4

ARITMÉTICA MODULAR

4.1 CONGRUENCIAS

Una de las más familiares particiones de un conjunto es la partición de $\mathbb{Z}$ en enteros pares y enteros impares. Es decir $\mathbb{Z}$ es la unión disjunta del conjunto de números pares y el de los números impares. Es claro que dos números a, b tienen la misma paridad si $a - b$ es divisible por 2. Para expresar este hecho es usual la notación

$$a \equiv b \pmod{2}$$

y se dice que a es *congruente* a b *módulo* 2. Es decir, a es congruente a b módulo 2 si y solo si a y b son ambos pares o ambos impares si y solo si $2|a - b$.

Claramente esta definición se puede extender a cualquier entero positivo m.

Definición 4.1.1. Sean a y b enteros y m un entero positivo. Diremos que a es *congruente* a b *módulo* m, y escribimos

$$a \equiv b \pmod{m}$$

si $a - b$ es divisible por m, o, escrito de otra forma, $m|a - b$.

Observar que $a \equiv 0 \pmod{m}$ si y sólo si $m|a$ y que $a \equiv b \pmod{m}$ si y sólo si $a - b \equiv 0 \pmod{m}$.

Notación. $a \equiv b \pmod{m}$ también lo denotamos $a \equiv b\ (m)$.

Ejemplo. Algunos ejemplos numéricos:

- $7 \equiv 3\ (2)$, pues $2|7 - 3 = 4$.
- $17 \equiv 8\ (3)$, pues $3|17 - 8 = 9$.
- $8 \equiv 17\ (3)$, pues $3|8 - 17 = -9$.
- $35 \equiv 13\ (11)$, pues $11|35 - 13 = 22 = 2 \cdot 11$.

Proposición 4.1.2. *Sean* a *entero y* m *un entero positivo. Sea* r *el resto de dividir* a *por* m*, entonces*

$$a \equiv r \pmod{m}.$$

Demostración. $a = mq + r$ con $0 \leqslant r < m$. Luego,

$$a - r = mq \quad \Rightarrow \quad m|a-r \quad \Rightarrow \quad a \equiv r \pmod{m}.$$

□

Es fácil verificar que la congruencia módulo m es una relación de equivalencia.

Proposición 4.1.3. *Sea* m *entero positivo y* $x, y, z \in \mathbb{Z}$*. Entonces la relación de congruencia es*

a) reflexiva, es decir $x \equiv x \pmod{m}$,

b) simétrica, es decir si $x \equiv y \pmod{m}$*, entonces* $y \equiv x \pmod{m}$*, y*

c) transitiva, es decir si $x \equiv y \pmod{m}$ *e* $y \equiv z \pmod{m}$*, entonces* $x \equiv z \pmod{m}$.

Demostración. Las tres propiedades son sencillas de demostrar y se deducen de propiedades de "divide a".

a) $m|x - x = 0$ y por lo tanto $x \equiv x \pmod{m}$.

b) $x \equiv y\ (m) \Rightarrow m|x-y \Rightarrow m|-(x-y) \Rightarrow m|y-x \Rightarrow y \equiv x\ (m)$.

c) $x \equiv y\ (m) \wedge y \equiv z\ (m) \quad \Rightarrow \quad m|x-y \wedge m|y-z \quad \Rightarrow$

$m|(x-y)+(y-z) = x-z \quad \Rightarrow \quad x \equiv z\ (m)$.

□ □

Observación. Dado un conjunto X una relación que cumple las propiedades de ser reflexiva, simétrica y transitiva es llamada una *relación de equivalencia*.

Ejemplos de relación de equivalencia son la igualdad y el paralelismo de rectas y ahora vimos la relación de congruencia. Veremos otra relación de equivalencia en el capítulo de grafos.

Por la proposición 4.1.2 y por ser "congruencia módulo m" una relación de equivalencia se deduce el siguiente resultado.

Proposición 4.1.4. *Sean* a *y* b *enteros y* m *un entero positivo. Entonces* $a \equiv b \pmod{m}$ *si y sólo si* a *y* b *tienen el mismo resto en la división por* m.

Demostración. ($\Rightarrow$) Si $a = mh + r$ y $b = mk + s$, con $0 \leqslant r, s < m$, entonces por la proposición 4.1.2 tenemos que

$$a \equiv r \pmod{m} \quad \wedge \quad b \equiv s \pmod{m}.$$

Por la propiedad transitiva de "congruencia módulo m", $r \equiv s \pmod{m}$, luego $m|r - s$. Podemos suponer, sin perdida de generalidad, que $s \leqslant r$, luego $0 \leqslant r - s < m$, y por lo visto recién $m|r - s$, lo cual implica que $r - s = 0$, es decir $r = s$.

($\Leftarrow$) Si a y b tienen el mismo resto en la división por m, entonces $a = mh + r$ y $b = mk + r$, luego $a - b = m(h - k)$ que es divisible por m. □

Así como antes podíamos separar $\mathbb{Z}$ en los números pares e impares, la propiedad anterior nos permite expresar $\mathbb{Z}$ como una unión disjunta de m subconjuntos. Es decir sea $\mathbb{Z}_{[r]} = \{x \in \mathbb{Z} : \text{el resto de dividir } x \text{ por } m \text{ es } r\}$, o más formalmente

$$\mathbb{Z}_{[r]} = \{x \in \mathbb{Z} : \ x \equiv r \pmod{m}\}, \qquad 0 \leqslant r < m.$$

Entonces,

$$\mathbb{Z} = \mathbb{Z}_{[0]} \cup \mathbb{Z}_{[1]} \cup \cdots \cup \mathbb{Z}_{[m-1]}.$$

La utilidad de las congruencias reside principalmente en el hecho de que son compatibles con las operaciones aritméticas. Específicamente, tenemos el siguiente teorema.

Teorema 4.1.5. *Sea m un entero positivo y sean x_1, x_2, y_1, y_2 enteros tales que*

$$x_1 \equiv x_2 \pmod{m}, \qquad y_1 \equiv y_2 \pmod{m}.$$

Entonces

a) $x_1 + y_1 \equiv x_2 + y_2 \pmod{m}$,

b) $x_1 y_1 \equiv x_2 y_2 \pmod{m}$,

c) Si $x \equiv y \pmod{m}$ *y* $j \in \mathbb{N}$, *entonces* $x^j \equiv y^j \pmod{m}$.

Demostración. La hipótesis nos dice que

$$m|x_1 - x_2 \qquad \wedge \qquad m|y_1 - y_2.$$

a) Como $m|x_1 - x_2$ y $m|y_1 - y_2$, entonces

$$m|(x_1 - x_2) + (y_1 - y_2) = (x_1 + y_1) - (x_2 + y_2).$$

Es decir, $x_1 + y_1 \equiv x_2 + y_2 \pmod{m}$.

b) Aquí tenemos,

$$\begin{aligned} m|x_1 - x_2 \wedge m|y_1 - y_2 &\Rightarrow m|(x_1 - x_2)y_1 \wedge m|x_2(y_1 - y_2) \\ &\Rightarrow m|(x_1 - x_2)y_1 + x_2(y_1 - y_2) \\ &\Rightarrow m|x_1y_1 - x_2y_1 + x_2y_1 - x_2y_2 \\ &\Rightarrow m|x_1y_1 - x_2y_2. \end{aligned}$$

Luego $x_1y_1 \equiv x_2y_2 \pmod{m}$.

c) Lo haremos por inducción sobre j.

Es claro que si $j = 1$ el resultado es verdadero. Supongamos ahora que el resultado vale para $j-1$, es decir que si $x \equiv y \pmod{m}$, entonces

$$x^{j-1} \equiv y^{j-1} \pmod{m}.$$

Como $x \equiv y \pmod{m}$, por *b)* tenemos que

$$x^{j-1}x \equiv y^{j-1}y \pmod{m},$$

es decir

$$x^j \equiv y^j \pmod{m}.$$

□

Veremos ahora una aplicación interesante de las congruencias y sus propiedades.

