}$ se relaciona con $\mathcal{Q}$ y $\mathcal{Q}$ se relaciona con $\mathcal{R}$. Pero $\mathcal{P}$ no se relaciona con $\mathcal{R}$ puesto que $p \vee r$ no es una tautología.

(ii) (10 puntos) En el conjunto $A = \{(p \vee \neg p), (q \Rightarrow p), p, (q \wedge \neg q)\}$ se define la relación siguiente: dados $\mathcal{P}, \mathcal{Q} \in \mathcal{A}$ se tiene que $\mathcal{P}$ se relaciona con $\mathcal{Q}$ si y solamente si $\mathcal{P} \Leftarrow \mathcal{Q}$ es una tautología. Construye el grafo dirigido asociado a esta relación. Computa la clausura reflexiva, la clausura simétrica y la clausura transitiva de la relación.

El grafo es (V, E) donde $V = \{v_1, v_2, v_3, v_4\}$ y

$$v_1 := (p \vee \neg p),\ v_2 := (q \Rightarrow p),\ v_3 := p,\ v_4 := (q \wedge \neg q).$$

Como $\mathcal{P} \Leftarrow \mathcal{P}$ es una tautología entonces la relación es reflexiva, de modo que $(v_i, v_i) \in E$, $1 \leq i \leq 4$. Como $(q \wedge \neg q)$ es una contradicción entonces $(v_i, v_4) \in E$, $1 \leq i \leq 4$. Como v_1, v_2 y v_3 no son contradicciones entonces $(v_4, v_i) \notin E$ para $i = 1, 2, 3$. Como v_1 es una tautología y v_2, v_3 y v_4 no lo son entonces $(v_i, v_1) \notin E$, $i = 2, 3, 4$. Verificamos el resto de posibles adyacencias:

$(v_1, v_2) \notin E$ ya que $(p \vee \neg p) \Leftarrow (q \Rightarrow p)$ no es una tautología.

$(v_1, v_3) \in E$ ya que $(p \vee \neg p) \Leftarrow p$ es una tautología.

$(v_2, v_3) \in E$ ya que $(q \Rightarrow p) \Leftarrow p$ es una tautología.

$(v_3, v_2) \notin E$ ya que $p \Leftarrow (q \Rightarrow p)$ no es una tautología.

De esta manera el grafo es

$$G = (V\{v_1, v_2, v_3, v_4\}, E = \{(v_1, v_1), (v_2, v_2), (v_3, v_3),$$

$$(v_4, v_4), (v_1, v_4), (v_2, v_4), (v_3, v_4), (v_1, v_3), (v_2, v_3)\}).$$

Como la relación es reflexiva entonces coincide con su clausura transitiva. La clausura simétrica añade las aristas

$$(v_4, v_1), (v_4, v_2), (v_4, v_3), (v_3, v_1), (v_3, v_2).$$

La relación es transitiva y por tanto coincide con su clausura transitiva.

LADE+ITIG. Matemática Discreta. Control: 20-12-2006

La duración del examen es de una hora.

Consta de dos ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (3 puntos) Supongamos que tenemos un algoritmo, llamado mcd, que calcula el máximo común divisor de dos números naturales m, n, para lo que basta escribir $mcd(m, n)$. Supongamos también que podemos asignar pares de números (m, n) a una variable x para lo que basta escribir $x := (m, n)$. Usando estos dos algoritmos (sin necesidad de construirlos) construir un algoritmo que tenga como entrada una lista $a_1, \ldots, a_n$ de números naturales y como salida la lista de pares (a_i, a_j) de números de la lista $a_1, \ldots, a_n$ que son primos entre sí.

(1 punto) Asumiendo que el algoritmo mcd es de complejidad constante determinar la complejidad del algoritmo anterior.

Ejercicio 2. En una biblioteca hay dos estanterías: Estantería 1, Estantería 2. En la Estantería 1 hay 10 libros de historia, 3 de matemáticas y 23 de filosofía. En la Estantería 2 hay 15 libros de historia, 25 de matemáticas y 5 de filosofía. Extraemos tres libros de la Estantería 1 y tres libros de la Estantería 2. Determinar la probabilidad de que:

(i) (1 punto) Los seis libros sean de historia.

