

Guia Practica 1

LyC 2024 2C

Pablo Mercado

Contents

Chapter 1

	Page
Ejercicio 1	2
Ejercicio 2	2
Ejercicio 3	4
Ejercicio 4	5
Ejercicio 5	5
Ejercicio 6	7
1.1 Random Examples	12
1.2 Random	14
1.3 Algorithms	15

Chapter 1

Question 1: pregunta1

Ejercicio 1. Mostrar que, dado un k fijo, la función constante $f(x) = k$ puede definirse usando las funciones iniciales y composición (*sin* usar recursión primitiva).

Solution: Cualquier $k \in \mathbb{N}$ puede ser construido aplicando sucesivamente la función $s(x)$ a la función $n(x)$.

$$f(x) = s_k(x) = \underbrace{s(s(\dots s(n(x)) \dots))}_{k \text{ veces}} \quad (1.1)$$

Ya que $s(n(x))$ es composición de funciones primitivas, es primitiva recursiva. Razonando de forma inductiva, cada vez que aplicamos $s(x)$ a un cierto $s_{k-1}(x)$, obtenemos un $s_k(x)$ que de nuevo, por composición, es primitiva recursiva.

Question 2

Ejercicio 2. Probar que las siguientes funciones son primitivas recursivas, mostrando que pueden obtenerse a partir de las funciones iniciales usando composición y/o recursión primitiva:

$$f_1(x, y) = x + y \quad f_2(x, y) = x \cdot y \quad f_3(x, y) = x^y \quad f_4(x, y) = \underbrace{x^{x^{\dots x}}}_{y \text{ veces}}$$

$$g_1(x) = x \div 1 \quad g_2(x, y) = x \div y \quad g_3(x, y) = \max\{x, y\} \quad g_4(x, y) = \min\{x, y\}$$

Observaciones : Se asume que $f_4(x, 0) = 1$.
$$x \div y = \begin{cases} x - y & \text{si } y \leq x \\ 0 & \text{si } y > x \end{cases}$$

Solution: $f_1(x, y) = x + y$

$$\begin{aligned} f_1(x, 0) &= 0 = n(x) \\ f_1(x, y) &= \underbrace{((\dots ((0 + 1) + 1) \dots + 1) + 1) + 1}_{\substack{\text{X} + \dots \quad y \text{ veces}}} \\ f_1(x, y) &= f_1(x, y - 1) + 1 \\ &= s(f_1(x, y - 1)) \end{aligned}$$

Pero para que cierre la aridad con el esquema de recursión primitiva, debemos encontrar una función g tal que: $f_1(x, y) = g(f(x, y - 1), x, y - 1)$, esto se arregla tomando $g(x, y, z) = s(u_1^3(x, y, z))$ Entonces nos queda:

$$f_1(x, y) = g(f(x, y - 1), x, y - 1) = s(u_1^3(f(x, y - 1), x, y - 1)) \quad (1.2)$$

Solution: $f_2(x, y) = x \cdot y$

$$\begin{aligned} f_2(x, 0) &= n(x) \\ f_2(x, y) &= f_2(x, y-1) + x = f_1(f_2(x, y-1), x) \\ f_2(x, y) &= f_1(u_1^3(f_2(x, y-1), x, y-1), u_2^3(f_2(x, y-1), x, y-1)) \\ f_2(x, y) &= g(f_2(x, y-1), x, y-1) \end{aligned}$$

Con $g(x, y, z) = f_1(u_1^3(x, y, z), u_2^3(x, y, z))$, ya que g se obtiene por composicion de funciones PR, entonces tambien es PR.

Solution: $f_3(x, y) = x^y$

$$\begin{aligned} f_3(x, 0) &= 1 \\ f_3(x, y) &= f_3(x, y-1) \cdot x = f_2(f_3(x, y-1), x) \\ f_3(x, y) &= f_2(u_1^3(f_3(x, y-1), x, y-1), u_2^3(f_3(x, y-1), x, y-1)) \\ f_3(x, y) &= g(f_3(x, y-1), x, y-1) \end{aligned}$$

Con $g(x, y, z) = f_2(u_1^3(x, y, z), u_2^3(x, y, z))$, ya que g se obtiene por composicion de funciones PR, entonces tambien es PR.

Solution: $f_4(x, y) = \underbrace{x^{x^{x^{\dots x}}}}_{y \text{ veces}}$

$$\begin{aligned} f_4(x, 0) &= 1 \\ f_4(x, y) &= f_4(x, y-1)^x = f_3(f_4(x, y-1), x) \\ f_4(x, y) &= f_3(u_1^3(f_4(x, y-1), x, y-1), u_2^3(f_4(x, y-1), x, y-1)) \end{aligned}$$

Con $g(x, y, z) = f_3(u_1^3(x, y, z), u_2^3(x, y, z))$, ya que g se obtiene por composicion de funciones PR, entonces tambien es PR.

Solution: $g_1(x) = x \div 1$

$$\begin{aligned} g_1(0) &= n(x) \\ g_1(x) &= u_2^2(g_1(x), x-1) \end{aligned}$$

Solution: $g_2(x, y) = x \div y$

$$\begin{aligned} g_2(x, 0) &= u_1^1(x) \\ g_2(x, y) &= g_2(x, y-1) - 1 = g_1(g_2(x, y-1)) \\ g_2(x, y) &= g_1(u_1^3(g_2(x, y-1), x, y-1)) \\ g_2(x, y) &= g(g_2(x, y-1), x, y-1) \end{aligned} \quad \mathfrak{g}(x, y, z) = \mathfrak{g}_1(u_1^3(x, y, z))$$

Con $g(x, y, z) = g_2(u_1^3(x, y, z))$, ya que g se obtiene por composicion de funciones PR, entonces tambien es PR.