Proposición 4.1.6. *Sea* $(x_nx_{n-1}\dots x_0)_{10}$ *la representación del entero positivo* x *en base* 10*, entonces*

$$x \equiv x_0 + x_1 + \cdots + x_n \pmod{9}$$

Demostración. Observemos primero que como $10 \equiv 1 \pmod{9}$, entonces $10^k \equiv 1^k \equiv 1 \pmod{9}$. Esto es debido al teorema 4.1.5 *c)*

Por la definición de representación en base 10, tenemos que

$$x = x_0 + 10x_1 + \cdots + 10^nx_n,$$

por el párrafo anterior y teorema 4.1.5 *b)* obtenemos que $x_k10^k \equiv x_k \pmod{9}$ y por teorema 4.1.5 *a)* se deduce que $x \equiv x_0 + x_1 + \cdots + x_n \pmod{9}$. □

Corolario 4.1.7. *ea* $(x_nx_{n-1}\dots x_0)_{10}$ *la representación del entero positivo* x *en base* 10*, entonces* x *es divisible por* 9 *si y solo si* $x_0 + x_1 + \cdots + x_n$ *es divisible por* 9*.*

Demostración.

$$\begin{aligned} 9|x &\Leftrightarrow 0 \equiv x \pmod 9 \\ &\Leftrightarrow 0 \equiv x_0 + x_1 + \cdots + x_n \pmod 9 \quad \text{(prop. 4.1.6 y transitividad)} \\ &\Leftrightarrow 9|x_0 + x_1 + \cdots + x_n. \end{aligned}$$

□

El procedimiento anterior a veces es llamado *regla del nueve.*

Ejemplo. Usando la regla del 9 es sencillo verificar que $X = 3475682568$ es divisible por 9, pues

$$3 + 4 + 7 + 5 + 6 + 8 + 2 + 5 + 6 + 8 = 54$$

y como $5 + 4 = 9$ es divisible por 9, también lo es 54 y, luego, también lo es x.

También la proposición 4.1.6 es útil para verificar si una multiplicación larga es incorrecta.

Ejemplo. Verifiquemos el siguiente cálculo

$$54\,321 \cdot 98\,765 = 5\,363\,013\,565.$$

Demostración. En la notación de la proposición 4.1.6 escribamos $\Sigma\, x$ en vez de

$$\sum_{i=0}^{n} x_i = x_0 + x_1 + \cdots + x_n.$$

Hemos visto que $\Sigma\, x \equiv x \pmod 9$. Por la parte *b)* del teorema 4.1.5 tenemos

$$\Sigma\, x \cdot \Sigma\, y \equiv xy \pmod 9,$$

y por consiguiente si $xy = z$ debemos tener $\Sigma\, x \cdot \Sigma\, y \equiv \Sigma\, z \pmod 9$. En el cálculo que se tiene en el ejemplo

$$\Sigma\, 54\,321 = 15, \quad \Sigma\, 98\,765 = 35, \qquad \Sigma\, 5\,363\,013\,565 = 37,$$

y

$$\Sigma\, 15 = 6, \qquad \Sigma\, 35 = 8, \quad \Sigma\, 37 = 10.$$

Puesto que $6 \cdot 8$ no es congruente a $10 \pmod 9$ se sigue que $15 \cdot 35$ no es congruente a $37 \pmod 9$ y que $54\,321 \cdot 98\,765$ no es congruente a $5\,363\,013\,565 \pmod 9$. En consecuencia el cálculo está errado. □

También a este verificación suele llamársela la regla del 9. Observar que esta última aplicación solo sirve para comprobar que una multiplicación es incorrecta, y no es útil para verificar que una multiplicación es correcta, pues $xy \equiv z \pmod 9$ no garantiza que $xy = z$.

§ *Ejercicios*

1) Sin hacer ninguna "multiplicación larga" probar que
   a) $1\,234\,567 \cdot 90\,123 \equiv 1 \pmod{10}$
   b) $2\,468 \cdot 13\,579 \equiv -3 \pmod{25}$
2) Usar la regla del nueve para verificar que dos de las siguientes ecuaciones son falsas. ¿Qué se puede decir de la otra ecuación?
   a) $5\,783 \cdot 40\,162 = 233\,256\,846$,
   b) $9\,787 \cdot 1\,258 = 12\,342\,046$,
   c) $8\,901 \cdot 5\,743 = 52\,018\,443$.
3) Encontrar el resto de dividir 3^{15} por 17 y el de dividir 15^{81} por 13.
4) Sea $(x_n x_{n-1} \dots x_0)_{10}$ la representación en base 10 de un entero positivo x.
   a) Probar que
   $$x \equiv x_0 - x_1 + x_2 + \cdots + (-1)^n x_n \pmod{11}.$$
   b) Enuncie la *regla del* 11.
   c) Probar que 1 213 141 516 171 819 es divisible por 11.
5) ¿Cuál es el último dígito de la representación en base 10 de 7^{93}.
6) Usar que $1\,001 = 7 \cdot 11 \cdot 13$ para construir una prueba para la división simultanea por los números 7, 11 y 13, similar a la prueba del 9.
7) Si m coprimo con n, entonces $ma \equiv mb \pmod{n}$ si y solo si $a \equiv b \pmod{m}$.

4.2 ECUACIÓN LINEAL DE CONGRUENCIA

Se trata primero de estudiar en general el problema de resolución de la ecuación en x

$$ax \equiv b \pmod{m}. \tag{4.2.1}$$

Es fácil ver que el problema no admite siempre solución, por ejemplo $2x \equiv 3 \pmod{2}$ no posee ninguna solución en $\mathbb{Z}$, pues cualquiera se $k \in \mathbb{Z}$, $2k - 3$ es impar, luego no es divisible por 2.

Notemos además que si x_0 es solución de la ecuación (4.2.1), también lo es $x_0 + km$ de manera que si la ecuación posee una solución, posee infinitas soluciones.

Ejemplo. La solución general de la ecuación $3x \equiv 7 \pmod{11}$ es $6 + k7$ con $k \in \mathbb{Z}$.

Demostración. Si probamos con los enteros x tal que $0 \leqslant x < 11$, veremos que la ecuación admite una única solución, a saber $x = 6$. Otras soluciones se obtienen tomando $6 + 11k$. Por otra parte si u es también solución de la ecuación, se tiene $3u \equiv 3 \cdot 6 \pmod{11}$, por lo tanto $3(u-6)$ es múltiplo de 11. Como 11 no divide a 3 se tiene que $11|(u-6)$, o sea $u = 6 + 11k$. □

Analicemos ahora la situación general de la ecuación $ax \equiv b \pmod{m}$. Si $\mathrm{mcd}(a, m) = 1$, entonces sabemos que existen enteros r y s tales que $1 = ra + sm$ y por lo tanto $b = (rb)a + (sb)m$, o sea que

$$a(rb) \equiv b \pmod{m},$$

es decir rb es solución de la ecuación. Veremos que el caso general se resuelve de forma análoga.

Teorema 4.2.1. *Sean a, b números enteros y m un entero positivo y denotemos $d = \mathrm{mcd}(a, m)$. La ecuación*

$$ax \equiv b \pmod{m} \tag{4.2.2}$$

admite solución si y sólo si $d|b$, y en este caso dada x_0 una solución, todas las soluciones son de la forma

$$x = x_0 + kn, \qquad \text{con } k \in \mathbb{Z} \text{ y } n = \frac{m}{d}$$

Demostración. Como $d = \mathrm{mcd}(a, m)$, existen $r, s \in \mathbb{Z}$ tales que

$$d = ra + sm.$$

Si $d|b$, entonces existe $h \in \mathbb{Z}$ tal que $b = dh$. Si multiplicamos por h la ecuación de arriba obtenemos

$$dh = (rh)a + (sh)m.$$

Luego $a(rh) \equiv a(rh) + (sh)m \equiv dh \equiv b \pmod{m}$, y por lo tanto rh es solución de la ecuación lineal de congruencia.

