a) Completar la relación binaria $R$ para que sea una relación de orden total.
b) Completar la relación binaria $R$ para que sea reflexiva y transitiva, pero no sea simétrica y antisimétrica.

■

*Ejercicio 8.2.* Sea $A$, el conjunto formado por los tres últimos dígitos distintos de tu DNI, calcular el producto cartesiano $A \times A$ y una relación de orden en éste.

■

*Ejercicio 8.3.* Dados los siguientes diagramas:

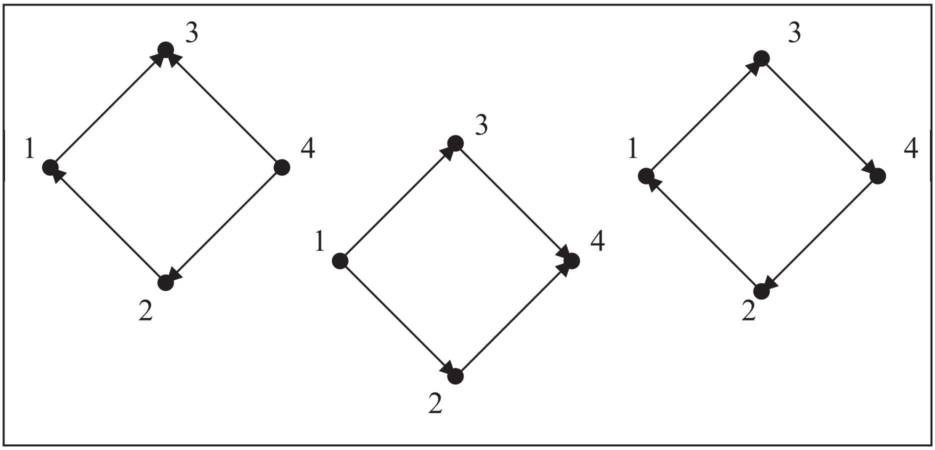



Ilustración 8.2. Ejercicio 8.3.

Determinar cuáles de ellos son conjuntos ordenados y dibujar los diagramas de orden correspondientes con Mathematica.

■

*Ejercicio 8.4.* Calcular el diagrama de orden de $\mathcal{P}(\{a, b, c\})$.

■

*Ejercicio 8.5.* En el conjunto $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ se establece la relación binaria

$$R = \{(a, b) : a + b < 12\}$$

Calcular explícitamente el conjunto $R$ y estudiar las propiedades que satisface dicha relación. ¿Es una relación de equivalencia? ¿Y de orden? En caso contrario, añadir los pares necesarios para que sea una relación de orden. Y comprobar que realmente lo es con Mathematica.

■

*Ejercicio 8.6.* Sean $A$ = {dígitos distintos de tu DNI} y $B$ = {dígitos distintos en tu fecha de nacimiento, distintos de cero}. En el conjunto $X = \{a/b$ tal que $a \in A$ y $b \in B\}$ se define la relación binaria: $a/b \; R \; c/d$ si y sólo si $ad = bc$. Calcular $R$ y comprobar que es una relación de equivalencia.

■

*Ejercicio 8.7.* Sea $A = \{n \in \mathbb{N} \mid n < 1000\}$, con la relación de equivalencia:

$$R = \{(a, b) \in A \times A : (a - b) \text{ es un múltiplo de } 5\}$$

Calcular la clase de equivalencia de un elemento cualquiera de $A$, y el conjunto cociente. Comprobar que el conjunto cociente obtenido es una partición del conjunto $A$.

■

*Ejercicio 8.8.* En el conjunto $A = \{1, 2, 3, 4\}$ se establece la relación binaria

$$R = \{(1, 1), (2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4)\}$$

Justificar que es una relación de equivalencia y calcular el conjunto cociente. Comprobar que dicho conjunto da una partición de $A$.