Solution: $g_3(x, y) = \max\{x, y\}$

$$\begin{aligned} \max\{x, y\} &= (x \leq y) \cdot y + \alpha(x \leq y) \cdot x \\ \max\{x, y\} &= f_2((x \leq y), y) + f_2(\alpha(x \leq y), y) \\ \max\{x, y\} &= f_1(\underbrace{f_2((x \leq y), y)}_{h_1}, \underbrace{f_2(\alpha(x \leq y), y)}_{h_2}) \end{aligned}$$

Las funciones α (negacion) y \leq son las definidas en la clase teorica numero 2. Se puede probar por composicion que h_1 y h_2 son PR. Por lo tanto queda demostrado por composicion que g_3 tambien es PR.

Solution: $g_4(x, y) = \min\{x, y\}$

$$\begin{aligned}\min\{x, y\} &= (x \leq y).x + \alpha(x \leq y).y \\ \min\{x, y\} &= f_2((x \leq y).x) + f_2(\alpha(x \leq y), y) \\ \min\{x, y\} &= \underbrace{f_1(f_2((x \leq y), y))}_{h_1}, \underbrace{f_2(\alpha(x \leq y), y))}_{h_2}\end{aligned}$$

Las funciones α (negacion) y \leq son las definidas en la clase teorica numero 2. Se puede probar por composicion que h_1 y h_2 son PR. Por lo tanto queda demostrado por composicion que g_3 tambien es PR.

Question 3

Ejercicio 3. Sea C_i la clase de funciones iniciales, es decir, aquella que contiene a:

$$n(x) = 0 \quad s(x) = x + 1 \quad u_i^n(x_1, \dots, x_n) = x_i \text{ para cada } n \in \mathbb{N} \text{ e } i \in \{1, \dots, n\}$$

y sea C_c la (mínima) clase que extiende a C_i y se encuentra cerrada por composición, i.e., si f, g_1, \dots, g_m están en C_c , entonces $h(x_1, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n))$ también lo está.

a. Demostrar que para toda $f : \mathbb{N}^n \rightarrow \mathbb{N}$, f está en C_c sii existe $k \geq 0$ tal que, o bien sucede $f(x_1, \dots, x_n) = k$, o bien para algún i fijo, se tiene $f(x_1, \dots, x_n) = x_i + k$.

b. Mostrar que existe una función primitiva recursiva que no está en C_c .

Solution: a)

(\rightarrow)

Vamos a usar induccion estructural

Casos Base: Veamos que se cumple para las funciones iniciales

$$\begin{aligned}n(x) &= 0, \quad k = 0 \\ s(x) &= x + 1, \quad k = 1 \\ u_i^n(x_1, \dots, x_n) &= x_i + 0, \quad k = 0\end{aligned}$$

Paso inductivo:

Sea $f \in C$, $f(x_1, \dots, x_n) = h(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n))$. Veamos que o bien $f(x_1, \dots, x_n) = k$, o bien $f(x_1, \dots, x_n) = x_i + k$

Las funciones que componen a h , cumplen con la HI. Por lo tanto, separemos en casos:

Caso $h(x_1, \dots, x_n) = k$: Entonces tenemos que $f(x_1, \dots, x_n) = k = k'$, ya esta!

Caso $h(x_1, \dots, x_n) = x_j + k'$: Entonces tenemos que $f(x_1, \dots, x_n) = g_i(x_1, \dots, x_n)$.

Ahora tenemos 2 sub-casos mas:

Caso $g_i(x_1, \dots, x_n) = k''$: $f(x_1, \dots, x_n) = k' + k'' = k$

Caso $g_i(x_1, \dots, x_n) = x_j + k''$: $f(x_1, \dots, x_n) = x_j + k'' + k' = x_j + k$

En ambos casos, vemos que se cumple lo que queriamos.

(\leftarrow)

Ya vimos que la funcion $h_k(x) = k$, puede ser definida por composicion y recursion a partir de las funciones iniciales, entonces $h_k \in C$. Entonces $f(x_1, \dots, x_n) = h_k(u_1^n(x_1, \dots, x_n)) = k$, xlt $f \in C$

Se puede definir la función $s_k(x) = x + k$ usando sucesivamente la función s , por composición $s_k \in C$. Xlt $f(x_1, \dots, x_n) = x_i + k = s_k(u_i^n(x_1, \dots, x_n))$. Y esta última al ser composición de funciones de C , nos asegura que f también está en C .

Question 4

Ejercicio 4. Llamamos *predicado* a cualquier función $p : \mathbb{N}^n \rightarrow \{0, 1\}$, escribimos $p(a_1, \dots, a_n)$ en lugar de $p(a_1, \dots, a_n) = 1$ y decimos, informalmente, en ese caso, que “ $p(a_1, \dots, a_n)$ es verdadero”. Mostrar que los predicados $\leq, \geq, =, \neq, <, >$: $\mathbb{N}^2 \rightarrow \{0, 1\}$ están en cualquier clase *PRC*.