Por otro lado si $ax \equiv b \pmod{m}$, entonces $ax - b = km$ para algún k, o sea

$$b = ax + (-k)m$$

de la cual se sigue que si $d|a$ y $d|m$, entonces $d|b$ y por lo tanto $\mcd(a,m)|b$.

Por lo tanto hemos demostrado que la condición necesaria y suficiente para que la ecuación $ax \equiv b \pmod{m}$ admita una solución es que $\mcd(a,m)|b$.

En el caso que $d|b$ veamos ahora cuales son todas las soluciones posibles de la ecuación (4.2.2). Sean x_1, x_2 soluciones, es decir

$$\begin{aligned} ax_1 &\equiv b \pmod{m} \\ ax_2 &\equiv b \pmod{m}, \end{aligned}$$

entonces, restando miembro a miembro, obtenemos

$$ax_1 - ax_2 \equiv b - b \equiv 0 \pmod{m}.$$

Es decir, x_1, x_2 son soluciones de la ecuación (4.2.2) si y sólo si $y = x_1 - x_2$ es solución de la ecuación lineal de congruencia

$$ay \equiv 0 \pmod{m}.$$

Si $\mcd(a,m) = 1$ es claro que la ecuación $ay \equiv 0 \pmod{m}$ tiene como solución todos los y tales que $m|ay$. Como m y a son coprimos, las soluciones son todos los y tal $m|y$, es decir todos los múltiplos de m.

Si $\mcd(a,m) = d > 1$, la ecuación $ay \equiv 0 \pmod{m}$ tiene como solución todos los y tales que $ay = mk$ para algún k. Si dividimos por d, podemos decir que las soluciones son todos los y tales que $(a/d)y = (m/d)k$, es decir todos los y tal que $(m/d)|(a/d)y$. Como m/d y a/d son coprimos, las soluciones son todos los múltiplos de m/d.

Sean x_0 y x tal que $ax_0 \equiv b \pmod{m}$ y $ax \equiv b \pmod{m}$, entonces $a(x_0 - x) \equiv 0 \pmod{m}$ y por lo tanto $x_0 - x = kn$ para algún k. Es decir, cualquier x que es solución lineal de congruencia es de la forma $x_0 = x + kn$ para algún k. □

De las demostraciones podemos obtener un método general para encontrar soluciones de la ecuación lineal de congruencia

$$ax \equiv b \pmod{m}.$$

con $\mcd(a,m)|b$

a) Encontrar, usando el algoritmo de Euclides, r, s tales que

$$d = \mcd(a,m) = ra + sm. \tag{4.2.3}$$

b) Como $d|b$, tenemos que $b = td$, para algún t entero, y multiplicamos la ecuación (4.2.3) por t:

$$dt = (rt)a + (st)m.$$

c) $b = dt = (rt)a + (st)m \equiv (rt)a \pmod{m}$.

Luego $x_0 = rt$ es solución de la ecuación lineal de congruencia.

d) Toda solución de la ecuación lineal de congruencia es $x = x_0 + k(m/d)$ con $k \in \mathbb{Z}$.

Observemos que, en las hipótesis del teorema, si $\text{mcd}(a, m) = 1$, entonces siempre existen soluciones a la ecuación $ax \equiv b \pmod{m}$ y todas las soluciones son de la forma $x_0 + km$, donde x_0 es una solución particular. Más aún, debido a esto, hay una única solución x, con $0 \leqslant x < m$.

Ejemplo. Hallemos las soluciones de la ecuación $13x \equiv 7 \pmod{15}$ con $0 \leqslant x < 15$.

Demostración. Hagamos, paso a paso, el procedimiento explicado anteriormente.

a) Usando el algoritmo de Euclides obtenemos el $\text{mcd}(13, 15)$.

$$\begin{aligned} 15 &= 13 \cdot 1 + 2 \\ 13 &= 2 \cdot 6 + 1 \\ 2 &= 1 \cdot 2 + 0 \end{aligned}$$

Luego $1 = \text{mcd}(13, 15)$. Como 1 divide a cualquier número, en este caso la ecuación tiene solución. Del algoritmo de Euclides deducimos

$$\begin{aligned} 1 &= 13 - 2 \cdot 6 \\ &= 13 - (15 - 13) \cdot 6 \\ &= 13 \cdot 7 - 15 \cdot 6. \end{aligned}$$

Es decir

$$1 = 13 \cdot 7 - 15 \cdot 6. \tag{4.2.4}$$

b) Multiplicando la ecuación (4.2.4) por 7 obtenemos

$$7 = 13 \cdot 49 - 15 \cdot 42.$$

c) Luego $13 \cdot 49 \equiv 7 \pmod{15}$, es decir 49 es solución de la ecuación.

d) Todas las soluciones son de la forma $x = 49 + 15k$.

Debemos ver ahora cuales soluciones x cumplen $0 \leqslant x < 15$. La forma más sencilla de hacerlo es buscando por tanteo: $49 + 15(-1) = 34$, $49 + 15(-2) = 19$, , $49 + 15(-3) = 4$, $49 + 15(-4) = -11$. Es decir la solución que buscamos es $x = 4$. □

Ejemplo. Hallemos las soluciones de la ecuación $42x \equiv 50 \pmod{76}$ con $0 \leqslant x < 76$.

Demostración. Como antes, hagamos paso a paso el procedimiento explicado anteriormente.

a) Usando el algoritmo de Euclides obtenemos el $\mathrm{mcd}(42, 76)$.

$$\begin{aligned} 76 &= 42 \cdot 1 + 34 \\ 42 &= 34 \cdot 1 + 8 \\ 34 &= 8 \cdot 4 + 2 \\ 8 &= 2 \cdot 4 + 0. \end{aligned}$$

Luego $2 = \mathrm{mcd}(42, 76)$. Como $2|50$ la ecuación tiene solución. Del algoritmo de Euclides deducimos

$$\begin{aligned} 2 &= 34 - 8 \cdot 4 \\ &= 34 - (42 - 34) \cdot 4 = 34 \cdot 5 - 42 \cdot 4 \\ &= (76 - 42) \cdot 5 - 42 \cdot 4 \\ &= 76 \cdot 5 - 42 \cdot 9. \end{aligned}$$

Es decir

$$2 = (-9) \cdot 42 + 5 \cdot 76.$$

b) $50 = 2 \cdot 25$ y tenemos que

$$\begin{aligned} 50 &= (-9 \cdot 25) \cdot 42 + (5 \cdot 25) \cdot 76 \\ 50 &= (-225) \cdot 42 + 125 \cdot 76 \end{aligned}$$

c) Luego $x_0 = -225$ es una solución de la ecuación lineal de congruencia.

d) Todas las soluciones son de la forma $-225 + (76/2)k$, es decir $x = -225 + 38k$.

Debemos ver ahora cuales soluciones x cumplen $0 \leqslant x < 76$. Como en el caso anterior podemos hacer esto por tanteo, pero aquí la forma más sencilla de hacerlo es escribir las inecuaciones

$$\begin{aligned} 0 \leqslant -225 + 38k &< 76 \\ 225 \leqslant 38k &< 76 + 225 = 301 \\ 225/38 \leqslant k &\leqslant 301/38 \\ 5.9 \leqslant k &\leqslant 7.9 \end{aligned}$$

Luego $k = 6$ o $k = 7$ y entonces $x_1 = -225 + 38 \cdot 6 = 3$ y $x_2 = -225 + 38 \cdot 7 = 41$ son las soluciones que buscamos. □

Operativamente es útil el siguiente resultado.