(ii) (1 punto) Al menos un libro sea de matemáticas.

(iii) (2 puntos) Haya el mismo número de libros de filosofía en los extraídos de la Estantería 1 que en los de la Estantería 2.

Si consideramos un éxito sacar al menos un libro de matemáticas en los tres extraídos de cada Estantería:

(iv) (1 punto) Determinar la probabilidad p de éxito y q de fracaso.

(v) (1 punto) Determinar la probabilidad de que al repetir el experimento 6 veces se hayan obtenido exactamente 3 fracasos.

LADE+ITIG. Matemática Discreta. Control resuelto: 22-11-2006

La duración del examen es de una hora.

Consta de dos ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. Sean a y d dos números naturales y $\{x_n\}$ una sucesión de numeros naturales definida de la siguiente manera: $x_1 = a$ y $x_n = x_{n-1} + d$ para cada número natural $n > 1$.

(i) (2 puntos) Demostrar por inducción que la fórmula general $x_n = a + (n-1)d$ es válida para cada n natural.

Para $n = 1$ la fórmula dice que $x_1 = a$ lo que es cierto. Si la fórmula es cierta para x_n entonces $x_{n+1} = x_n + d$ es, por hipótesis de inducción, $x_{n+1} = a + (n-1)d + d = a + nd$, lo que demuestra que la fórmula es válida.

(ii) (3 puntos) Demostrar que $\Sigma_{i=1}^{n} x_i = na + \frac{n(n-1)}{2}d$. Lo demostramos por inducción. Para $n = 1$ la fórmula dice que $x_1 = x_1$, lo cual es cierto. Ahora $\Sigma_{i=1}^{n+1} x_i = \Sigma_{i=1}^{n} x_i + x_{n+1}$. Aplicando la hipótesis de inducción y el apartado (i) tenemos

$$\Sigma_{i=1}^{n+1} x_i = na + \frac{n(n-1)}{2}d + a + nd = (n+1)a + \frac{n(n+1)}{2}d,$$

como queríamos demostrar.

Ejercicio 2.

(3 puntos) Construir un algoritmo que tenga como entrada un número natural n y construya la lista de los números naturales menores o iguales que n que son múltiplos de 7 y cuyo resto de dividir por 11 es 3.

Entrada: n

$j := 0$

For $i = 1$ to n
    If $i \bmod 7 = 0$, $i \bmod 11 = 3$ then
    $j := j + 1$, $a_j = i$
Salida: $a_1, \cdots, a_m$.

(1 punto) Determinar el mínimo valor de n para el cual la salida del anterior algoritmo es una lista no vacía.

Por el teorema chino la solución existe y es única módulo 77. Aplicando el algoritmo del teorema chino obtenemos la solución $x = 14$.

(1 puntos) Determinar la función $T(n)$ (con la mayor precisión posible) y la complejidad del algoritmo.

El tamaño de la entrada es n. Primero tenemos una asignación. Después un bucle que se repite n veces. Dentro del bucle se tienen: 1 suma y una asignación del contador, 2 cocientes, 2 comparaciones y 2 asignaciones más 1 suma que se harán algunas veces, del orden de $[n/77]$. La comparación final que nos saca del bucle. La asignación de la salida no la contamos. Entonces $T(n) = 3 + 5n + 3[n/77]$, la complejidad es lineal.

LADE+ITIG. Matemática Discreta. Control: 17-01-2007

La duración del examen es de una hora.

Consta de dos ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (3 puntos) Sea M una matriz cuadrada de tamaño n con coeficientes enteros. Supongamos que la matriz es una lista ordenada con dos subíndices de manera que $M_{i,j}$ es la entrada de M que ocupa la fila i-ésima y la columna j-ésima. Construir un algoritmo que tenga como entrada M y s, donde s es un número entero que verifica $1 \leq s \leq n$, y obtenga el grado del vértice s en el grafo que tiene M como su matriz de adyacencias.

(1 punto)Determinar la complejidad del algoritmo anterior.