Solution:

- $\leq (x, y) = \alpha(x - y)$
- $\geq (x, y) = \alpha(y - x)$
- $= (x, y) = (x \leq y) \cdot (y \leq x)$
- $\neq (x, y) = \alpha(x = y)$
- $< (x, y) = \alpha(x \geq y)$
- $> (x, y) = \alpha(x \leq y)$

Si una función f está en cualquier clase *PRC*, entonces está en la intersección de todas las clases *PRC*, por el teorema visto en clases, podemos decir que ser función primitiva recursiva es condición necesaria y suficiente para esto. En el ejercicio 2 vimos que la función de multiplicación, resta de naturales y la suma son primitivas recursivas. Probemos que $\alpha(x)$ es primitiva recursiva.

$$\alpha : \mathbb{N} \rightarrow \{0, 1\}$$

$$\alpha(x) = \begin{cases} 1 & \text{si } x = 0 \\ 0 & \text{si } x \neq 0 \end{cases}$$

Ahora vamos a definir $\alpha(x)$ por recursión primitiva.

$$\begin{aligned} \alpha(0) &= s(n(x)) = 1 \\ \alpha(x+1) &= g(\alpha(x-1), x) = 0 \quad \text{con } g(x, y) = n(u_2^2(x, y)) \\ \alpha(x+1) &= n(u_2^2(\alpha(x), x)) = 0 \end{aligned}$$

Entonces $\alpha(x)$ es primitiva recursiva. Las funciones que describimos arriba están formadas por composición entre funciones primitivas recursivas, por lo tanto son primitivas recursivas.

Question 5

Ejercicio 5. Sea C una clase *PRC*, sean $f_1, \dots, f_k, g : \mathbb{N}^n \rightarrow \mathbb{N}$ funciones en C y sean también $p_1, \dots, p_k : \mathbb{N}^n \rightarrow \{0, 1\}$ predicados disjuntos en C (i.e., no sucede $p_i(a_1, \dots, a_n) = p_j(a_1, \dots, a_n) = 1$ con $i \neq j$ para ningún $(a_1, \dots, a_n) \in \mathbb{N}^n$). Mostrar que también está en C cualquier función h que cumpla:

$$h(x_1, \dots, x_n) = \begin{cases} f_1(x_1, \dots, x_n) & \text{si } p_1(x_1, \dots, x_n) \\ \vdots & \\ f_k(x_1, \dots, x_n) & \text{si } p_k(x_1, \dots, x_n) \\ g(x_1, \dots, x_n) & \text{si no} \end{cases} \quad (1.3)$$

Observar que h queda completamente determinada por el esquema.

Solution:

$$\begin{aligned}
h(x_1, \dots, x_n) &= f_1(x_1, \dots, x_n).p_1(x_1, \dots, x_n) + \\
&\quad f_2(x_1, \dots, x_n).p_2(x_1, \dots, x_n) + \\
&\quad f_k(x_1, \dots, x_n).p_k(x_1, \dots, x_n) + \\
&\quad \vdots \\
&\quad g(x_1, \dots, x_n).\alpha(p_1(x_1, \dots, x_n) + \dots + p_k(x_1, \dots, x_n))
\end{aligned}$$

Notemos que $g(x_1, \dots, x_n).\alpha(p_1(x_1, \dots, x_n) + \dots + p_k(x_1, \dots, x_n))$ toma el valor 1, sii todos los predicados valen 0 a la vez.

Sabemos que toda clase PRC contiene a las funciones inicales. En el ejercicio 2, vimos que las funciones $f_1(x, y) = x + y$ y $f_2(x, y) = x.y$ son primitivas recursivas, por teorema, sabemos que esta en cualquier clase PRC. Para que h este en C tiene que poder obtenerse a partir de otras en C mediante recursion primitiva o composicion.

Claim 1.0.1

$$f_i(x_1, \dots, x_n).p_i(x_1, \dots, x_n) \in C$$

Proof: Trivial, ya que $f_2(x, y) = x.y \in C$ por ser pr. ⊗

Claim 1.0.2

$a_n(x_1, \dots, x_n) = x_1 + \dots + x_n$ es primitiva recursiva.

Proof: Notemos que la funcion a_n puede ser obtenida aplicando la funcion suma (f_1) sucesivamente (composicion) de esta manera: $a_n(x_1, \dots, x_n) = f_1(\dots(f_1(f_1(x_1, x_2), x_3))\dots, x_n)$. Entoces $a_n(x_1, \dots, x_n) \in C$. Mas generalmente, culaquier funcion que sume todos sus parametros es primitiva recursiva. ⊗

Claim 1.0.3

$$g(x_1, \dots, x_n) = \alpha(p_1(x_1, \dots, x_n) + \dots + p_k(x_1, \dots, x_n)) \in C$$

Proof: $g(x_1, \dots, x_n) = \alpha(a_n(p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n))) = g'(p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n))$
Con $h'(x_1, \dots, x_n) = \alpha \circ a_n(x_1, \dots, x_n)$ ⊗

Claim 1.0.4

$$\begin{aligned}
h(x_1, \dots, x_n) &= f_1(x_1, \dots, x_n).p_1(x_1, \dots, x_n) + \\
&\quad f_2(x_1, \dots, x_n).p_2(x_1, \dots, x_n) + \\
&\quad f_k(x_1, \dots, x_n).p_k(x_1, \dots, x_n) + \\
&\quad \vdots \\
&\quad g(x_1, \dots, x_n).\alpha(p_1(x_1, \dots, x_n) + \dots + p_k(x_1, \dots, x_n)) \in C
\end{aligned}$$

Proof: Sean:

$$\begin{aligned}
f'_i(x_1, \dots, x_n) &= f_i(x_1, \dots, x_n).p_i(x_1, \dots, x_n) \in C, \forall i : 1 \dots k \\
g'(x_1, \dots, x_n) &= g(x_1, \dots, x_n).\alpha(p_1(x_1, \dots, x_n) + \dots + p_k(x_1, \dots, x_n)) \in C
\end{aligned}$$