Proposición 4.2.2. *Sean* a, b *números enteros y* m *un entero positivo, denotemos* $d = \mathrm{mcd}(a, m)$ *y supongamos que* $d|b$. *Entonces las soluciones de la ecuación*

$$ax \equiv b \pmod{m}$$

son las misma que las de la ecuación

$$\frac{a}{d}x \equiv \frac{b}{d} \pmod{\frac{m}{d}}.$$

Demostración. x_0 cumple que $ax_0 \equiv b \pmod{m}$ si y solo si $m|ax_0 - b$, es decir, si y solo si existe $q \in \mathbb{Z}$ tal que $ax_0 - b = mq$. Si dividimos por d la ecuación $ax_0 - b = mq$, obtenemos $\frac{a}{d}x_0 - \frac{b}{d} = \frac{m}{d}q$ y esto vale si y solo si $\frac{a}{d}x_0 \equiv \frac{b}{d} \pmod{\frac{m}{d}}$. □

Observación. Sean a, b números enteros y m un entero positivo, denotemos $d = \mathrm{mcd}(a, m)$ y supongamos que $d|b$. Entonces es fácil ver que

$$\mathrm{mcd}\left(\frac{a}{d}, \frac{m}{d}\right) = 1,$$

Lo cual implica, por la proposición anterior, que encontrar las soluciones de la ecuación lineal de congruencia se puede reducir al caso que $\mathrm{mcd}(a, b) = 1$.

Ejemplo. Hallemos las soluciones de la ecuación $21x \equiv 48 \pmod{114}$ con $0 \leqslant x \leqslant 1000$.

Demostración. Los divisores de 21 son 3 y 7, $3|114$ y $7 \not| 114$, por lo tanto $\mathrm{mcd}(21, 114) = 3$. Por la proposición 4.2.2, dividimos por 3 la ecuación y tenemos que el problema original es equivalente a encontrar todos los x tal que

$$7x \equiv 12 \pmod{38} \quad \text{tal que} \quad 0 \leqslant x \leqslant 1000.$$

Ahora bien, usando el algoritmo de Euclides, obtenemos el $\mathrm{mcd}(7, 38)$:

$$\begin{aligned} 38 &= 7 \cdot 5 + 3 \quad \Rightarrow \quad 3 = 38 - 7 \cdot 5 \\ 7 &= 3 \cdot 2 + 1 \quad \Rightarrow \quad 1 = 7 - 3 \cdot 2 \qquad (*) \\ 3 &= 1 \cdot 3 + 0. \end{aligned}$$

Luego $1 = \mathrm{mcd}(7, 38)$ y de (*) deducimos

$$\begin{aligned} 1 &= 7 - 3 \cdot 2 \\ &= 7 - (38 - 7 \cdot 5) \cdot 2 \\ &= 7 + (-2) \cdot 38 + 10 \cdot 7 \\ &= 11 \cdot 7 + (-2) \cdot 38. \end{aligned}$$

Por lo tanto

$$1 \equiv 11 \cdot 7 \pmod{38},$$

o equivalentemente,

$$7 \cdot 11 \equiv 1 \pmod{38}.$$

Si multiplicamos por 12 la ecuación obtenemos:

$$7 \cdot (11 \cdot 12) \equiv 12 \pmod{38}.$$

Por lo tanto, $x_0 = 11 \cdot 12 = 132$ es solución de la ecuación original.

Como $\mcd(7, 38) = 1$, por el teorema 4.2.1 deducimos que todas las soluciones de la ecuación $7x \equiv 12 \pmod{38}$ son de la forma $x = x_0 + 38k$ donde $k \in \mathbb{Z}$.

Debemos ver ahora cuales soluciones x cumplen $0 \leqslant x \leqslant 1000$. Lo haremos escribiendo las inecuaciones:

$$\begin{aligned} 0 &\leqslant 132 + 38k \leqslant 1000 \\ -132 &\leqslant 38k \leqslant 1000 - 132 = 868 \\ -\frac{132}{38} &\leqslant k \leqslant \frac{868}{38} \\ -3.4 &\leqslant k \leqslant 22.8 \end{aligned}$$

Luego $-4 \leqslant k \leqslant 22$, es decir k recorre todos los valores enteros desde -4 a 22.

Concluyendo: los x tales que $21x \equiv 48 \pmod{114}$ y $0 \leqslant x \leqslant 1000$ son los $x = 132 + 38k$ donde $k \in \mathbb{Z}$ y $-4 \leqslant k \leqslant 22$. □

§ *Ejercicios*

1) Resolver las siguientes ecuaciones lineales de congruencia

   a) $2x \equiv 1 \pmod{7}$.

   b) $3\,970\,x \equiv 560 \pmod{2755}$.

2) Determinar todas las posibles soluciones de las congruencias

   a) $5x \equiv 1 \pmod{11}$,

   b) $5x \equiv 7 \pmod{15}$.

3) Sea m un número entero $\geqslant 2$ y sea

$$\mathbb{Z}_m := \{0, 1, \ldots, m-1\}$$

el conjunto de restos de dividir por m. En $\mathbb{Z}_m$ definimos suma y producto de la siguiente manera: sean $a, b \in \mathbb{Z}_m$, entonces

$$\begin{array}{lll} a+b & = c & \text{si} \quad a+b \equiv c \pmod{m} \quad \wedge \quad 0 \leqslant c \leqslant m-1, \\ a \cdot b & = d & \text{si} \quad a \cdot b \equiv d \pmod{m} \quad \wedge \quad 0 \leqslant d \leqslant m-1. \end{array}$$

a) Probar que $\mathbb{Z}_m$ es un *anillo conmutativo con unidad*, es decir se cumplen los axiomas **I1**,...,**I6** de la sección 1.1 (cambiando $\mathbb{Z}$ por $\mathbb{Z}_m$).

b) Probar que si p es un número primo, entonces $\mathbb{Z}_p$ es *cuerpo*, es decir se cumplen los axiomas **I1**,...,**I6** de la sección 1.1 y además, para todo $a \in \mathbb{Z}_p$, $a \neq 0$, existe un único $b \in \mathbb{Z}_p$ tal que $ab = 1$ [Ayuda: usar la ecuación lineal de congruencia].

4) Sea p número entero positivo.

a) (Teorema de Wilson). Probar que si p es un número primo, entonces

$$(p-1)! \equiv -1 \pmod{p}.$$

[Ayuda: $1 \cdot (p-1) \equiv -1 \pmod{p}$ y debido al ejercicio *3 b)* los números $2, \ldots, p-2$ se pueden ordenar de a pares m_1, m_2 tal que $m_1 m_2 \equiv 1 \pmod{p}$].

b) Probar que si $(p-1)! \equiv -1 \pmod{p}$, entonces p es un número primo. [Ayuda: probar el contrarrecíproco y usar el hecho que si p es compuesto existe m con $1 < m < p$ y tal que $m|p$].

4.3 TEOREMA DE FERMAT

El siguiente lema nos sirve de preparación para la demostración del Teorema (o fórmula) de Fermat.

Lema 4.3.1. *Sea p un número primo, entonces*

a) $p \left| \binom{p}{r} \right.$, *con* $0 < r < p$,

b) $(a+b)^p \equiv a^p + b^p \pmod{p}$.

