

Práctico 1

Inducción y recursión

Ejercicio 1 (Reglas sobre reales)

Considere el conjunto \mathbb{R} de los números reales; la lista de subconjuntos de \mathbb{R} de la columna izquierda; y los conjuntos de reglas de la columna derecha.

A. \mathbb{N}	1. I $2 \in S$ II Si $n \in S$, entonces $n + 2 \in S$
B. \mathbb{Z}	
C. $\{2n + 1 : n \in \mathbb{N}\}$	2. I $3 \in S$ II Si $n \in S$, entonces $n + 1 \in S$ III Si $n \in S$, entonces $n - 1 \in S$
D. $\{\pi + k : k \in \mathbb{Z}\}$	
E. \emptyset	
F. \mathbb{R}	3. I Si $n \in S$, entonces $n + 1 \in S$

a. Indique cuáles subconjuntos satisfacen cuáles conjuntos de reglas.

Por ejemplo, para investigar si el conjunto A. cumple el conjunto de reglas 1. es necesario preguntarse si se cumplen las siguientes reglas:

- I $2 \in \mathbb{N}$
- II Si $n \in \mathbb{N}$, entonces $n + 2 \in \mathbb{N}$

b. Para cada conjunto de reglas indique qué conjunto definen.

Ejercicio 2 (Inducción sobre los naturales)

a. Enuncie el principio de inducción primitiva para el conjunto *Par* definido inductivamente por las siguientes cláusulas:

- I $0 \in Par$
- II Si $n \in Par$ entonces $n + 2 \in Par$

b. Pruebe utilizando este principio, que para todo $n \in Par$, existe $m \in \mathbb{N}$ tal que $n = m + m$.

Ejercicio 3 (Inducción sobre los naturales)

Considere el conjunto inductivo *F* definido mediante las siguientes reglas:

- I $\langle 0, 1 \rangle \in F$
- II Si $\langle n, m \rangle \in F$, entonces $\langle m, n + m \rangle \in F$

Demostrar que

$$(\forall n \in \mathbb{N}) \langle \text{Fibo}(n), \text{Fibo}(n + 1) \rangle \in F$$

donde la función *Fibo* se define como:

$$\begin{aligned} \text{Fibo} : \mathbb{N} &\rightarrow \mathbb{N} \\ \text{Fibo}(0) &= 0 \\ \text{Fibo}(1) &= 1 \\ \text{Fibo}(n + 2) &= \text{Fibo}(n) + \text{Fibo}(n + 1) \end{aligned}$$

Ejercicio 4 (Reglas sobre palabras)

Considere el alfabeto Σ formado por las letras a y b ; la lista de subconjuntos de Σ^* de la columna izquierda; y los conjuntos de reglas de la columna derecha.

A. Σ	1. I $\varepsilon \in S$ II Si $w \in S$, entonces $awa \in S$
B. $\{w \in \Sigma^* : w \text{ tiene largo par}\}$	
C. $\{w \in \Sigma^* : w \text{ termina en } a\}$	2. I $a \in S$ II $b \in S$ III Si $w \in S$, entonces $ww \in S$
D. $\{w \in \Sigma^* : w \text{ es un palíndromo}\}$	
E. \emptyset	
F. Σ^*	3. I Si $w \in S$, entonces $wa \in S$

- Indique cuáles subconjuntos satisfacen cuáles conjuntos de reglas.
- Indique qué conjunto define cada conjunto de reglas.

Ejercicio 5 (Principio de inducción primitiva)

Sea un alfabeto $\Sigma = \{\bullet, \circ, \Delta, \square\}$ y el conjunto PAL definido inductivamente por las siguientes reglas:

- I $\varepsilon \in PAL$
- II Si $x \in \Sigma$, entonces $x \in PAL$
- III Si $w \in PAL$ y $x \in \Sigma$, entonces $xwx \in PAL$

- Proporcione 3 elementos de PAL .
- Proporcione 3 elementos de Σ^* que no pertenezcan a PAL .
- Enuncie el principio de inducción primitiva para PAL .

Ejercicio 6 (Lenguajes)

Considere el alfabeto $\Sigma = \{a, b, c\}$, y los lenguajes Δ y Γ definidos inductivamente con las siguientes reglas.

I $\varepsilon \in \Gamma$ II $a \in \Gamma$ III Si $\alpha \in \Gamma$ y $\beta \in \Gamma$, entonces $b\alpha c\beta b \in \Gamma$	I $\varepsilon \in \Delta$ II Si $\alpha \in \Delta$, entonces $babc \in \Delta$ III Si $\alpha \in \Delta$, entonces $b\alpha b a \in \Delta$
---	--

- Encuentre palabras de Σ^* que no pertenezcan a Γ . Análogo para Δ .
- Muestre que Γ no está incluido en Δ y que Δ tampoco está incluido en Γ .