Ejercicio 2. Tenemos tres urnas: Urna 1, Urna 2 y Urna 3. En la Urna 1 hay 7 bolas rojas, 3 negras y 5 verdes; en la Urna 2 hay 10 bolas rojas, 4 negras y 6 verdes; en la Urna 3 hay 11 bolas rojas, 13 negras y 34 verdes. Extraemos 4 bolas de cada urna. Determinar la probabilidad de que:

(i) (1 punto) Las 12 bolas sean verdes.

(ii) (1 punto) Al menos una bola sea roja.

(iii) (2 puntos) Haya al menos una bola de cada color en las extraidas en la Urna 1.

Si consideramos un éxito sacar al menos una bola roja en cada urna:

(iv) (1 punto) Determinar la probabilidad p de éxito y q de fracaso.

(v) (1 punto) Determinar la probabilidad de que al repetir el experimento 9 veces se hayan obtenido exactamente 3 fracasos.

LADE+ITIG. Matemática Discreta. Control resuelto: 17-01-2007

La duración del examen es de una hora.

Consta de dos ejercicios. La puntuación de cada uno de los apartados la indica el enunciado.

Las respuestas sin justificación se considerarán no correctas.

Se pueden usar los apuntes pero no calculadoras.

Ejercicio 1. (3 puntos) Sea M una matriz cuadrada de tamaño n con coeficientes enteros. Supongamos que la matriz es una lista ordenada con dos subíndices de manera que $M_{i,j}$ es la entrada de M que ocupa la fila i-ésima y la columna j-ésima. Construir un algoritmo que tenga como entrada M y s, donde s es un número entero que verifica $1 \leq s \leq n$, y obtenga el grado del vértice s en el grafo que tiene M como su matriz de adyacencias.

Entrada: $M_{11}, \ldots, M_{nn}$; s.
$grado := 0$
For $i = 1$ to n
    $grado := grado + M_{si}$
Salida: $grado$

(1 punto)Determinar la complejidad del algoritmo anterior.

Consideramos n el tamaño de la entrada. El bucle suma las n entradas de la columna s-ésima de M, por tanto es lineal.

Ejercicio 2. Tenemos tres urnas: Urna 1, Urna 2 y Urna 3. En la Urna 1 hay 7 bolas rojas, 3 negras y 5 verdes; en la Urna 2 hay 10 bolas rojas, 4 negras y 6 verdes; en la Urna 3 hay 11 bolas rojas, 13 negras y 34 verdes. Extraemos 4 bolas de cada urna. Determinar la probabilidad de que:

(i) (1 punto) Las 12 bolas sean verdes.

$$\frac{\binom{5}{4}}{\binom{15}{4}} \frac{\binom{6}{4}}{\binom{20}{4}} \frac{\binom{34}{4}}{\binom{58}{4}}.$$

(ii) (1 punto) Al menos una bola sea roja.

Este suceso es complementario del suceso: las 12 bolas sean rojas.

$$1 - \frac{\binom{8}{4}}{\binom{15}{4}} \frac{\binom{6}{4}}{\binom{10}{4}} \frac{\binom{47}{4}}{\binom{58}{4}}.$$

(iii) (2 puntos) Haya al menos una bola de cada color en las extraídas en la Urna 1.

Escribimos este suceso como unión disjunta de los tres sucesos siguientes: S_V =sacar dos bolas verdes, una roja y una negra, S_R =sacar dos bolas rojas, una negra y una verde y S_N =sacar dos bolas negras, una roja y una verde. Entonces la probabilidad buscada es $P(S_V) + P(S_R) + P(S_N)$, donde:

$$P(S_V) = \frac{\binom{5}{2} \times 7 \times 3}{\binom{15}{4}},\ P(S_R) = \frac{\binom{7}{2} \times 3 \times 5}{\binom{15}{4}},\ P(S_N) = \frac{\binom{3}{2} \times 7 \times 5}{\binom{15}{4}}.$$

Si consideramos un éxito sacar al menos una bola roja en cada urna:

(iv) (1 punto) Determinar la probabilidad p de éxito y q de fracaso.