Demostración.

a) Escribamos el número binomial de otra forma:

$$\binom{p}{r} = \frac{p!}{r!(p-r)!} = p \cdot \frac{(p-1)!}{r!(p-r)!}$$

es un número entero, digamos k, luego

$$k \cdot r!(p-r)! = p \cdot (p-1)! \tag{4.3.1}$$

Como $p-1$, r y $p-r$ son menores que p, entonces $(p-1)!$, $r!$ y $(p-r)!$ son producto de números menores que p y por lo tanto son producto de primos menores que p. Por lo tanto, el primo p no aparece en la descomposición prima de $(p-1)!$, $r!$ y $(p-r)!$. Por la igualdad de la ecuación (4.3.1), p debe ser factor de $k = \binom{p}{r}$, luego $p|\binom{p}{r}$.

b) Por el teorema del binomio (teorema 2.5.1) sabemos que

$$(a+b)^p = \sum_{i=0}^{p} \binom{p}{i} a^i b^{p-i}.$$

Por *a)* es claro que $\binom{p}{i} a^i b^{p-i} \equiv 0 \pmod{p}$, si $0 < i < p$. Luego,

$$\begin{aligned}(a+b)^p &\equiv \tbinom{p}{0} b^p + \tbinom{p}{1} a^1 b^{p-1} + \cdots + \tbinom{p}{p-1} a^{p-1} b^1 + \tbinom{p}{p} a^p \pmod{p} \\ &\equiv b^p + 0 + \cdots + 0 + a^p \pmod{p} \\ &\equiv b^p + a^p \pmod{p}\end{aligned}$$

□

El siguiente es el llamado teorema de Fermat o teorema pequeño de Fermat.

Teorema 4.3.2. *Sea p un número primo y a número entero. Entonces*

$$a^p \equiv a \pmod{p}.$$

Demostración. Supongamos que $a \geqslant 0$, entonces hagamos inducción en a. Si $a = 0$, el resultado es trivial. Supongamos el resultado probado para k, es decir $k^p \equiv k \pmod{p}$. Entonces $(k+1)^p \equiv k^p + 1^p \equiv k+1 \pmod{p}$. La primera congruencia es debido al lema 4.3.1 *b)* y la segunda es válida por hipótesis inductiva. Luego $a^p \equiv a \pmod{p}$ cuando $a > 0$.

Si $a < 0$, entonces $-a > 0$ y ya vimos que $(-a)^p \equiv -a \pmod{p}$, es decir que $(-1)^p a^p \equiv (-1)a \pmod{p}$. Si $p \neq 2$, entonces $(-1)^p = -1$ y se deduce el resultado. Si $p = 2$, entonces $(-1)^p = 1$, pero como $1 \equiv -1 \pmod{2}$, obtenemos también $a^p \equiv a \pmod{p}$. □

Corolario 4.3.3. *Sea p primo y a entero tal $p \nmid a$, entonces $a^{(p-1)} \equiv 1 \pmod{p}$.*

Demostración. Por Fermat $a^p \equiv a \pmod{p}$, es decir

$$p|(a^p - a) = a(a^{(p-1)} - 1).$$

Como p no divide a a, tenemos que $p|(a^{(p-1)} - 1)$ (teorema 3.4.6), es decir $a^{(p-1)} \equiv 1 \pmod{p}$. □

Este último corolario es también conocido como teorema de Fermat.

La función de Euler $\phi(n)$, para $n \geqslant 1$, está definida como el cardinal del conjunto de los x entre 1 y n que son coprimos con n. El teorema de Fermat admite la siguiente generalización.

Teorema 4.3.4 (Teorema de Euler). *Sea n un entero positivo y a un número entero coprimo con n, entonces*

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

Demostración. Ver ejercicios 2 y 3, a continuación. □

§ *Ejercicios*

1) Usar el teorema de Fermat para calcular el resto de dividir 3^{47} por 23.

2) Sean $x_1, \ldots, x_k$ los números coprimos con n comprendidos entre 1 y n (es decir $k = \phi(n)$) y sea y coprimo con n. Entonces hay un reordenamiento de $yx_1, \ldots, yx_k$, es decir una permutación σ de $1, \ldots, k$, tal que $x_i \equiv yx_{\sigma_i} \pmod{n}$, para $1 \leqslant i \leqslant k$. [Ayuda: como y coprimo con n, existe v tal que $yv \equiv 1 \pmod{n}$].

3) Demostrar el teorema de Euler. [Ayuda: Sean $x_1, \ldots, x_k$ los números coprimos con n comprendidos entre 1 y n, por el ejercicio anterior $y^{\phi(n)}x_1 \ldots x_k = yx_1 \ldots yx_k \equiv x_1 \ldots x_k \pmod{n}$. Como $u = x_1 \ldots x_k$ coprimo con n, existe v tal que $uv \equiv 1 \pmod{n}$].

4.4 EL CRIPTOSISTEMA RSA

Una de las aplicaciones más elementales y difundidas de la aritmética es en el diseño de sistemas criptográficos. El RSA es el más conocido de ellos y será presentado en esta sección.

Por criptosistema nos referimos a sistemas de encriptamiento o codificación esencialmente pensados para proteger la confidencialidad de datos que se desean transmitir. Entre los criptosistemas encontramos los simétricos y los de clave pública o asimétricos.

Los sistemas criptográficos simétricos son aquellos en que tanto el emisor como el receptor conocen una función, digamos f y una palabra, digamos x (la clave), tanto la función como la clave deben ser confidenciales o más comúnmente solo la clave debe ser confidencial. Cuando el emisor desea enviar un mensaje M, entonces aplica la función a M y x, es decir

$M' = f(M, x)$, envía M' y el receptor aplica la función inversa y recupera M, es decir $M = f^{-1}(M', x)$. Es llamada *simétrica* porque tanto el emisor como el receptor manejan las mismas claves y el emisor puede pasar a receptor y viceversa usando la misma encriptación.

En los sistemas de clave pública el receptor conoce una clave privada y (no compartida por nadie) y publicita una clave pública x, de la misma manera que antes, si alguien desea enviar un mensaje M al receptor debe hacer $M' = f(M, x)$, pero el receptor para decodificar debe hacer $M = g(M', y)$, donde g es una función adecuada. Una ventaja evidente de los sistemas de clave pública es que no es necesario poner en conocimiento del emisor ninguna clave confidencial, más aún cualquier persona puede enviar en forma confidencial datos a otra persona que ha publicitado su clave.

Rivest, Shamir y Adleman descubrieron el primer criptosistema práctico de clave pública, que es llamado RSA. La seguridad del RSA se basa en la dificultad de factorizar números enteros grandes. Este sistema es el más comúnmente recomendado para uso en sistemas de clave pública. La mayor ventaja del RSA es que puede ser usado para proveer privacidad y autenticación (firma digital) en las comunicaciones. Su principal desventaja es que su implementación se basa en exponenciación de números enteros grandes, una operación que consume recursos de la computadora, aunque esto es cada vez menos significativo.

Antes de describir el RSA digamos que se basa fuertemente en el teorema de Fermat visto en la sección anterior.

En el sistema RSA deben realizarse algunos paso previos para fijar ciertos parámetros que luego nos permitirán encriptar y desencriptar los mensajes.

Idea del algoritmo

Supongamos que la persona B quiere enviar a la persona A un mensaje m pero encriptado de tal forma que sólo A pueda leer su contenido. Por su parte A hace públicos dos números e y n que son los que se utilizarán para encriptar los mensajes que le envíen.

Entonces a partir de m la persona B genera un mensaje cifrado c mediante la siguiente operación:

$$c \equiv m^e \pmod{n},$$

donde e y n es la clave pública de A.

Ahora A recupera le mensaje m a partir del mensaje en clave c mediante la operación inversa dada por

$$m \equiv c^d \pmod{n},$$

donde d es la clave privada que solo A conoce.

Elección de claves

Sean p y q primos distintos suficientemente .