Para probar que un lenguaje no está incluido en otro debe proporcionar una palabra que pertenezca al primer lenguaje y no pertenezca al segundo, con las justificaciones que correspondan.

Ejercicio 7 (Lenguajes)

Considere el alfabeto $\Sigma = \{a, b, c\}$, y los lenguajes Δ y Γ definidos en el Ejercicio 6. Demuestre, usando el principio de inducción que corresponda, que

- a. el largo de las palabras de Δ es múltiplo de tres.
- b. todas las palabras de Δ tiene una cantidad par de ocurrencias de la letra b .
- c. todas las palabras de Γ tiene una cantidad par de ocurrencias de la letra b .
- d. en Γ no hay palabras de largo dos.

Ejercicio 8 (Definiciones inductivas)

Cada ítem describe un lenguaje sobre el alfabeto $\Sigma = \{a, b\}$. Se le pide la definición inductiva de cada lenguaje.

L1: $\{\varepsilon, a, aa, aaa, aaaa, \dots\}$

L2: $\{\varepsilon, ab, aabb, aaabbb, aaaabbbb, \dots\}$

L3: $\{b, aba, aabaa, aaabaaa, aaaabaaaa, \dots\}$

L4: $\{\varepsilon, ab, abab, ababab, abababab, \dots, ba, baba, bababa, babababa, \dots\}$

L5: $\{\varepsilon, a, ab, aba, abab, ababa, ababab, \dots\}$

L6: $\{\alpha \in \Sigma^* : \alpha \text{ es un palíndromo}\}$

Ejercicio 9 (Inserción a derecha)

Considere el alfabeto $\Sigma = \{a, b\}$, y los lenguajes Δ y Γ definidos inductivamente con las siguientes reglas. Demuestre que $\Gamma = \Delta$.

<p>I $\varepsilon \in \Gamma$</p> <p>II Si $\alpha \in \Gamma$, entonces $a\alpha \in \Gamma$</p> <p>III Si $\alpha \in \Gamma$, entonces $b\alpha \in \Gamma$</p>	<p>I $\varepsilon \in \Delta$</p> <p>II Si $\alpha \in \Delta$, entonces $\alpha a \in \Delta$</p> <p>III Si $\alpha \in \Delta$, entonces $\alpha b \in \Delta$</p>
---	---

Ejercicio 10 (Relaciones inductivas)

Considere la siguiente definición inductiva de la relación $S \subseteq \mathbb{N} \times \mathbb{N}$:

- I Si $n \in \mathbb{N}$ entonces $\langle n, n \rangle \in S$
- II Si $\langle n, m \rangle \in S$ entonces $\langle n, m + 1 \rangle \in S$

a. Indique cuáles de las siguientes afirmaciones son correctas y justifique su respuesta usando la definición de S .

$$\langle 0, 0 \rangle \in S \quad 0 \in S \quad \langle \pi, \pi \rangle \in S \quad \langle 2, 3 \rangle \in S \quad \langle 3, 2 \rangle \in S$$

b. Enuncie el principio de inducción primitiva para S .

c. Considere la siguiente definición inductiva de la relación $Q \subseteq \mathbb{N} \times \mathbb{N}$:

- I Si $n \in \mathbb{N}$ entonces $\langle 0, n \rangle \in Q$
- II Si $\langle n, m \rangle \in Q$ entonces $\langle n + 1, m + 1 \rangle \in Q$

Demuestre que $S = Q$. Para esto demuestre que $S = T$ y $Q = T$

$$\text{con } T = \{ \langle n, m \rangle / \langle n, m \rangle \in \mathbb{N} \times \mathbb{N}, n \leq m \} = \{ \langle n, n + m \rangle / n, m \in \mathbb{N} \}$$

Ejercicio 11 (Recursión en los naturales)

Considere la siguiente función definida por recursión primitiva en \mathbb{N} :

$$\begin{aligned} f : \mathbb{N} &\rightarrow \mathbb{N} \\ f(0) &= 0 \\ f(n + 1) &= f(n) + 2 \times (n + 1) \end{aligned}$$

Pruebe que $(\forall n \in \mathbb{N}) f(n) = n \times (n + 1)$.

Ejercicio 12 (Funciones sobre tiras)

Considere un alfabeto $\Sigma = \{a, b, c, d\}$

a. Defina las siguientes funciones.

$\text{largo} : \Sigma^* \rightarrow \mathbb{N}$	$\text{largo} : \Sigma^+ \rightarrow \mathbb{N}$	$\text{resto} : \Sigma^+ \rightarrow \Sigma^*$
$\text{duplicar} : \Sigma^* \rightarrow \Sigma^*$	$\text{último} : \Sigma^+ \rightarrow \Sigma$	$\text{duplicar} : \Sigma^+ \rightarrow \Sigma^+$
$\text{invertir} : \Sigma^* \rightarrow \Sigma^*$	$\text{principio} : \Sigma^+ \rightarrow \Sigma^*$	$\text{invertir} : \Sigma^+ \rightarrow \Sigma^+$
	$\text{primero} : \Sigma^+ \rightarrow \Sigma$	

Observe que el dominio de las funciones de las columnas a la derecha no considera la palabra ε . Las funciones cuya aplicación a la palabra vacía no tienen sentido¹ tienen como dominio a Σ^+ .