Sacar al menos una bola roja es el suceso complementario de que ninguna bola sea roja. Entonces:

$$p = (1 - \frac{\binom{8}{4}}{\binom{15}{4}})(1 - \frac{\binom{10}{4}}{\binom{20}{4}})(1 - \frac{\binom{47}{4}}{\binom{58}{4}})$$

y $q = 1 - p$

(v) (1 punto) Determinar la probabilidad de que al repetir el experimento 9 veces se hayan obtenido exactamente 3 fracasos. Aplicando la fórmula oportuna se tiene:

$$\binom{9}{3} p^6 (1-p)^3.$$

Matemática Discreta. Examen final LADE+ITIG: 10 de febrero del 2007

El examen está compuesto por problemas y se valorará sobre 10 puntos.

La respuesta a cada uno de los problemas se valorará sobre el número de puntos indicado.

Las respuestas sin justificación se considerarán como no contestadas.

No está permitido el uso de calculadoras.

Pueden usarse apuntes.

Problema 1. (2 puntos) Los ocho participantes en una comisión deben alojarse en un hotel. Dicho hotel dispone de una habitación triple, dos habitaciones dobles y una individual.

(1 punto) Determinar de cuántas maneras distintas pueden repartirse los comisionados en las distintas habitaciones.

(1 punto) Determinar la probabilidad de que el comisionado llamado Pepe esté alojado en la habitación individual.

Problema 2. (1 punto) Demostrar por inducción que para cada número natural $n \geq 2$ se tiene

$$2^{2^n} \equiv 6 \bmod 10.$$

Problema 3. (1.5 puntos) Sea el grafo simple $G = (V, E)$ donde $V = \{1, 2, 3\}$ y $E = \{\{1, 2\}, \{2, 3\}\}$.

(0.5 puntos) Construir una matriz de adyacencias del grafo producto $G \times G$.

(0.5 puntos) Determinar el grado de cada vértice de $G \times G$.

(0.5 puntos) Determinar si G es conexo, determinar si $G \times G$ es conexo, determinar si $G \times G$ es Euleriano.

Problema 4. (2 puntos) Sea $\mathbb{Z}_{131} = \{\overline{0}, \overline{1}, \ldots, \overline{129}, \overline{130}\}$. Considera la función siguiente:

$$\begin{array}{rccc} f: & \mathbb{Z}_{131} & \to & \mathbb{Z}_{131} \\ & \overline{a} & \mapsto & \overline{7a+9} \end{array}$$

(0.5 puntos) Determinar $f(\overline{81})$ y $f(\overline{111})$.

(0.75 puntos) Determinar si f es inyectiva.
(0.75 puntos) Determinar si f es sobreyectiva.

Problema 5. (2 puntos) Supongamos que tenemos un algoritmo de complejidad lineal, $Borrar(L, i)$, que tiene como entrada una lista L y un entero i y borra de L el término i-ésimo L_i. Supongamos también que podemos hacer asignaciones de listas $L := M$.

(1.5 puntos) Construir un algoritmo que tenga como entrada una lista L y como salida la lista donde se han borrado todos los elementos de L que son ´múltiplos de 6.

(0.5 puntos) Estudiar su complejidad.

Problema 6. (1.5 puntos) Sea A el conjunto de los números naturales escritos en el sistema de numeración de base 2 y caracterizados por la propiedad de ser menores o iguales que 1000, donde 1000 está escrito también en base 2. En A definimos la siguiente relación: para $a, b \in A$ se tiene que aRb si $a + b \in A$.

(0.5 puntos) Escribir A y determinar si la relación es reflexiva, simétrica y transitiva.

(0.5 puntos) Escribir una matriz de la relación.

(0.5 puntos) Escribir su clausura reflexiva, su clausura simétrica y su clausura transitiva.

Matemática Discreta. Examen final resuelto LADE+ITIG: 10 de febrero del 2007

El examen está compuesto por problemas y se valorará sobre 10 puntos.

La respuesta a cada uno de los problemas se valorará sobre el número de puntos indicado.

Las respuestas sin justificación se considerarán como no contestadas.

No está permitido el uso de calculadoras.

Pueden usarse apuntes.

