Consecuencias del algoritmo del Euclides y Factorización única

Pablo L. De Nápoli

Departamento de Matemática
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Álgebra I - Primer cuatrimestre de 2020

El MCD como combinación lineal

Una consecuencia muy importante del algoritmo de Euclides es el siguiente teorema:

Teorema (4.4.5 en el apunte de la profesora Kirck)

El máximo común divisor entre a y b se puede escribir como una combinación lineal de ellos: es decir, existen enteros $s = s(a, b)$ y $t = t(a, b)$ tales que

$$s \cdot a + t \cdot b = (a : b)$$

El algoritmo de Euclides nos permite dar una prueba constructiva de este teorema, construyendo las funciones $s(a, b)$ y $t(b, a)$ de manera recursiva. De nuevo, podemos suponer $a \geq b$.

Definiendo $s(a, b)$ y $t(a, b)$ recursivamente

Para encontrar la recurrencia, efectuamos la división entera de a por b, obteniendo un cociente $k = k(a, b)$ y un resto $r = r(a, b)$, como antes. Entonces por el invariante del algoritmo de Euclides sabemos que

$$(a : b) = (b : r)$$

Ahora supongamos que ya sabemos encontrar $s(b, r)$ y $t(b, r)$ tales que

$$s(b, r) \cdot b + t(b, r) \cdot r = \text{mcd}(b, r)$$

Sustituyendo $r = a - k \cdot b$, encontramos que

$$s(b, r) \cdot b + t(b, r) \cdot [a - k \cdot b] = (b : r)$$

y efectuando la distributiva:

$$t(b, r) \cdot a + (s(b, r) - k \cdot t(b, r)) \cdot b = (b : r)$$

Esto sugiere definir las funciones s y t recursivamente por:

$$s(a, b) = t(b, r), \quad t(a, b) = s(b, r) - k(a, b) \cdot t(b, r)$$

Definiendo $s(a,b)$ y $t(a,b)$(continuación)

Para que esta recurrencia funcione, debemos definir s y t cuando el algoritmo de Euclides termina, es decir $s(a,0)$ y $t(a,0)$. (¡El caso base!)

Queremos que

$$s(a,0)\cdot a + t(a,0)\cdot 0 = (a:0) = a$$

Una manera de lograrlo es definiendo

$$s(a,0) = 1, \quad t(a,0) = 0$$

(Esta elección es arbitraria, pero necesaria si quereemos que t sea una función bien definida)

Por inducción global en $a \in \mathbb{N}_0$, se prueba facilmente que $s, t : D \to \mathbb{Z}$ quedan bien definidas en el dominio

$$D = \{(a,b) \in \mathbb{N}_0 \times \mathbb{N}_0 : a \geq b\}$$

y que cumplen que:

$$s(a,b)\ a + t(a,b)\ b = mcd(a,b)$$

Bonus Track: ¡lo programamos!

El algoritmo de Euclides con combinación lineal

```
def Euclides_extendido(a,b):
  if b>a:
   return Euclides_extendido(b,a)
  if b==0:
      s_a_b=1
      t_a_b=0
      mcd_a_b = a
  else:
     k,r=divmod(a,b)
     s_b_r, t_b_r, mcd_b_r = Euclides_extendido(b,r)
     s_a_b =  t_b_r
     t_a_b = s_b_r - k*t_b_r
     mcd_a_b = mcd_b_r
  chequea_invariante (a,b,s_a_b,t_a_b,mcd_a_b)
  return (s_a_b,t_a_b,mcd_a_b)
```

Función que chequea el invariante del algoritmo

La condición

$$s(a,b)\ a + t(a,b)\ b = mcd(a,b)$$

es ahora el invariante del algoritmo. La siguiente función (en el sentido informático del término), permite chequearla en cada paso. Es decir nos ayuda a comprobar la correctitud de nuestro algoritmo:

Chequeamos el invariante

```
def chequea_invariante(a,b,s_a_b,t_a_b,mcd_a_b):
  print("s(",a,",",b,")=",s_a_b,end=', ')
  print("t(",a,",",b,")=",t_a_b,end=', ')
  print("mcd(",a,",",b,")=",mcd_a_b)
  print(mcd_a_b,"=",s_a_b,"*",a,"+",t_a_b,"*",b)
  if not(s_a_b *a + t_a_b * b == mcd_a_b):
    sys.exit("¡No se cumple!")
```

Un Ejemplo

Escribamos al $mcd(32, 17)$ como combinación lineal entre ellos:

```
Calculamos el máximo común divisor entre  32  y  17
s( 1 , 0 )= 1, t( 1 , 0 )= 0, mcd( 1 , 0 )= 1
1 = 1 * 1 + 0 * 0
s( 2 , 1 )= 0, t( 2 , 1 )= 1, mcd( 2 , 1 )= 1
1 = 0 * 2 + 1 * 1
s( 15 , 2 )= 1, t( 15 , 2 )= -7, mcd( 15 , 2 )= 1
1 = 1 * 15 + -7 * 2
s( 17 , 15 )= -7, t( 17 , 15 )= 8, mcd( 17 , 15 )= 1
1 = -7 * 17 + 8 * 15
s( 32 , 17 )= 8, t( 32 , 17 )= -15, mcd( 32 , 17 )= 1
1 = 8 * 32 + -15 * 17
```