- La *clave pública* es (n, e) con $n = pq$ y e tal que $1 < e < (p-1)(q-1)$ y $\text{mcd}(e, (p-1)(q-1)) = 1$.
- La *clave privada* es un entero d tal que $ed \equiv 1 \pmod{(p-1)(q-1)}$ y $0 \leqslant d < (p-1)(q-1)$.

Observación. Algunos comentarios sobre la elección de p, q, e, d.

- Los dos primos p y q deberían tener alrededor de 100 dígitos cada uno (longitud considerada segura en este momento).
- El número e puede elegirse pequeño y se selecciona haciendo prueba y error con el algoritmo de Euclides, es decir probando hasta encontrar un e tal que $\text{mcd}(e, (p-1)(q-1)) = 1$.
- La existencia de d está garantizada por el teorema 4.2.1 (ecuación lineal de congruencia), pues e y $(p-1)(q-1)$ son coprimos.

Encriptar y desencriptar mensajes

El receptor de mensajes publicita la clave pública (n, e). Obviamente no da a conocer ni p, ni q y mantiene segura la clave privada d. Como mencionamos anteriormente, el envío del mensaje y su decodificación requiere dos pasos

a) El emisor desea *encriptar* un número $m \in \{0, \ldots, n-1\}$ y para ello calcula $c \equiv m^e \pmod{n}$ y envía c al receptor.

b) El receptor desea *desencriptar* el mensaje, es decir usando la clave pública (n, e) y c, desea recuperar m: calcula $c^d \pmod{n}$ y veremos a continuación que este número es m.

Demostración del método

Debemos probar que el método anterior funciona y lo haremos en la siguiente proposición.

Proposición 4.4.1. *Sean*

- $n = pq$ *producto de dos números primos,*
- e *coprimo con* $(p-1)(q-1)$, *y*
- d *tal que* $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Entonces si $m \in \{0, \ldots, n-1\}$,

$$c \equiv m^e \pmod{n} \quad \Rightarrow \quad m \equiv c^d \pmod{n}.$$

Demostración. Como $ed \equiv 1 \pmod{(p-1)(q-1)}$, entonces existe k tal que

$$ed = 1 + k(p-1)(q-1). \tag{4.4.1}$$

Consideremos el mensaje m y si es o no coprimo con p.

Si $\mcd(m,p) = 1$, el Teorema de Fermat dice que $m^{p-1} \equiv 1 \pmod{p}$. Entonces $(m^{p-1})^x \equiv 1 \pmod{p}$ para cualquier x. En particular, para $x = k(q-1)$. Asi que tenemos:

$$m^{k(p-1)(q-1)} = (m^{p-1})^{k(q-1)} \equiv 1 \pmod{p}.$$

Multiplicando esta ecuación por m obtenemos

$$m^{1+k(p-1)(q-1)} \equiv m \pmod{p}. \tag{4.4.2}$$

Usando las ecuaciones (4.4.1) y (4.4.2) obtenemos:

$$m^{ed} \equiv m \pmod{p} \tag{4.4.3}$$

Esto, si $\mcd(m,p) = 1$. Pero si esto último no es cierto, entonces al ser p primo debemos tener $m \equiv 0 \pmod{p}$ y en ese caso la ecuación (4.4.3) es trivial (dice que $0 \equiv 0$). Concuyendo, la ecuación (4.4.3) se cumple para todo m.

Obviamente podemos reemplazar p por q en el razonamiento anterior, así que la ecuación (4.4.3) también es verdadera si reemplazamos p por q. De esa forma obtenemos:

$$p|(m^{ed} - m) \qquad \text{y} \qquad q|(m^{ed} - m).$$

Como p y q son primos distintos, entonces concluimos que $pq|(m^{ed} - m)$, es decir,

$$m^{ed} \equiv m \pmod{pq}.$$

□

Ejemplo. Probemos el sistema en forma práctica usando primos pequeños, por ejemplo $p = 31$, $q = 73$. En este caso $n = pq = 2\,263$.

Busquemos ahora un e: tenemos que $(p-1)(q-1) = 30 \cdot 72 = 2\,160$. Vemos que 2, 3, 5 dividen a $2\,160$, pero 7 es coprimo con $2\,160$. Tomemos entonces $e = 7$.

Usando el algoritmo de Euclides obtenemos $1 = 2 \cdot 2\,160 + (-617) \cdot 7$. Luego, -617 es solución de $7x \equiv 1 \pmod{2\,160}$, pero como queremos que la solución sea positiva menor que $2\,160$, podemos tomar $d = 2\,160 - 617 = 1\,543$.

Por lo tanto el receptor tiene clave pública $(2\,263, 7)$ y conserva en secreto su clave privada $1\,543$

- Supongamos que el emisor ha nacido en el año 1993 y quiere enviarle en secreto al receptor su año de nacimiento. Entonces encripta el año haciendo

$$1\,993^7 \equiv 10 \pmod{2\,263},$$

y envía, por una vía insegura, por ejemplo un email, el número 10 al receptor.

- El receptor calcula

$$10^{1\,543} \pmod{2\,263}$$

y obtiene nuevamente $1\,993$ (si quiere convencerse de esto ingrese `10^1543 mod 2263` en la ventana de búsqueda de Wolfram Alpha).

Pese a que el cálculo de desencriptado (en este caso $10^{1\,543} \pmod{2\,263}$) puede parecer costoso computacionalmente, hay métodos eficientes para hacerlo como se puede ver, por ejemplo, en la sección 4.5.

Firma digital

Una propiedad importante del RSA es que puede ser usado para firma digital o autenticación. En las hipótesis de la proposición 4.4.1, es claro que lo que probamos es que

$$(m^e)^d \equiv m \pmod{n},$$

para $m \in \{0, \dots, n-1\}$. Ahora bien

$$(m^e)^d = m^{ed} = (m^d)^e,$$

es decir

$$(m^d)^e \equiv m \pmod{n},$$

para $m \in \{0, \dots, n-1\}$. Por lo tanto, el receptor puede codificar un número o mensaje m calculando $b \equiv m^d \pmod{n}$ y cualquiera que conozca la clave pública puede obtener el original calculando $b^e \pmod{n}$.

Lo interesante de esto es que si el receptor envía m (el mensaje) y b (la codificación de m), cualquiera puede comprobar que el mensaje ha sido codificado por el receptor (y no por otra persona) verificando que $m \equiv b^e \pmod{n}$.

Ejemplo. Como ya hemos mencionado, podemos ver que el RSA también puede ser usado para un sistema de *autenticación*, es decir es posible comprobar quien es la persona que envía el mensaje. Veamos una forma de hacerlo: la persona A tiene clave pública (e, n) y clave privada d y la persona B tiene clave pública (e', n') y clave privada d'.

a) La persona B desea enviar un mensaje m (en forma segura) a la persona A y quiere certificar que el mensaje fue enviado por él.

b) B calcula $x \equiv m^{d'} \pmod{n'}$. Es decir encripta su mensaje usando su clave privada.

c) Ahora B codifica m y x con la clave pública de A, es decir calcula $c \equiv m^e \pmod{n}$ e $y \equiv x^e \pmod{n}$.

d) B envía c e y al receptor A.

e) La persona A recupera m y x calculando $m \equiv c^d \pmod{n}$ y $x \equiv y^d \pmod{n}$.

f) A comprueba que el mensaje proviene de B o, mejor dicho, proviene de la persona con clave pública (e', n'), verificando que $m \equiv x^{e'} \pmod{n'}$.

4.5 MÉTODO BINARIO PARA EXPONENCIACION MODULAR (*)

Vimos en la sección anterior que para implementar el criptosistema RSA es esencial tener una forma eficiente de calcular exponenciación modular, es decir tener la capacidad de calcular el resto de dividir por cierto número la potencia muy grande de otro número.