Problema 1. (2 puntos) Los ocho participantes en una comisión deben alojarse en un hotel. Dicho hotel dispone de una habitación triple, dos habitaciones dobles y una individual.

(1 punto) Determinar de cuántas maneras distintas pueden repartirse los comisionados en las distintas habitaciones.

Basta elegir los tres que van a la habitación triple, de los restantes, los dos que van a la primera doble, de los restantes los que van a la segunda doble y finalmente queda uno, que irá a la individual. De este modo:

$$\binom{8}{3}\binom{5}{2}\binom{3}{2}\binom{2}{2}.$$

(1 punto) Determinar la probabilidad de que el comisionado llamado Pepe esté alojado en la habitación individual.

Como son 8 personas la probabilidad es de $1/8$.

Problema 2. (1 punto) Demostrar por inducción que para cada número natural $n \geq 2$ se tiene

$$2^{2^n} \equiv 6 \bmod 10.$$

Para $n = 2$ se tiene $2^{2^2} = 2^4 = 16 \equiv 6 \bmod 10$, que es justamente lo que queríamos demostrar.

Como $2^{2^{n+1}} = 2^{2\times 2^n} = (2^{2^n})^2$, ahora podemos aplicar la hipótesis de inducción de modo que $(2^{2^n})^2 \equiv 6^2 \bmod 10$. Terminamos la demostración observando que $6^2 = 36 \equiv 6 \bmod 10$.

Problema 3. (1.5 puntos) Sea el grafo simple $G = (V, E)$ donde $V = \{1, 2, 3\}$ y $E = \{\{1, 2\}, \{2, 3\}\}$.

(0.5 puntos) Construir una matriz de adyacencias del grafo producto $G \times G$.

Ordenamos los vértices de la siguiente manera:

$$V = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)\}.$$

Entonces la matriz es:

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

(0.5 puntos) Determinar el grado de cada vértice de $G \times G$.

Con la ordenación de los vértices anteriormente establecida se tiene que la sucesión de los grados es:

$$2, 3, 2, 3, 4, 3, 2, 3, 2.$$

(0.5 puntos) Determinar si G es conexo, determinar si $G \times G$ es conexo, determinar si $G \times G$ es Euleriano.

El grafo G es conexo, pues todos los vértices están unidos entre sí por un camino.

También lo es $G \times G$. Los vértices que tienen su primera (respectivamente su segunda) coordenada igual están unidos por un camino en G y por tanto en $G \times G$. Si ambas coordenadas son distintas, basta elegir un camino que una el vértice primero con el que tiene su primera coordenada igual y la segunda igual a la del vértice segundo. Cuando las coordenadas segundas son iguales, se usa el razonamiento escrito anteriormente.

Aunque $G \times G$ es conexo no todos sus vértices son de grado par, por lo que no es Euleriano.

Problema 4. (2 puntos) Sea $\mathbb{Z}_{131} = \{\overline{0}, \overline{1}, \ldots, \overline{129}, \overline{130}\}$. Considera la función siguiente:

$$\begin{array}{rccc} f: & \mathbb{Z}_{131} & \to & \mathbb{Z}_{131} \\ & \overline{a} & \mapsto & \overline{7a+9} \end{array}$$

(0.5 puntos) Determinar $f(\overline{81})$ y $f(\overline{111})$.

Simplemente evaluando se tiene $f(\overline{81}) = \overline{52}$ y $f(\overline{111}) = \overline{0}$.

(0.75 puntos) Determinar si f es inyectiva.

Si $7a + 9 \equiv 7b + 9 \bmod 131$ entonces $7(a - b) \equiv 0 \bmod 131$. Como $\mathrm{mcd}(7, 131) = 1$ entonces multiplicando por el inverso de 7 módulo 131 obtenemos $a - b \equiv 0 \bmod 131$ de modo que $\overline{a} = \overline{b}$. Por tanto es inyectiva.

(0.75 puntos) Determinar si f es sobreyectiva.

Como $\mathrm{mcd}(7, 131) = 1$ entonces dado $\overline{a} \in \mathbb{Z}_{131}$ la congruencia lineal $7x + 9 \equiv a \bmod 131$ tiene solución, de modo que f es sobreyectiva.

