Some Class
Random Examples

Your Name

Contents

Chapter 1

Question 1

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

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

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

Ya que $s(n(x))$ es composicion 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 composicion, es primitiva recursiva.

Question 2

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

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

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

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

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

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

Pero para que cierre la aridad con el esquema de recursion primitiva, debemos encontrar una funcion 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:

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

Solution: $f_2(x,y) = x.y$

$$
\begin{aligned}
&f_2(x,0) = n(x)\\
&f_2(x,y) = f_2(x,y-1) + x = f_1(f_2(x,y-1),x)\\
&f_2(x,y) = f1(u_1^3(f_2(x,y-1),x,y-1),\ u_2^3(f_2(x,y-1),x,y-1))\\
&f_2(x,y) = g(f_2(x,y-1),x,y-1)
\end{aligned}
$$

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 coposicion de funciones PR, entonces tambien es PR.

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

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

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 coposicion de funciones PR, entonces tambien es PR.

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

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

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 coposicion de funciones PR, entonces tambien es PR.

Solution: $g_1(x) = x \dot{-} 1$

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

Solution: $g_2(x,y) = x \dot{-} y$

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

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

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

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

Las funciones α (negacion) y $\leqslant$ 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: $g_4(x, y) = \min\{x, y\}$

$$\min\{x,y\} = (x \leqslant y).x + \alpha(x \leqslant y).y$$
$$\min\{x,y\} = f_2((x \leqslant y).x) + f_2(\alpha(x \leqslant y), y)$$
$$\min\{x,y\} = f_1(\underbrace{f_2((x \leqslant y), y)}_{h_1}, \underbrace{f_2(\alpha(x \leqslant y), y)}_{h_2})$$

Las funciones α (negacion) y $\leqslant$ 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:

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

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

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

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

Solution: a)

$(\to)$

Vamos a usar induccion estructural

Casos Base: Veamos que se cumple para las funciones iniciales

$$n(x) = 0, \quad k = 0$$
$$s(x) = x + 1, \quad k = 1$$
$$u_i^n(x_1, ..., x_n) = x_i + 0, \quad k = 0$$

Paso inductivo:

Sea $f \in C$, $f(x_1, \ldots, x_n) = h(g_1(x1, \ldots, x_n), \ldots, g_n(x1, \ldots, x_n))$. Veamos que o bien $f(x_1, \ldots, x_n) = k$, o bien $f(x_1, \ldots, x_n) = x_i + k$

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

Caso $h(x1, \ldots, x_n) = k$: Entonces tenemos que $f(x1, \ldots, x_n) = k = k'$, ya esta!

Caso $h(x1, \ldots, x_n) = x_i + k'$: Entonces tenemos que $f(x1, \ldots, x_n) = g_i(x1, \ldots, x_n)$.

Ahora tenemos 2 sub-casos mas:

Caso $g_i(x1, \ldots, x_n) = k''$: $f(x1, \ldots, x_n) = k' + k'' = k$

Caso $g_i(x1, \ldots, x_n) = x_j + k''$: $f(x1, \ldots, x_n) = x_j + k'' + k' = x_j + k$

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 $f(x_1, \ldots, x_n) = h_k(u_1^n(x_1, \ldots, x_n)) = k$, xlt $f \in C$

Se puede defnir la funcion $s_k(x) = x + k$ usando sucesivamente la funcion s, por composicion $s_k \in C$. Xlt $f(x_1, ..., x_n) = x_i + k = s_k(u_i^n(x_1, ..., x_n))$. Y esta ultima al ser composicion de funciones de C, nos asegura que f tambien esta en C.

Question 4

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

Solution:

- $\leqslant (x, y) = \alpha(x - y)$
- $\geqslant (x, y) = \alpha(y - x)$
- $= (x, y) = (x \leqslant y).(y \leqslant x)$
- $\neq (x, y) = \alpha(x = y)$
- $< (x, y) = \alpha(x \geqslant y)$
- $> (x, y) = \alpha(x \leqslant y)$

Si una funcion f esta en cualquier clase PRC, entonces esta en la interseccion de todas las clases PRC, por el teorema visto en clases, podemos decir que ser funcion primitiva recursiva es condicion necesaria y suficiente para esto. En el ejericio 2 vimos que la funcion de multiplicacion, resta de naturales y la suma son primitivas recursivas. Probemos que $\alpha(x)$ es primitiva recursiva.

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

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

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

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

Question 5

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

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

Observar que h queda completamente determinada por el esquema.

Solution:

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

Notemos que $g(x_1,\ldots,x_n).\alpha(p_1(x_1,\ldots,x_n)+\ldots+p_k(x_1,\ldots,x_n))$ 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 ejericio 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

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

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

Claim 1.0.2

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

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

Claim 1.0.3

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

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

Claim 1.0.4

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

Proof: Sean:
$f_i'(x_1,\ldots,x_n)=f_i(x_1,\ldots,x_n).p_i(x_1,\ldots,x_n)\in C,\forall i:1\ldots k$
$g'(x_1,\ldots,x_n)=g(x_1,\ldots,x_n).\alpha(p_1(x_1,\ldots,x_n)+\ldots+p_k(x_1,\ldots,x_n))\in C$

Escribamos a h como composicion de funciones en C
$h(x_1,\ldots,x_n)=a_{k+1}(f_1'(x_1,\ldots,x_n),\ldots,f_k'(x_1,\ldots,x_n),g'(x_1,\ldots,x_n))$.
$\therefore h\in C$.

