DEFINICIONES RECURSIVAS

- Una función $f : \mathbb{Z}_m \longrightarrow A$ se define recursivamente como sigue:

  Caso Base: Sea $l \in \mathbb{Z}_m$ y sean $a_m, \ldots a_l \in A$. Definimos

  $f(m) = a_m$

  $\vdots$

  $f(l) = a_l$

  Caso Recursivo:
  Sea $n > l$

  $f(n) = e(..n..f(i)..) \quad m \leq i < n$

  donde $e(..n..f(i)..)$ es una expresión que depende de n y de $f(m), \ldots, f(n-1)$

  Distintos valores de a_i y e dan lugar a distintas funciones.

- Ej.1
  Función de Fibonacci $\quad f : \mathbb{Z}_1 \longrightarrow \mathbb{Z}$

  (B) $f(1) = 1 \quad f(2) = 1$

  (R) $\forall n > 2 \;\; f(n) = f(n-1) + f(n-2)$

- Ej.2
  $\sum_{i=1}^{n} a_i$ donde a_i es una expresión que depende de i (ej. $a_i = (2i-1)$) es una función recursiva.

  (B) $f(1) = a_1$

  (R)$f(n) = \sum_{i=1}^{n} a_i = a_1 + a_2 + \ldots + a_{n-1} + a_n = f(n-1) + a_n \quad \forall n \geq 2$

- Ej.3
  $\prod_{i=1}^{n} a_i$ donde a_i es una expresión que depende de i (ej. $a_i = 2^i$) es una función recursiva.

  (B) $f(1) = a_1$

  (R)$f(n) = \prod_{i=1}^{n} a_i = a_1 \cdot a_2 \cdot \ldots \cdot a_{n-1} \cdot a_n = f(n-1) \cdot a_n \quad \forall n \geq 2$

  $\sum_{i=m}^{n} a_i = 0 \quad m > n \qquad \prod_{i=m}^{n} a_i = 1 \quad m > n$

- Ej.4

  (B) $f(0) = 3$

  (R) $f(n) = 2f(n-1) + 3 \quad \forall n > 0$

  Hallar $f(5)$

  - Por reescritura
    $f(5) = 2f(4)+3 = 2(2f(3)+3)+3 = 4f(3)+9 = 4(2f(2)+3)+9 = 8f(2)+21 = 8(2f(1)+3)+21 = 16f(1)+45 = 16(2f(0)+3)+45 = 32f(0)+93 = 32\cdot 3+93 = 96 + 93 = 189$

- Por iteración

  Primero calculamos $f(0) \ldots f(4)$ y sustituimos y operamos en la expresión e de la definición.

  $f(0) = 3$

  $f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$

  $f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$

  $f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45$

  $f(4) = 2f(3) + 3 = 2 \cdot 45 + 3 = 93$

  $f(5) = 2f(4) + 3 = 2 \cdot 93 + 3 = 189$

Los conceptos de recursión e inducción están muy relacionados. Muchas veces demostramos propiedades de funciones recursivas por inducción.

Por ejemplo, sea $f : \mathbb{Z}_1 \longrightarrow \mathbb{Z}$ la función de Fibonacci.

(B) $f(1) = 1 \quad f(2) = 1$

(R) $\forall n > 2 \;\; f(n) = f(n-1) + f(n-2)$

Demostrar:

$\forall n \in \mathbb{Z}_3 \;\; f(1)^2 + \ldots + f(n)^2 = f(n)f(n+1)$

Base: $S(3) \equiv f(1)^2 + f(2)^2 + f(3)^2 = 1 + 1 + 2^2 = 6 = 2 \cdot 3 = f(3)f(4)$
$(f(4) = f(3) + f(2) = 2 + 1 = 3)$

(HI) $f(1)^2 + \ldots + f(k)^2 = f(k)f(k+1)$

Paso Inductivo $\forall k \in \mathbb{Z}_3 \quad S(k) \Longrightarrow S(k+1)$

Tenemos que demostrar

$S(k+1) \equiv (f(1)^2 + \ldots + f(k)^2) + f(k+1)^2 = f(k+1)f(k+2)$

$S(k+1) \equiv (f(1)^2 + \ldots + f(k)^2) + f(k+1)^2 = f(k)f(k+1) + f(k+1)^2 = f(k+1)(f(k+1) + f(k)) = f(k+1)f(k+2)$

Esta demostración es correcta y no es necesario utilizar inducción completa porque basta con aplicar la hipótesis de inducción $S(k) \equiv (f(1)^2 + \ldots + f(k)^2) = f(k)f(k+1)$. No es necesario aplicar $S(k-1)$ u otros casos anteriores.

Además, puesto que $n = 4$ es el primer caso no incluido en la base hay que comprobar si la demostración de este caso, que se realiza ya aplicando el paso inductivo, es correcta.

Podemos afirmar que lo es pues sólo utilizamos $S(3)$ esn la demostración (demostrado en la base) y la definición de f.

Extendemos la definición a **funciones con varios argumentos:**

- $f : B \times \mathbb{Z}_m \longrightarrow A$ se define recursivamente como sigue:

  Base: Sea $l \geq m, \quad l \in \mathbb{Z}_m$ Para $m \leq n \leq l$ definimos

  $$f(b,n) = e_n(..b..) \begin{cases} f(b,m) = e_m(..b..) \\ \quad \vdots \\ f(b,l) = e_l(..b..) \end{cases}$$

  donde $e_n(..b..)$ $n = m \ldots l$ son expresiones que dependen de b.

  Caso Recursivo:

  Sea $n > l$

  $$f(b,n) = e(..b..n..f(c,i)..) \quad m \leq i < n \quad (c \text{ puede ser } b.)$$

  Ej.1

  $+ : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$

  (B) $+(m,0) = m \quad e_0(..b..) = b$

  (R) $+(m,s(n)) = s(+(m,n)) \quad e(..b..n..f(c,i)..) = s(f(b,n))$

  Ej.2

  $* : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$

  (B) $*(m,0) = 0 \quad e_0(..b..) = 0$

  (R) $*(m,s(n)) = m + *(m,n) \quad e(..b..n..f(c,i)..) = b + f(b,n)$

- $f : \mathbb{Z}_m \times C \longrightarrow A$ se define recursivamente como sigue:

  Base: Sea $l \geq m, \quad l \in \mathbb{Z}_m$ Para $m \leq n \leq l$ definimos

  $$f(n,c) = e_n(..c..) \begin{cases} f(m,c) = e_m(..c..) \\ \quad \vdots \\ f(lcl) = e_l(..c..) \end{cases}$$

  donde $e_n(..c..)$ $m \leq n \leq l$ son expresiones que dependen de c.

  Paso Recursivo:

  Sea $n > l$

  $$f(n,c) = e(..c..n..f(i,d)..) \quad m \leq i < n \quad (d \text{ puede ser } c.)$$

- $f : B \times \mathbb{Z}_m \times C \longrightarrow A$ se define recursivamente como sigue:

  Base: Sea $l \geq m, \quad l \in \mathbb{Z}_m$ Definimos

  $$f(b,m,c) = e_m(..b..c..)$$
  $$\vdots$$
  $$f(b,l,c) = e_l(..b..c..)$$

  donde $e_j(..b..c..)$ $j = m \ldots l$ son expresiones que dependen de b y c.

  Paso Recursivo:

  Sea $n > l$

  $$f(b,n,c) = e(..b..n..c..f(d,i,g)..) \quad m \leq i < n \quad (d, g \text{ pueden ser } b, c.)$$