Problema 5. (2 puntos) Supongamos que tenemos un algoritmo de complejidad lineal, $Borrar(L, i)$, que tiene como entrada una lista L y un entero i y borra de L el término L_i. Supongamos también que podemos hacer asignaciones de listas $L := M$.

(1.5 puntos) Construir un algoritmo que tenga como entrada una lista L y como salida la lista donde se han borrado todos los elementos de L que son ´múltiplos de 6.

Entrada: $L_1, \ldots, L_n$
$i := n$
while $i \geq 1$
    if $L_i \bmod 6 = 0$ then $L := Borrar(L, i)$
    $i := i - 1$
Salida: L

(0.5 puntos) Estudiar su complejidad.

Como $Borrar$ es una algoritmo de complejidad lineal (y lo mismo asignar listas) al que se le llama n veces, entonces la complejidad de nuestro algoritmo es cuadrática.

Problema 6. (1.5 puntos) Sea A el conjunto de los números naturales escritos en el sistema de numeración de base 2 y caracterizados por la propiedad

de ser menores o iguales que 1000, donde 1000 está escrito también en base 2. En A definimos la siguiente relación: para $a, b \in A$ se tiene que aRb si $a + b \in A$.

(0.5 puntos) Escribir A y determinar si la relación es reflexiva, simétrica y transitiva.

Como 1000 en base 2 es 8 en base 10 entonces

$$A = \{1, 10, 11, 100, 101, 110, 111, 1000\}.$$

La relación no es reflexiva, por ejemplo $1000 + 1000 \notin A$.

La relación es simétrica puesto que $a + b = b + a$.

La relación no es transitiva, por ejemplo $110 + 1 = 111 \in A$, $1 + 110 = 111 \in A$ y sin embargo $110 + 110 = 1100 \notin A$

(0.5 puntos) Escribir una matriz de la relación.

Con el orden establecido se tiene:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

(0.5 puntos) Escribir su clausura reflexiva, su clausura simétrica y su clausura transitiva.

La clausura reflexiva consiste en añadir los 3 unos que faltan en la diagonal.

La clausura simétrica coincide con la relación pues cumple esta propiedad.

Como la matriz al cuadrado tiene todos sus elementos no nulos salvo la última fila y columna (donde tampoco hay elementos no nulos en las potencias sucesivas), entonces la clausura transitiva es $(A - \{1000\}) \times (A - \{1000\})$.

Bibliografía

[B] Biggs, N. *Matemática Discreta*, Vicens Vives 1994.

[CBVB] Criado, R., Bujosa, A., Vega, C., Banerjee, R. *Fundamentos Matemáticos I*, Centro de Estudios Ramón Areces, 1998.

[F] Fernández-Sáez Vacas, *Fundamentos de Informática*, Alianza-Informática, 1987.

[GLP] García, C., López, J. M., Puigjaner, D., *Matemática Discreta: Problemas y ejercicios resueltos*. Prentice Practica, Prentice Hall, Madrid, 2002.

[G] Grimaldi, R.P. *Matemática discreta y combinatoria*. Addison Wesley, 1989.

[M] Muñoz, R. *Tres problemas clásicos y complejidad*. SUMA 47, páginas 29-36, 2004.

[KBR] Kolman, B., Busby, R. C., Ross, S. *Estructuras de matemáticas discretas para la computación*, Prentice Hall, México, 1995.

[R] Rosen, K. H. *Discrete Mathematics and its applications*, McGraw-Hill 1995 y 1999. Manual para el curso. (Edición en español, 2004)

[R2] Rosen, K.H. *Students solutions guide for Discrete Mathematics and its applications*, McGraw-Hill 1999.

[R3] Rosen, K.H. *Exploring Discrete Mathematics with Maple*, McGraw-Hill 1997.

[SL] Lipschutz, S, Lipson, M. *2000 problemas resueltos de Matemática Discreta*, Mc Graw Hill, 2004.

[T] Truss, J.K. *Discrete Mathematics for computer scientists*, International Computer Science Series, 1991.

[W] R. W. Hamming, *Coding and information theory*, Prentice Hall 1980.