Escribamos a h como composicion de funciones en C

$$\begin{aligned}
h(x_1, \dots, x_n) &= a_{k+1}(f'_1(x_1, \dots, x_n), \dots, f'_k(x_1, \dots, x_n), g'(x_1, \dots, x_n)). \\
\therefore h &\in C.
\end{aligned}$$

⊗

Question 6

Ejercicio 6. a) Demostrar que el predicado $\text{par}(x) = \begin{cases} 1 & \text{si } x \text{ es par} \\ 0 & \text{si no} \end{cases}$ esta en toda clase PRC.

b. Demostrar que la función $f(x) = \lfloor x/2 \rfloor$ está en toda clase PRC.

c. Sea C una clase PRC, y sean $f : \mathbb{N}^n \rightarrow \mathbb{N}$ y $g_1, g_2 : \mathbb{N}^{n+2} \rightarrow \mathbb{N}$ funciones en C . Mostrar que también está en C cualquier h que cumpla:

$$h(x_1, \dots, x_n, t) = \begin{cases} f(x_1, \dots, x_n) & \text{si } t = 0 \\ g_1(x_1, \dots, x_n, k, h(x_1, \dots, x_n, t-1)) & \text{si } t = 2 \cdot k + 1 \\ g_2(x_1, \dots, x_n, k, h(x_1, \dots, x_n, t-1)) & \text{si } t = 2 \cdot k + 2 \end{cases}$$

Observar que h queda completamente determinada por este esquema.

Solution: a)

Proposition 1.0.1

$\text{par}(x) = \begin{cases} 1 & \text{si } x \text{ es par} \\ 0 & \text{si no} \end{cases}$ esta en toda clase PRC.

Proof: Sabemos que estar en toda clase PRC es equivalente a ser función primitiva recursiva, veamos que $\text{par}(x)$ cumple esto.

$$\begin{aligned} \text{par}(0) &= 1 = s(n(x)) \\ \text{par}(x+1) &= \alpha(\text{par}(x)) = \alpha(u_1^2(\text{Par}(x), x)) \\ \text{par}(x+1) &= g(\text{Par}(x), x) \quad \text{con } g = \alpha \circ u^2 \end{aligned}$$

Ya que pudimos definir a la función par por recursión primitiva, entonces par es primitiva recursiva. \therefore par está en toda clase PRC. \odot

Solution: b)

Proposition 1.0.2

$f(x) = \lfloor x/2 \rfloor$ está en toda clase PRC.

Proof:

Note:-

x	0	1	2	3	4	5	6	7	8	9	10
$\lfloor \frac{x}{2} \rfloor$	0	0	1	1	2	2	3	3	4	4	5

$$f(x) = \lfloor \frac{x}{2} \rfloor = \begin{cases} 0 & \text{si } x = 0 \\ f(x-1) + \underbrace{\alpha(\text{par}(x))}_{\text{impar}(x)} & \text{c.c} \end{cases}$$

$$\begin{aligned} f(0) &= 0 = n(x) \\ f(x+1) &= f(x) + \text{impar}(x) \\ f(x+1) &= f_1(f(x), \text{impar}(x)) \\ f(x+1) &= g(f(x), x) \quad \text{con } g(x) = f_1(\underbrace{u_1^1(x)}_{u_1^1(x, y)}, \text{impar}(x)) \end{aligned}$$

Entonces f es primitiva recursiva y por lo tanto está en toda clase PRC. \odot

Solution: c)

Proposition 1.0.3

h puede ser re-escrito como:

$$h(x_1, \dots, x_n, t) = \begin{cases} f(x_1, \dots, x_n) & \text{si } t = 0 \\ g_1(x_1, \dots, x_n, \lfloor \frac{t-1}{2} \rfloor, h(x_1, \dots, x_n, t-1)) & \text{si } t \text{ es impar} \\ g_2(x_1, \dots, x_n, \lfloor \frac{t-1}{2} \rfloor, h(x_1, \dots, x_n, t-1)) & \text{si } t \text{ es par} \end{cases}$$

Proof:

- si $t = 2k + 1$ es impar, entonces $\left\lfloor \frac{(2k+1)-1}{2} \right\rfloor = k$
- si $t = 2k + 2$ es par, entonces $\left\lfloor \frac{(2k+2)-1}{2} \right\rfloor = k$

De esto podemos deducir que $\lfloor \frac{t-1}{2} \rfloor = k$

☺

Proposition 1.0.4

h puede ser obtenido por recursion primitiva y por lo tanto esta en \mathcal{C} :

$$h(x_1, \dots, x_n, t) = \begin{cases} f(x_1, \dots, x_n) & \text{si } t = 0 \\ g(h(x_1, \dots, x_n, t-1), x_1, \dots, x_n, t-1) & \text{c.c} \end{cases}$$

Proof: Nos gustaria encontrar una funcion $g \in \mathcal{C}$, tal que:

$$g_i(x_1, \dots, x_n, \lfloor \frac{t-1}{2} \rfloor, h(x_1, \dots, x_n, t-1)) = g(h(x_1, \dots, x_n, t-1), x_1, \dots, x_n, t-1)$$

Esto lo podemos hacer cambiando de lugar los parametros y "desenvolviendo" el parametro $t-1$, para que quede solo. Definamos la funcion g .

$$g(u_{n+2}^{n+2}(x_1, \dots, x_{n+2}), u_1^{n+2}(x_1, \dots, x_{n+2}), \dots, u_n^{n+2}(x_1, \dots, x_{n+2}), e(x_1, \dots, x_{n+2}))$$