😄

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 $f : \mathbb{N}^n \to \mathbb{N}$ y $g_1, g_2 : \mathbb{N}^{n+2} \to \mathbb{N}$ funciones en C. Mostrar que también está en C cualquier h que cumpla:

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

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 funcion primitiva recursiva, veamos que $\text{par}(x)$ cumple esto.

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

Ya que pudimos definir a la funcion par por recursion primitiva, entonces par es primitiva recursiva. $\therefore$ par esta en toda clase PRC. 😃

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

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

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

Entonces f es primitiva recursiva y por lo tanto esta en toda clase PRC. 😃

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

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 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 encontar una funcion $g \in C$, 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.

$$g(u^{n+2}_{n+2}(x_1,\dots,x_{n+2}), u^{n+2}_1(x_1,\dots,x_{n+2}),\dots,u^{n+2}_n(x_1,\dots,x_{n+2}), e(x_1,\dots,x_{n+2}))$$

Donde:

$$d(\text{x}) = 2.\text{x} + \text{impar}(\text{x})$$
$$e(x_1,\dots,x_{n+2}) = d(u^{n+2}_{n+1}(x_1,\dots,x_{n+2}))$$

Se puede probar que d y e son funciones primitivas recursivas, por lo tanto, estan en 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 C. $\therefore h \in C$. 😊

Question 7

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

$$\text{cantidad}_p(x_1,\ldots,x_n,y,z) = |\{t \mid y \leqslant t \leqslant z \wedge p(x_1,\ldots,x_n,t)\}|$$

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

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

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

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

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

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

Question 8

Question 9

Question 10

Question 11

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

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

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 $x-$axis$\setminus\{$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

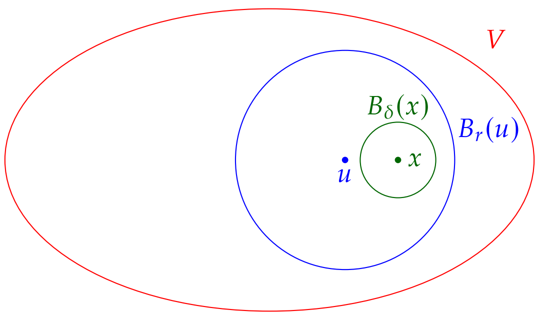
Closed Set:
- $X,\ \phi$
- $\overline{B_r(x)}$

$x-$axis $\cup$ $y-$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

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

😄

Corollary 1.1.1

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

Lenma 1.1.1

Suppose $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^n$ 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 $\|\cdot\|\ V \longrightarrow \mathbb{R}_{\geqslant 0}$ 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) $\|x+y\| \leqslant \|x\| + \|y\|\ \forall\ x, y \in V$ (Triangle Inequality/Subadditivity)

And V is called a normed linear space.
• Same definition works with V a vector space over $\mathbb{C}$ (again $\|\cdot\| \longrightarrow \mathbb{R}_{\geqslant 0}$) 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)
$V = \mathbb{R}^m$, $p \in \mathbb{R}_{\geqslant 0}$. Define for $x = (x_1, x_2, \cdots, x_m) \in \mathbb{R}^m$

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

(In school $p = 2$)

Special Case $p = 1$: $\|x\|_1 = |x_1| + |x_2| + \cdots + |x_m|$ is clearly a norm by usual triangle inequality.
Special Case $p \to \infty$ ($\mathbb{R}^m$ with $\|\cdot\|_\infty$): $\|x\|_\infty = \max\{|x_1|, |x_2|, \cdots, |x_m|\}$
For $m = 1$ these $p-$norms are nothing but $|x|$. Now exercise

Question 13

Prove that triangle inequality is true if $p \geqslant 1$ 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

$$\sum_i (x_i + y_i)^2 \leqslant \left(\sqrt{\sum_i x_i^2} + \sqrt{\sum_i y_i^2}\right)^2$$

$$\implies \sum_i (x_i^2 + 2x_iy_i + y_i^2) \leqslant \sum_i x_i^2 + 2\sqrt{\left[\sum_i x_i^2\right]\left[\sum_i y_i^2\right]} + \sum_i y_i^2$$

$$\implies \left[\sum_i x_iy_i\right]^2 \leqslant \left[\sum_i x_i^2\right]\left[\sum_i y_i^2\right]$$

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

$$\langle x, y\rangle = \sum_i x_iy_i$$

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

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

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

$$\langle x, y \rangle = \sum_i \overline{x_i} y_i$$

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

1.3 Algorithms

Algorithm 1: what

Input: This is some input
Output: This is some output
`/* This is a comment */`

some code here;
$x \leftarrow 0$;
$y \leftarrow 0$;
if $x > 5$ **then**
  x is greater than 5 ; `// This is also a comment`
else
  x is less than or equal to 5;
end
foreach *y in 0..5* **do**
  $y \leftarrow y + 1$;
end
for y *in* 0..5 **do**
  $y \leftarrow y - 1$;
end
while $x > 5$ **do**
  $x \leftarrow x - 1$;
end
return Return something here;