Otro Ejemplo

Escribamos al $mcd(360, 28) = 4$ como combinación lineal entre ellos:

```
Calculamos el máximo común divisor entre  360  y  28
s( 4 , 0 )= 1, t( 4 , 0 )= 0, mcd( 4 , 0 )= 4
4 = 1 * 4 + 0 * 0
s( 24 , 4 )= 0, t( 24 , 4 )= 1, mcd( 24 , 4 )= 4
4 = 0 * 24 + 1 * 4
s( 28 , 24 )= 1, t( 28 , 24 )= -1, mcd( 28 , 24 )= 4
4 = 1 * 28 + -1 * 24
s( 360 , 28 )= -1, t( 360 , 28 )= 13, mcd( 360 , 28 )= 4
4 = -1 * 360 + 13 * 28
```

Divisibilidad y coprimalidad

Definición (Proposición 4.5.10 del apunte de la profesora Krick)

Se dice que $a, b \in \mathbb{Z}$ no ambos nulos son coprimos si y solo si $(a : b) = 1$, es decir si y solo si los únicos divisores comunes de a y b son ± 1 .
Notación: $a \perp b$.

Como consecuencia del algoritmo de Euclides a y b son coprimos si y sólo si existen $s, t \in \mathbb{Z}$ tales que $s \cdot a + t \cdot b = 1$.

Propisición (Proposición 4.5.12 en dicho apunte)

Sean $a, b, c, d \in \mathbb{Z}$ con $c \neq 0$ y $d \neq 0$. Entonces

1. $c|a \wedge d|a \wedge c \perp d \Rightarrow cd|a$.
2. $d|ab \wedge d \perp a \Rightarrow d|b$.

OJO: estas afirmaciones no son ciertas si no se piden las pro- piedades de coprimalidad.

La propiedad Fundamental de los números primos

Definición

Un número $p \in \mathbb{Z}$ se dice primo si es distinto de $0, \pm 1$ y sus únicos divisores son ± 1 y $\pm p$.

Observación

Si p es primo y p no divide a $a \in \mathbb{Z}$, entonces a y p son coprimos.

Teorema (de Euclides, 4.6.3 en el apunte)

Sea p un primo y sean $a, b \in \mathbb{Z}$. Entonces

$$p|a \cdot b \Rightarrow p|a \vee p|b.$$

El Teorema fundamental de la aritmética

Teorema (4.6.5 en el apunte)

Sea $a \in \mathbb{Z}$, $a \neq 0, \pm 1$. Entonces a se escribe en forma única como producto de primos (positivos), (o se factoriza en forma única como producto de primos (positivos),) es decir:

1. $\forall \in \mathbb{Z}, a \neq 0, \pm 1$, *existe* $r \in \mathbb{N}$ *y existen primos positivos* $p_1, \ldots p_r$ *distintos y* $m_1, \ldots m_r \in N$ *tales que*

$$a = \pm p_1^{m_1} \cdot p_2^{m_2} \ldots p_r^{m_r}$$

2. *Esta escritura es única salvo permutación de los primos.*

La factorización en forma estándar

Definición

Si p es un primo, definimos la ***valuación p-ádica*** $v_p : \mathbb{N} \to \mathbb{N}_0$ *del siguiente modo: $v_p(n)$ como el exponente del primo p en la factorización de n si p dvide a n, y 0 sino. De esta forma la factorización de n puede escribirse*

$$n = \prod_{p|n} p^{v_p(n)}$$

Teorema

Sean $a, b \in \mathbb{N}$, entonces

i) *Para todo primo p,*

$$v_p(a \cdot b) = v_p(a) + v_p(b)$$

ii) *$a|b$ si y sólo si $v_p(a) \leq v_p(b)$ para todo primo p.*

Máximo común divisor y mínimo común múltiplo en términos de primos

Teorema

Sean $a, b \in \mathbb{N}$, *entonces*

i) *El máximo comun divisor* $(a : b) = mcd(a, b)$ *entre ellos se puede escribir*

$$(a : b) = mcd(a, b) = \prod_{p|a \wedge p|b} p^{\text{mín}(v_p(a), v_p(b))}$$

ii) *Similarmente, el mínimo comun múltiplo* $[a : b] mcm(a, b)$ *entre ellos se puede escribir*

$$[a : b] = mcm(a, b) = \prod_{p|a \vee p|b} p^{\text{máx}(v_p(a), v_p(b))}$$

ii)

$$[a, b] \cdot (a : b) = a \cdot b$$

Ejemplo

360 =2^3 * 3^2 * 5^1

$$v_p(360) = \begin{cases} 3 & \text{si} & p = 2 \\ 2 & \text{si} & p = 3 \\ 1 & \text{si} & p = 5 \\ 0 & \text{para los otros} & p \end{cases}$$

490 =2^1 * 5^1 * 7^2

mcd(360,490) = 10 = 2^1 * 5^1

mcm(360,490) = 17640 = 2^3 * 3^2 * 5^1 * 7^2