Donde:

$$\begin{aligned} d(x) &= 2 \cdot x + \text{impar}(x) \\ e(x_1, \dots, x_{n+2}) &= d(u_{n+1}^{n+2}(x_1, \dots, x_{n+2})) \end{aligned}$$

Se puede probar que d y e son funciones primitivas recursivas, por lo tanto, estan en \mathcal{C} . La funcion d es la que "desenvuelve" el parametro $t-1$ de la funcion $\lfloor \frac{t}{2} \rfloor$. Entonces g esta correctamente definida por composicion de funciones en \mathcal{C} . $\therefore h \in \mathcal{C}$. ☺

Question 7

Ejercicio 7. Sea \mathcal{C} una clase PRC y sea $p : \mathbb{N}^{n+1} \rightarrow \{0, 1\}$ un predicado en \mathcal{C} . Mostrar que también están en \mathcal{C} las siguientes funciones:

$$\begin{aligned}
\text{cantidad}_p(x_1, \dots, x_n, y, z) &= |\{t \mid y \leq t \leq z \wedge p(x_1, \dots, x_n, t)\}| \\
\text{todos}_p(x_1, \dots, x_n, y, z) &= \begin{cases} 1 & \text{si } (\forall t : y \leq t \leq z) p(x_1, \dots, x_n, t) \\ 0 & \text{si no} \end{cases} \\
\text{alguno}_p(x_1, \dots, x_n, y, z) &= \begin{cases} 1 & \text{si } (\exists t : y \leq t \leq z) p(x_1, \dots, x_n, t) \\ 0 & \text{si no} \end{cases} \\
\text{minimo}_p(x_1, \dots, x_n, y, z) &= \begin{cases} \min\{t \mid y \leq t \leq z \wedge p(x_1, \dots, x_n, t)\} & \text{si existe tal } t \\ 0 & \text{si no} \end{cases} \\
\text{maximo}_p(x_1, \dots, x_n, y, z) &= \begin{cases} \max\{t \mid y \leq t \leq z \wedge p(x_1, \dots, x_n, t)\} & \text{si existe tal } t \\ 0 & \text{si no} \end{cases} \\
\text{unico}_p(x_1, \dots, x_n, y, z) &= \begin{cases} u & \text{si } \{u\} = \{t \mid y \leq t \leq z \wedge p(x_1, \dots, x_n, t)\} \\ z + 1 & \text{si no} \end{cases}
\end{aligned}$$

Observacion: pueden usarse los operadores acotados (\min , \sum , \forall , \exists) vistos en la teorica.

Solution: $\text{cantidad}_p(x_1, \dots, x_n, y, z) = |\{t \mid y \leq t \leq z \wedge p(x_1, \dots, x_n, t)\}|$

$$\text{cantidad}_p(x_1, \dots, x_n, y, z) = \sum_{t=y}^z p(x_1, \dots, x_n, t)$$

Solution: $\text{todos}_p(x_1, \dots, x_n, y, z) = \begin{cases} 1 & \text{si } (\forall t : y \leq t \leq z) p(x_1, \dots, x_n, t) \\ 0 & \text{si no} \end{cases}$

$$\text{todos}_p(x_1, \dots, x_n, y, z) = (\forall t)_{y \leq t \leq z} (t \rightarrow P(x_1, \dots, x_n, t))$$

Solution: $\text{alguno}_p(x_1, \dots, x_n, y, z) = \begin{cases} 1 & \text{si } (\exists t : y \leq t \leq z) p(x_1, \dots, x_n, t) \\ 0 & \text{si no} \end{cases}$

$$\text{alguno}_p(x_1, \dots, x_n, y, z) = 1 \leq \text{cantidad}_p(x_1, \dots, x_n, y, z)$$

Solution: $\text{minimo}_p(x_1, \dots, x_n, y, z) = \begin{cases} \min\{t \mid y \leq t \leq z \wedge p(x_1, \dots, x_n, t)\} & \text{si existe tal } t \\ 0 & \text{si no} \end{cases}$

$$\text{minimo}_p(x_1, \dots, x_n, y, z) = \min_{t \leq z} (t \rightarrow y \wedge p(x_1, \dots, x_n, t))$$

Solution: $\text{maximo}_p(x_1, \dots, x_n, y, z) = \begin{cases} \max\{t \mid y \leq t \leq z \wedge p(x_1, \dots, x_n, t)\} & \text{si existe tal } t \\ 0 & \text{si no} \end{cases}$

$$\max_{t \leq z} (y \leq t \wedge p(x_1, \dots, x_n, t))$$

Solution: $\text{unico}_p(x_1, \dots, x_n, y, z) = \begin{cases} u & \text{si } \{u\} = \{t \mid y \leq t \leq z \wedge p(x_1, \dots, x_n, t)\} \\ z + 1 & \text{si no} \end{cases}$

Devolvemos u, si u es el unico numero en rango que cumple.

$$f(x_1, \dots, x_n, y, z) = \begin{cases} \sum_{t=y}^z p(x_1, \dots, x_n, t) \cdot t & \text{si } \text{cantidad}_p(x_1, \dots, x_n, y, z) = 1 \\ z + 1 & \text{sino} \end{cases}$$