Más explícitamente, sean a, d, n enteros positivos se desea calcular r tal que $a^d \equiv r \pmod{n}$ con $0 \leqslant r < n$. En otras palabras, se desea calcular el resto de de dividir a^d por n. Estaríamos tentados de calcular a^d y luego hacer la congruencia módulo n, pero cuando d es grande, por ejemplo $d > 10^{20}$, este cálculo es imposible para cualquier computador.

Para realizar el cálculo de $a^d \equiv r \pmod{n}$ en forma eficiente podemos utilizar el *método binario de exponenciación modular* que explicaremos a continuación.

Primero, recordemos la definición recursiva de la potencia de un número: sea a número (entero, racional, real, etc.), entonces

$$a^d = \begin{cases} 1 & , \quad \text{si } d = 0, \\ a \cdot a^{d-1} & , \quad \text{si } d > 0. \end{cases} \tag{4.5.1}$$

Aplicar esta definición tiene el inconveniente de que si d es un número grande, la cantidad de pasos que se deben realizar para calcular a^d es directamente proporcional (y mayor) a d, número que puede ser inmanejable para una computadora.

Una definición alternativa de las potencias de un número nos permite bajar el número de operaciones en forma significativa. Sea a número (entero, racional, real, etc.), entonces

$$a^d = \begin{cases} 1 & , \quad \text{si } d = 0, \\ a \cdot a^{d-1} & , \quad \text{si } d > 0 \text{ y } d \text{ impar;} \\ (a^2)^{\frac{d}{2}} & , \quad \text{si } d > 0 \text{ y } d \text{ par.} \end{cases} \tag{4.5.2}$$

Con esta definición la cantidad de operaciones necesarias para calcular la potencia de un número decrece enormemente. Observemos que mientras que con la definición (4.5.1) la cantidad de operaciones es del orden de d, con la definición (4.5.2) la cantidad de operaciones es del orden de $\log_2(d)$.

Ejemplo 4.5.1. Supongamos que queremos calcular 9^{17}. En este caso, podríamos hacer este cálculo directamente, multiplicando 17 veces 9 y así obtenemos 16677181699666569 con 17 multiplicaciones. Este sería el método que se deduce de la definición recursiva (4.5.4).

Probemos con la segunda definición recursiva:

$$\begin{aligned} 9^{17} &= 9 \cdot 9^{16} \\ &= 9 \cdot (9^2)^8 && = 3 \cdot 81^8 \\ &= 9 \cdot (81^2)^4 && = 3 \cdot 6561^4 \\ &= 9 \cdot (6561^2)^2 && = 3 \cdot 43046721^2 \\ &= 9 \cdot 1853020188851841 && = 16677181699666569. \end{aligned} \tag{4.5.3}$$

Es decir, en bastante menos pasos hemos podido calcular 9^{17}.

Pasemos ahora al problema de calcular exponenciación modular.

Supongamos ahora que queremos calcular r tal que, por ejemplo,

$$5^{1125899986842625} \equiv r \pmod{100000037},$$

y $0 \leqslant r < 100000037$. Hacer este cálculo directamente no nos da un resultado satisfactorio, ni siquiera con un programa de computadora. Pueden hacer

el intento con un lenguaje de programación, Python por ejemplo, y verán que el programa no termina. Esto se debe a que $5^{1125899986842625}$ es un número inmenso cuya representación no cabría en la memoria de ninguna computadora ni actual ni futura. La clave para poder calcular r es usar un análogo a la definición (4.5.2), pero con congruencias. Sea a número entero y sean $d \geqslant 0$ y $n \geqslant 1$. Definimos $\%$ el *operador módulo* de tal forma que $a \% n$ devuelve el resto de dividir a por n. Hay muchas forma de calcular $a^d \% n$, y nosotros elegimos la siguiente forma recursiva:

$$a^d \% n = \begin{cases} 1 & , \quad \text{si } d = 0, \\ a \cdot (a^{n-1} \% n) \% n & , \quad \text{si } d > 0 \text{ y } d \text{ impar}; \\ (a^2 \% n)^{\frac{n}{2}} \% n & , \quad \text{si } d > 0 \text{ y } d \text{ par}. \end{cases} \tag{4.5.4}$$

Esta definición no solo reduce la cantidad de pasos para calcular r, si no que, veremos un poco más adelante, también mantiene la cantidad de dígitos de los cálculos intermedios acotada.

Ejemplo 4.5.2. Sea $n = 23$, calculemos usando la definición 4.5.4, r tal que $9^{17} \equiv r \pmod{23}$ con $0 \leqslant r < 23$, es decir $9^{17} \% 23$. Apliquemos la definición (4.5.4) y aprovechemos los cálculos hechos en el ejemplo 4.5.1:

$$\begin{aligned} 9^{17} &\equiv 9 \cdot 9^{16} \\ &\equiv 9 \cdot (9^2)^8 \quad \equiv \quad 9 \cdot 12^8 && (\text{pues } 9^2 \equiv 12 \pmod{23}) \\ &\equiv 9 \cdot (12^2)^4 \quad \equiv \quad 9 \cdot 6^4 && (\text{pues } 12^2 \equiv 6 \pmod{23}) \\ &\equiv 9 \cdot (6^2)^2 \quad \equiv \quad 9 \cdot 13^2 && (\text{pues } 6^2 \equiv 13 \pmod{23}) \\ &\equiv 9 \cdot 8 && (\text{pues } 13^2 \equiv 8 \pmod{23}) \\ &\equiv 3 && (\text{pues } 9 \cdot 8 \equiv 3 \pmod{23}). \end{aligned}$$

Es decir, $9^{17} \equiv 3 \pmod{23}$. Observar que pese a que 9^{17} es un número de 17 dígitos, los cálculos que hacemos (módulo 23) involucran pocos dígitos.

Como observamos en el ejemplo, en cada paso donde d es par tenemos que calcular el cuadrado de un número y elevarlo a una potencia que es la mitad que la anterior. Luego, si el exponente es $d < 2^k$ en alrededor de k pasos (unos pocos más en realidad) obtendremos el resultado deseado.

En el caso que habíamos planteado: calcular $5^{1125899986842625} \equiv r \pmod{100000037}$, como $1125899986842625 < 2^{51}$, si lo resolvemos usando el método de la fórmula (4.5.4) necesitaremos poco más de 50 pasos,y en cada paso la longitud de los dígitos involucrados es totalmente manejable.

La definición (4.5.4) nos muestra como casi inmediatamente podemos obtener una fórmula recursiva para la exponenciación modular: sea $n > 0$, definimos $//$ el *operador cociente* de tal forma que $a // n$ es el cociente entero de dividir a por n. Es decir

$$a = n * (a // n) + (a \% n).$$

Entonces, nuestro interés es calcular $f(a,d) = a^d \,\%\, n$ en forma eficiente y la fórmula (4.5.4) es equivalente a

$$f(a,d) = \begin{cases} 1 & , \quad \text{si } d = 0, \\ (a \cdot f(a, d-1)) \,\%\, n & , \quad \text{si } d > 0 \text{ y } d \text{ impar;} \\ f(a^2 \,\%\, n, d \,//\, 2) & , \quad \text{si } d > 0 \text{ y } d \text{ par.} \end{cases} \tag{4.5.5}$$

Observar que en la definición hay dos $\%\, n$ que podrían haber sido reemplazados por una instancia de la función f pues $x \,\%\, n = f(x,1)$. Pero, en este caso, se pueden calcular directamente por el algorimo de división, pues están aplicados en números considerados "no grandes".

Esta función recursiva es fácilmente trasladable a pseudocódigo.

EXPONENCIACIÓN MODULAR RECURSIVA

```
def f(a, d):
    if d == 0:
        res = 1
    elif d % 2 == 1:
        res =  (a * f(a, d - 1)) % n
    else:
        res =  f(a**2 % n, d // 2)
    return res
```

En pseudocódigo una versión iterativa de la exponenciacion modular se puede describir de la siguiente forma.