■

*Ejercicio 8 .9.* En el conjunto $A = \{1, 2, 3, 4\}$ se establece la relación binaria

$$R = \{(1, 1), (2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4)\}$$

Justificar que es una relación de equivalencia y calcular el conjunto cociente.

■

*Ejercicio 8.10.* Comprobar si son verdaderas o falsas las siguientes afirmaciones, en el caso de ser falsas demostrar con un contraejemplo usando Mathematica:

a) Una relación binaria $R$ sobre un conjunto $A$ define una correspondencia de $A$ en $A$.
b) Una relación de equivalencia o de orden $R$ sobre un conjunto $A$ define una aplicación de $A$ en $A$.
c) El grafo de una aplicación de $A$ en $A$ es una relación binaria en $A$.
d) Toda relación de equivalencia $R$ en $A$ verifica $A/R = \mathcal{P}(A)$.
e) $\mathcal{P}(A)$ es una partición de $A$.

■

*Ejercicio 8.11.* Dada una aplicación $f: A \to B$, la relación binaria $R_f = \{(a, b) \in A \times A : f(a) = f(b)\}$ es una relación de equivalencia. Comprobarlo para la aplicación $f: A \to B$, con $A = \{n \in \mathbb{N} : n < 1000\}$, $B = \{a, b, c\}$ y definida por

$$f(n) = \begin{cases} a & \text{si } n \text{ es par} \\ b & \text{si } n \text{ es impar} \end{cases}$$

■

*Ejercicio 8.12.* ***Descomposición Canónica.*** Dada una aplicación $f: A \to B$, entonces

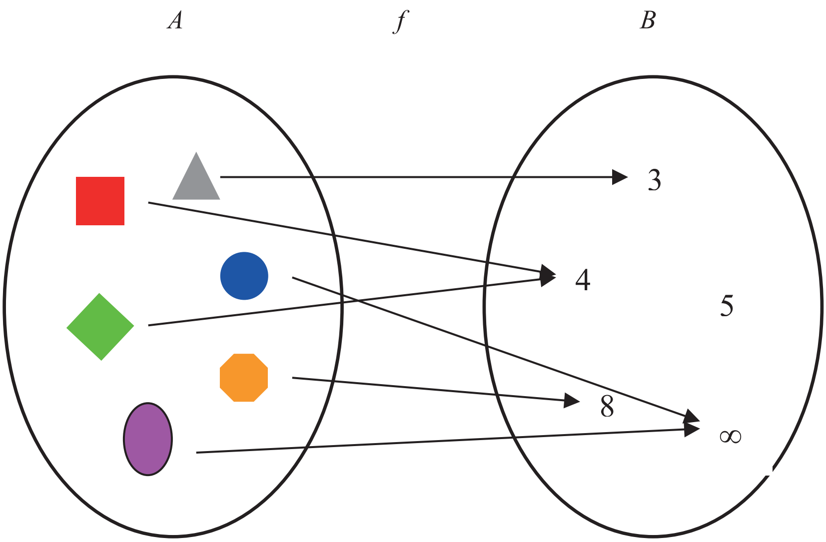
a) Existe una aplicación sobreyectiva $p: A \to A/R_f$ definida por $p(a) = \overline{a}$, donde $R_f$ es la relación de equivalencia del ejercicio anterior.
b) Existe una aplicación biyectiva $\overline{f}: A/R_f \to Im(f)$ definida por $\overline{f}(\overline{a}) = f(a)$.
c) Existe una aplicación inyectiva $i: Im(f) \to B$ definida por $i(b) = b$.

Además $i \circ \bar{f} \circ p = f$.

Comprobarlo con la aplicación del ejercicio anterior.

■

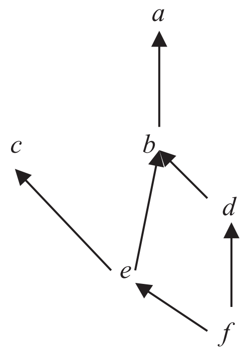
*Ejercicio 8.13.* Se considera la siguiente aplicación:




Definirla con Mathematica y calcular su descomposición canónica como en el ejercicio anterior.