Question 8

Ejercicio 8 Mostrar que las siguientes funciones están en toda clase PRC:

$$\text{cociente}(x, y) = \lfloor x/y \rfloor$$

$$\text{resto}(x, y) = x \bmod y$$

$$\text{divide}(x, y) = \begin{cases} 1 & \text{si } x \text{ divide a } y \\ 0 & \text{si no} \end{cases}$$

$$\text{primo}(x) = \begin{cases} 1 & \text{si } x \text{ es un número primo} \\ 0 & \text{si no} \end{cases}$$

$$\text{raiz}(x, y) = \begin{cases} \lfloor \sqrt[y]{x} \rfloor & \text{si } x \neq 0 \\ 0 & \text{si } x = 0 \end{cases}$$

$$\text{nprimo}(n) = k \text{ si } k \text{ es primo y hay sólo } n - 1 \text{ primos positivos menores que } k$$

Solution: $\text{cociente}(x, y) = \lfloor x/y \rfloor$

Como lo vimos en la practica, lo podemos pensar como el minimo numero t , tal que $t + 1$ se pasa de x al multiplicarlo por y .

$$t = \frac{x}{y} \Rightarrow t \cdot y = x$$

$$\text{cociente}(x, y) = \min_{t \leq x} ((t + 1) \cdot y > x)$$

Solution: $\text{resto}(x, y) = x \bmod y$

$$\text{resto}(x, y) = x - \text{cociente}(x, y) \cdot y$$

Solution: $\text{divide}(x, y) = \begin{cases} 1 & \text{si } x \text{ divide a } y \\ 0 & \text{si no} \end{cases}$

$$\text{divide}(x, y) = (\text{resto}(x, y) = 0)$$

Solution: $\text{raiz}(x, y) = \begin{cases} \lfloor \sqrt[y]{x} \rfloor & \text{si } x \neq 0 \\ 0 & \text{si } x = 0 \end{cases}$

$$\text{raiz}(x, y) = \min_{t \leq y} ((t + 1)^x > x)$$

Solution: $\text{primo}(x) = \begin{cases} 1 & \text{si } x \text{ es un número primo} \\ 0 & \text{si no} \end{cases}$

$$\text{primo}(x) = (\forall d)_{<x} (d > 1 \vee \alpha(\text{divide}(d, x)))$$

Solution: $\text{nprimo}(x) = k$ sii k es primo y solo hay $n - 1$ primos menores que k
Esto fue resuelto en las diapos de la teorica.

Question 9

Ejercicio 9. Considerar la codificación de pares de naturales dada por $\langle x, y \rangle = 2^x(2y + 1) \div 1$. Mostrar que las funciones observadoras $l, r : \mathbb{N} \rightarrow \mathbb{N}$ tales que $l(\langle x, y \rangle) = x$ y $r(\langle x, y \rangle) = y$ están en toda clase PRC.

Solution: Esto fue resuelto en las diapos de la teorica.

Question 10

Ejercicio 10. Mostrar que fib, la función de Fibonacci, está en toda clase PRC, donde:

$$\begin{aligned}
fib(0) &= 0 \\
fib(1) &= 1 \\
fib(n+2) &= fib(n+1) + fib(n)
\end{aligned}$$

Proof: Para resolver este problema, vamos a construir una funcion auxiliar h que guarde tres valores de f a la vez y que nos permita recuperar la funcion f componiendola con algo p.r. Luego vamos a probar que h es pr.

$$h(n) = [fib(n), fib(n+1)]$$

Ahora vemos que $h(n)$ es pr.

$$h(0) = [fib(0), fib(1)] = [0, 1]$$

$[0,1]$ se codifica con un numero natural, entonces es pr.
Por definicion de h , sabemos que:

$$\begin{aligned}
h(n+1) &= [fib(n+1), fib(n+2)] \\
h(n+1) &= [fib(n+1), fib(n+1) + fib(n)]
\end{aligned}$$

De nuevo usando la definicion de h .

$$h(n+1) = [h(n)[2], h(n)[2] + h(n)[1]]$$

Dado que $h(0)$ resulto una funcion p.r. (es constante) y que el valor $h(n+1)$ se puede expresar mediante una funcion p.r. que solo depende de n y del valor anterior $h(n)$, la funcion h se ajusta al esquema de recursion primitiva visto en clase y, por lo tanto, es p.r.

Ya que la lista solo tiene 2 valores en cada caso, tambien se podia resolver usando pares (tuplas).

$$\therefore fib(n) = h(n)[1]$$

☺

Question 11

Ejercicio 11 Demostrar que toda clase PRC se encuentra cerrada por recursion mutua. Es decir, dada C , una clase PRC y dadas f_1, f_2, g_1 y g_2 funciones en C , mostrar que también están en C las funciones h_1 y h_2 que cumplen:

$$\begin{aligned}
h_1(x_1, \dots, x_n, t) &= \begin{cases} f_1(x_1, \dots, x_n) & \text{si } t = 0 \\ g_1(h_1(x_1, \dots, x_n, t-1), h_2(x_1, \dots, x_n, t-1), x_1, \dots, x_n, t) & \text{si no} \end{cases} \\
h_2(x_1, \dots, x_n, t) &= \begin{cases} f_2(x_1, \dots, x_n) & \text{si } t = 0 \\ g_2(h_2(x_1, \dots, x_n, t-1), h_1(x_1, \dots, x_n, t-1), x_1, \dots, x_n, t) & \text{si no} \end{cases}
\end{aligned}$$

Observar que h_1 y h_2 quedan completamente determinadas por el esquema de recursion mutua.

Proof: Sea:

$$H(x_1, \dots, x_n, t) = \langle h_1(x_1, \dots, x_n, t), h_2(x_1, \dots, x_n, t) \rangle$$

Si probamos que H es PR, entonces queda probado que h_1 y h_2 son PR, ya que se obtienen por composicion de funciones PR (codificacion de tupla y obseadores $l(x)$ y $r(x)$)

Caso Base:

$$H(x_1, \dots, x_n, 0) = \langle f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n) \rangle = f(x_1, \dots, x_n)$$

Es trivial ver que $f(x_1, \dots, x_n)$ es PR.