EXPONENCIACIÓN MODULAR

```
def exp_modular(a, d, n):
    res = 1
    base, exponente = a, d
    while exponente > 0:
        # invariante: (a**d) % n = (res * base**exponente) % n
        if exponente % 2 == 1:
            res = (res * base) % n
            exponente = exponente // 2
            base = base**2 % n
        elif exponente % 2 == 0:
            exponente = exponente // 2
            base = base**2 % n
    return res
```

4.6 PRUEBAS DE PRIMALIDAD (*)

En la implementación del criptosistema RSA es fundamental el uso de primos de grandes dimensiones, por ejemplo de más de 100 dígitos. Ahora bien, tratar de demostrar que un número (grande) es primo buscando sus divisores es imposible con una arquitectura de computadoras determinística, como la es la de las computadoras actuales. Justamente en la dificultad de la descomposición prima se basa la fortaleza del criptosistema RSA.

Sin embargo, veremos en esta parte del apunte que hay algoritmos que permiten determinar en forma probabilística si un número es primo o no. También mencionaremos, al final de esta sección, algoritmos que determinan eficientemente y en forma determinística si un número es primo a o no.

El algoritmo que veremos principalmente es el test de primalidad Miller-Rabin probabilístico. Como su nombre lo indica es una prueba probabilística de primalidad: un algoritmo que determina si un número dado es probable que sea primo.

Este test es ampliamente utilizado en la práctica (en RSA, por ejemplo) y es una de las pruebas más simples y rápidas conocidas.

Describamos en forma amplia el método del test: dado m un entero positivo.

- Si le hacemos el test a m y no supera la prueba, entonces el número no es primo.
- Si hacemos k veces el test y m supera las k pruebas, entonces m tiene la probabilidad $1-(1/4^k)$ de ser primo.

El test de Miller-Rabín se basa en comprobar k veces (para un k dado) si el número es fuertemente probable primo respecto a una base. Veamos a continuación las definiciones necesarias para comprender este concepto.

Definición 4.6.1. Sea $n > 2$ un entero impar, entonces $n = 2^s \cdot d + 1$ con d impar. Sea a entero tal que $0 < a < n$. Entonces diremos que n es *fuertemente probable primo (FPP) respecto a la base* a si se cumple

- $a^d \equiv 1 \pmod{n}$, o
- $a^{2^r \cdot d} \equiv -1 \pmod{n}$ para algún r tal que $0 \leqslant r < s$.

Con la aplicación del teorema de Fermat y la ecuación lineal de congruencia probaremos que todo número primo es FPP respecto a cualquier base. El contrarrecíproco de esta afirmación nos dice que un número que no es FPP respecto a alguna base es compuesto.

Lema 4.6.2. *Sea n un primo impar, entonces las únicas raíces cuadradas de 1 modulo n son 1 y -1. Es decir,*

$$x^2 \equiv 1 \pmod{n} \quad \Rightarrow \quad x \equiv \pm 1 \pmod{n}.$$

Demostración. Sea x tal que $x^2 \equiv 1 \pmod{n}$, luego $x^2 - 1 \equiv 0 \pmod{n}$, como $x^2 - 1 = (x-1)(x+1)$, obtenemos $(x-1)(x+1) \equiv 0 \pmod{n}$. Esto quiere decir que $n|(x-1)(x+1)$. Como n es primo, $n|x-1$ o $n|x+1$, es decir $x \equiv 1 \pmod{n}$ o $x \equiv -1 \pmod{n}$. □

Teorema 4.6.3. *Si n es un primo impar, entonces n es FPP para cualquier base a con $0 < a < n$.*

Demostración. Consideremos la sucesión $a^{2^s \cdot d}, a^{2^{s-1} \cdot d}, \ldots, a^{2d}, a^d$ y observemos que cada término de la sucesión es el cuadrado del siguiente.

Por el teorema de Fermat, $a^{2^s \cdot d} = a^{n-1} \equiv 1 \pmod{n}$. Luego $(a^{2^{s-1} \cdot d})^2 \equiv 1 \pmod{n}$ y por lo tanto $a^{2^{s-1} \cdot d}$ es una raíz cuadrada de 1 módulo n. Por el lema anterior obtenemos que $a^{2^{s-1} \cdot d} \equiv \pm 1 \pmod{n}$.

Si $a^{2^{s-1} \cdot d} \equiv -1 \pmod{n}$, obtenemos el resultado. En caso contrario $a^{2^{s-1} \cdot d} \equiv 1 \pmod{n}$, luego $(a^{2^{s-2} \cdot d})^2 \equiv 1 \pmod{n}$ y por lo tanto $a^{2^{s-2} \cdot d}$ es una raíz cuadrada de 1 módulo n y en consecuencia $a^{2^{s-2} \cdot d} \equiv \pm 1 \pmod{n}$.

Iterando el razonamiento anterior concluimos que alguno de los términos de la sucesión $a^{2^r \cdot d}$ es congruente a -1 módulo n o bien todos los términos son congruentes a 1, en particular $a^d \equiv 1 \pmod{n}$, con lo cual n resulta ser probable primo fuerte. □

Ahora bien, también es cierto el recíproco (que no demostraremos): un número n que es FPP respecto a todas las bases $0 < a < n$ es primo. Podríamos intentar ver que un número es primo probando que es FPP para cualquier base, pero este cálculo es computacionalmente imposible para primos grandes.

Por otro lado, un número n que es FPP respecto alguna base $0 < a < n$ podría ser compuesto, pero hay una probabilidad mayor que 0.75 de que sea primo. La verificación con diferentes bases de que un número es FPP acerca a 1 la probabilidad de que el número sea primo.

El test probabilístico de primalidad de Miller-Rabin se basa en las observaciones realizadas más arriba: sea n entero positivo impar y sea k entero positivo.

(1) Elegir al azar a entero tal que $0 < a < n$.

(2) Verificar que n es FPP respecto a la base a.

(3) Repetir (1) y (2) k veces.

Si n es FPP las k veces, entonces decimos que n supera el test probabilístico de primalidad de Miller-Rabin y tiene probabilidad $1 - (1/4^k)$ de ser primo (y lo consideramos primo).

Mostramos a continuación una implementación en pseudocódigo.

TEST DE PRIMALIDAD DE MILLER-RABIN

```
def test_Miller_Rabin(n: int, k: int) -> bool:
    s, d = satisfacen n = 2**s * d + 1, s impar
    repetir k veces:
        a = entero al azar entre 2 y n-1
        fpp = False # suponemos n no es fuertemente primo en base a
        if 1 == a**d % n:
            fpp = True
        else:
            r = 0
            while r <= s and fpp == False:
                if n - 1 == a**(2**r * d) % n:
                    fpp = True
                r = r + 1
            if fpp == False: # si n no pasa la prueba
                return False
    return True # n pasó las k pruebas
```

Observemos que el test de primalidad de Miller-Rabin hace uso de la exponenciación modular y por lo tanto debe ser implementado usando el método binario de exponenciación modular.

El test de Miller-Rabin probabilístico se deriva del test de Miller: asumiendo que es válida la Hipótesis de Riemman, sea n entero positivo, si comprobamos que n es FPP en base a con $2 < a < 2\ln(n)^2$, entonces n es primo.

El test de Miller no se utiliza en la práctica. Para la mayoría de los propósitos, el uso adecuado de la prueba probabilística de Miller-Rabin es mucho más rápida y brinda suficente seguridad del resultado. Para fines teóricos que requieren un algoritmo de tiempo polinomial determinista, el test de Miller fue reemplazado por la prueba de primalidad de Agrawal–Kayal–Saxena (AKS), que no se basa en suposiciones no probadas.