■

*Ejercicio 8.14.* Sea $X = \{a, b, c, d, e, f\}$ junto con la ordenación dada por el siguiente diagrama:




Sea $Y = \{b, f, e\}$ y $Z = \{e, b, c, a\}$ subconjuntos de $X$. Se pide:

a) Cotas superiores e inferiores de $Y$ y de $Z$. ¿Existen supremo e ínfimo?
b) Máximos y mínimos de $Y$ y $Z$.

c) Elementos maximales y minimales de $X$, $Y$ y $Z$.
d) Representar con Mathematica el diagrama de orden de $X$, $Y$ y $Z$.

■

*Ejercicio 8.15.* Sea $X = \{x \in \mathbb{Z} \,/\, 0 < x < 100\}$ y $A = (\mathrm{D}(40) \cap X) - \{1\}$, con D(40) es el conjunto de los divisores de 40. Si existen, calcular al menos dos cotas superiores e inferiores de $A$ en $X$, supremo, ínfimo, máximo, mínimo, elementos maximales y minimales de $A$.

■

*Ejercicio 8.16.* Calcular el diagrama de Hasse del conjunto de los divisores positivos de 60 con la relación de orden divisibilidad y hallar los elementos notables de los subconjuntos $A = \{3, 4, 5, 12, 15, 20\}$ y $B = \{2, 4, 5, 10, 20, 30\}$.

■

*Ejercicio 8.17.* Crear rutinas que determinen si una relación de orden es total o parcial. Aplicarlas a los ejemplos de órdenes que aparecen en el capítulo.

■

*Ejercicio 8.18.** Uno de los órdenes más utilizados y habituales es el lexicográfico (orden alfabético) usado para ordenar palabras. El conjunto de las palabras, con este orden, es totalmente ordenado y está bien ordenado. Si suponemos que empleamos sólo caracteres de la A a la Z, la palabra "A" sería la más pequeña, seguida por: "AA", "AAA",…, "AAAAA…A",…, "AB",…, "AZ",…, "B", "BA",… Los lenguajes de programación, en particular Mathematica, por lo general, son capaces de comparar "Strings" usando este orden. Si quisiéramos formalizar este orden, identificaríamos cada palabra con una $n$-upla formada por los n caracteres que la componen, por ejemplo, "HOLA" se identificaría con (H, O, L, A). Definimos por $\mathbf{A} = \{A, B, C, D, \ldots, Z\}$ al conjunto de todas las letras del alfabeto donde tenemos el orden:

$$A < B < C < \ldots < Z.$$

El conjunto de todas las palabras posibles ("diccionario") que podemos formar sería:

$$\mathbf{D} = \bigcup_{n \in N - \{0\}} \mathbf{A} \times \mathbf{A} \times \overset{n\text{ - veces}}{\ldots} \times \mathbf{A}\,.$$

Para $(X_1, X_2, \ldots, X_n), (Y_1, Y_2, \ldots, Y_m) \in \mathbf{D}$ (esto es, $X_i, Y_j \in \mathbf{A}$ para cada $0 < i < n$, $0 < j < m$), el orden lexicográfico vendría dado por:

$$(X_1, X_2, \ldots, X_n) < (Y_1, Y_2, \ldots, Y_m) \Leftrightarrow \text{existe } k \leq m \text{ tal que } X_i = \mathrm{Y}_i \text{ para } 0 < i < k \text{ y } X_k < Y_k.$$

Consideremos para el ejercicio como alfabeto el conjunto $\{\alpha, \beta, \gamma, \delta, \varepsilon\}$ con el orden,

$$\alpha < \beta < \gamma < \delta < \varepsilon.$$

Escribir un programa que ordene de menor a mayor un conjunto cualquiera de palabras formadas a partir del alfabeto dado.

■