Caso recursivo:

$$\begin{aligned}
 H(x_1, \dots, x_n, t+1) &= \langle h_1(x_1, \dots, x_n, t), h_2(x_1, \dots, x_n, t) \rangle \\
 H(x_1, \dots, x_n, t+1) &= \langle g_1(h_1(x_1, \dots, x_n, t), h_2(x_1, \dots, x_n, t), x_1, \dots, x_n, t), \\
 &\quad g_2(h_1(x_1, \dots, x_n, t), h_2(x_1, \dots, x_n, t), x_1, \dots, x_n, t)) \rangle \\
 H(x_1, \dots, x_n, t+1) &= \langle g_1(l(H(x_1, \dots, x_n, t)), r(H(x_1, \dots, x_n, t)), x_1, \dots, x_n, t), \\
 &\quad g_2(r(H(x_1, \dots, x_n, t)), l(H(x_1, \dots, x_n, t)), x_1, \dots, x_n, t)) \rangle
 \end{aligned}$$

Para H sea PR, debe poder escribirse de la forma:

$$H(x_1, \dots, x_n, t+1) = F(H(x_1, \dots, x_n, t), x_1, \dots, x_n, t)$$

Claim 1.0.5

Existe una funcion F tal que:

$$F(H(x_1, \dots, x_n, t), x_1, \dots, x_n, t) = \langle g_1(h_1(x_1, \dots, x_n, t), h_2(x_1, \dots, x_n, t), x_1, \dots, x_n, t), \\ g_2(h_1(x_1, \dots, x_n, t), h_2(x_1, \dots, x_n, t), x_1, \dots, x_n, t)) \rangle$$

Y ademas F es PR.

Proof:

$$F(x_1, \dots, x_{n+2}) = \langle F_r(x_1, \dots, x_{n+2}), F_l(x_1, \dots, x_{n+2}) \rangle$$

Con F_r y F_l funciones PR que cumplen lo siguiente:

$$\begin{aligned}
 F_l(x_1, \dots, x_{n+2}) &= g_1(l(x_1), r(x_1), x_2, \dots, x_{n+2}) \\
 F_r(x_1, \dots, x_{n+2}) &= g_2(r(x_1), l(x_1), x_2, \dots, x_{n+2})
 \end{aligned}$$

Con un poco de cuidado, se puede ver que F_r y F_l se pueden obtener por composicion de funciones PR y usando muchas funciones proyectoras para ubicar los parametros en los lugares correctos. Entonces F_r y F_l son Pr. ☺

Entonces $H(x_1, \dots, x_n, t+1) = F(H(x_1, \dots, x_n, t), x_1, \dots, x_n, t)$ siguiendo el esquema de recursion primitiva, por lo tanto H es PR. Y de esto sigue que h_1 y h_2 son PR. ☺

1.1 Random Examples

Definition 1.1.1: Limit of Sequence in \mathbb{R}

Let $\{s_n\}$ be a sequence in \mathbb{R} . We say

$$\lim_{n \rightarrow \infty} s_n = s$$

where $s \in \mathbb{R}$ if \forall real numbers $\epsilon > 0 \exists$ natural number N such that for $n > N$

$$s - \epsilon < s_n < s + \epsilon \text{ i.e. } |s - s_n| < \epsilon$$

Question 12

Is the set $x\text{-axis} \setminus \{\text{Origin}\}$ a closed set

Solution: We have to take its complement and check whether that set is a open set i.e. if it is a union of open balls

Note:-

We will do topology in Normed Linear Space (Mainly \mathbb{R}^n and occasionally \mathbb{C}^n) using the language of Metric Space

Claim 1.1.1 Topology

Topology is cool

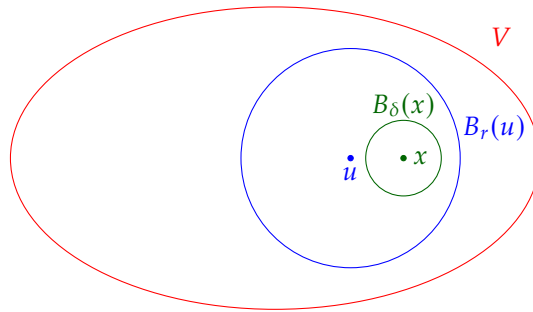
Example 1.1.1 (Open Set and Close Set)

- Open Set:
- ϕ
 - $\bigcup_{x \in X} B_r(x)$ (Any $r > 0$ will do)
 - $B_r(x)$ is open
- Closed Set:
- \overline{X}, ϕ
 - $\overline{B_r(x)}$
 - $x\text{-axis} \cup y\text{-axis}$

Theorem 1.1.1

If $x \in$ open set V then $\exists \delta > 0$ such that $B_\delta(x) \subset V$

Proof: By openness of V , $x \in B_r(u) \subset V$



Given $x \in B_r(u) \subset V$, we want $\delta > 0$ such that $x \in B_\delta(x) \subset B_r(u) \subset V$. Let $d = d(u, x)$. Choose δ such that $d + \delta < r$ (e.g. $\delta < \frac{r-d}{2}$)

If $y \in B_\delta(x)$ we will be done by showing that $d(u, y) < r$ but

$$d(u, y) \leq d(u, x) + d(x, y) < d + \delta < r$$

☺

Corollary 1.1.1

By the result of the proof, we can then show...

Lemma 1.1.1

Suppose $\vec{v}_1, \dots, \vec{v}_n \in \mathbb{R}^n$ is subspace of \mathbb{R}^n .

Proposition 1.1.1

$1 + 1 = 2$.

1.2 Random

Definition 1.2.1: Normed Linear Space and Norm $\|\cdot\|$

Let V be a vector space over \mathbb{R} (or \mathbb{C}). A norm on V is function $\|\cdot\| : V \rightarrow \mathbb{R}_{\geq 0}$ satisfying

- ① $\|x\| = 0 \iff x = 0 \ \forall x \in V$
- ② $\|\lambda x\| = |\lambda| \|x\| \ \forall \lambda \in \mathbb{R}(\text{or } \mathbb{C}), x \in V$
- ③ $\|x + y\| \leq \|x\| + \|y\| \ \forall x, y \in V$ (Triangle Inequality/Subadditivity)

And V is called a normed linear space.

• Same definition works with V a vector space over \mathbb{C} (again $\|\cdot\| \rightarrow \mathbb{R}_{\geq 0}$) where ② becomes $\|\lambda x\| = |\lambda| \|x\|$ $\forall \lambda \in \mathbb{C}, x \in V$, where for $\lambda = a + ib$, $|\lambda| = \sqrt{a^2 + b^2}$

Example 1.2.1 (p -Norm)

$V = \mathbb{R}^m$, $p \in \mathbb{R}_{\geq 0}$. Define for $x = (x_1, x_2, \dots, x_m) \in \mathbb{R}^m$

$$\|x\|_p = \left(|x_1|^p + |x_2|^p + \dots + |x_m|^p \right)^{\frac{1}{p}}$$

(In school $p = 2$)

Special Case $p = 1$: $\|x\|_1 = |x_1| + |x_2| + \dots + |x_m|$ is clearly a norm by usual triangle inequality.

Special Case $p \rightarrow \infty$ (\mathbb{R}^m with $\|\cdot\|_\infty$): $\|x\|_\infty = \max\{|x_1|, |x_2|, \dots, |x_m|\}$

For $m = 1$ these p -norms are nothing but $|x|$. Now exercise

Question 13

Prove that triangle inequality is true if $p \geq 1$ for p -norms. (What goes wrong for $p < 1$?)

Solution: For Property ③ for norm-2

When field is \mathbb{R} :

We have to show

$$\begin{aligned} \sum_i (x_i + y_i)^2 &\leq \left(\sqrt{\sum_i x_i^2} + \sqrt{\sum_i y_i^2} \right)^2 \\ \implies \sum_i (x_i^2 + 2x_i y_i + y_i^2) &\leq \sum_i x_i^2 + 2\sqrt{\left[\sum_i x_i^2 \right] \left[\sum_i y_i^2 \right]} + \sum_i y_i^2 \\ \implies \left[\sum_i x_i y_i \right]^2 &\leq \left[\sum_i x_i^2 \right] \left[\sum_i y_i^2 \right] \end{aligned}$$

So in other words prove $\langle x, y \rangle^2 \leq \langle x, x \rangle \langle y, y \rangle$ where

$$\langle x, y \rangle = \sum_i x_i y_i$$

Note:-

- $\|x\|^2 = \langle x, x \rangle$
- $\langle x, y \rangle = \langle y, x \rangle$

- $\langle \cdot, \cdot \rangle$ is \mathbb{R} -linear in each slot i.e.

$$\langle rx + x', y \rangle = r\langle x, y \rangle + \langle x', y \rangle \text{ and similarly for second slot}$$

Here in $\langle x, y \rangle$ x is in first slot and y is in second slot.

Now the statement is just the Cauchy-Schwartz Inequality. For proof

$$\langle x, y \rangle^2 \leq \langle x, x \rangle \langle y, y \rangle$$

expand everything of $\langle x - \lambda y, x - \lambda y \rangle$ which is going to give a quadratic equation in variable λ

$$\begin{aligned} \langle x - \lambda y, x - \lambda y \rangle &= \langle x, x - \lambda y \rangle - \lambda \langle y, x - \lambda y \rangle \\ &= \langle x, x \rangle - \lambda \langle x, y \rangle - \lambda \langle y, x \rangle + \lambda^2 \langle y, y \rangle \\ &= \langle x, x \rangle - 2\lambda \langle x, y \rangle + \lambda^2 \langle y, y \rangle \end{aligned}$$

Now unless $x = \lambda y$ we have $\langle x - \lambda y, x - \lambda y \rangle > 0$ Hence the quadratic equation has no root therefore the discriminant is greater than zero.

When field is \mathbb{C} :

Modify the definition by

$$\langle x, y \rangle = \sum_i \bar{x}_i y_i$$

Then we still have $\langle x, x \rangle \geq 0$

1.3 Algorithms

Algorithm 1: what

Input: This is some input

Output: This is some output

/ This is a comment */*

```

1 some code here;
2  $x \leftarrow 0$ ;
3  $y \leftarrow 0$ ;
4 if  $x > 5$  then
5   |  $x$  is greater than 5 ;                                // This is also a comment
6 else
7   |  $x$  is less than or equal to 5;
8 end
9 foreach  $y$  in 0..5 do
10  |  $y \leftarrow y + 1$ ;
11 end
12 for  $y$  in 0..5 do
13  |  $y \leftarrow y - 1$ ;
14 end
15 while  $x > 5$  do
16  |  $x \leftarrow x - 1$ ;
17 end
18 return Return something here;
```