Los siguientes ejemplos aclaran el significado de las funciones pedidas.

$\text{largo}(abcdc) = 5$	$\text{primero}(abcdc) = a$	$\text{principio}(abcdc) = abcd$
$\text{resto}(abcdc) = bc dc$	$\text{último}(abcdc) = c$	$\text{invertir}(abcdc) = cdcba$
$\text{duplicar}(abc) = aabbcc$		

b. Demuestre inductivamente que el largo de una palabra duplicada es el doble de la palabra original.

¹En algunos casos, podemos totalizar la función: por ejemplo, podemos definir $\text{principio}(\varepsilon) := \varepsilon$. Pero en ocasiones esto no es posible: ¿Cuál sería la primera letra de ε ? ¿Y la última?

Ejercicio 13 (Concatenación)

Considere la siguiente definición de concatenación:

$$\begin{aligned} \text{append} &: \Sigma^* \times \Sigma^* \rightarrow \Sigma^* \\ \text{append}(\varepsilon, w) &= w \\ \text{append}(au, w) &= a\text{append}(u, w) \end{aligned}$$

- a. Demuestre que ε es neutro de esta función, es decir:

$$(\forall u \in \Sigma^*) \text{append}(\varepsilon, u) = u \text{ y } (\forall u \in \Sigma^*) \text{append}(u, \varepsilon) = u$$

- b. Demuestre que el largo de concatenar dos tiras es la suma de sus largos.

Ejercicio 14

Considere el alfabeto $\Sigma = \{a, b, c\}$.

- a. Defina siguiendo el esquema de recursión primitiva la función **invertir**: $\Sigma^* \rightarrow \Sigma^*$ (ej: **invertir**(*abc*): = *cba*)
- b. Demuestre usando inducción en Σ^* que $(\forall w \in \Sigma^*) \text{invertir}(wx) = x \text{invertir}(w)$
- c. Defina de forma inductiva el conjunto $A \subseteq \Sigma^*$ de las tiras palíndromas (se leen igual de izquierda a derecha que de derecha a izquierda).
- d. Demuestre usando inducción en A que $(\forall w \in A) \text{invertir}(w) = w$

Ejercicio 15 (Más funciones sobre tiras)

- a. Identifique la siguiente función.

$$\begin{aligned} f &: \Sigma^* \times \Sigma \rightarrow \{0, 1\} \\ f(\varepsilon, x) &= 0 \\ f(xw, y) &= \text{si } x = y \text{ entonces } 1 \text{ sino } f(w, y) \end{aligned}$$

- b. Defina usando el esquema de recursión primitiva adecuado las siguientes funciones.

$$\begin{aligned} \text{cant} &: \Sigma^* \times \Sigma \rightarrow \mathbb{N} \\ \text{copiar} &: \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \\ \text{sacar_de_la_izquierda} &: \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \\ \text{primera_posición} &: \Sigma^* \times \Sigma \rightarrow \mathbb{N} \end{aligned}$$

Los siguientes ejemplos aclaran el significado de las funciones pedidas.

$$\begin{aligned} \text{cant}(abcdc, c) &= 2 & \text{copiar}(ab, 3) &= ababab \\ \text{sacar_de_la_izquierda}(abcdc, 3) &= dc & \text{primera_posición}(abcdc, c) &= 3 \\ \text{sacar_de_la_izquierda}(abcdc, 7) &= \varepsilon & \text{primera_posición}(abcdc, e) &= 6 \end{aligned}$$

Ejercicio 16 (Orden)

Considere un alfabeto Σ enriquecido con una relación de orden \leq .

- a. Defina la función **entre** : $\Sigma^* \times \Sigma \times \Sigma \rightarrow \{0, 1\}$ que devuelve 1 si y solamente si todas las letras de su primer argumento están entre las letras dadas. Por ejemplo, considerando las letras del alfabeto castellano con el orden usual, **entre**(jose, *b, t*) = 1 y **entre**(mama, *b, z*) = 0.

- b. Defina la función **insord** : $\Sigma^* \times \Sigma \rightarrow \Sigma^*$ que inserta ordenadamente una nueva letra en una tira ya ordenada. Por ejemplo, **insord**(*abcde*, *d*) = *abcdde*. Indique qué hace su función si la lista de entrada se encuentra desordenada.

Ejercicio 17 (Reemplazo)

Se le llama función de Alto Orden a una función que recibe como parámetro o devuelve como resultado otra función. Para definir funciones de esta clase, uno de los parámetros se usa como una función en su definición.

Ejemplo:

$$f(g, x) = g(x) + 1$$

entonces:

$$f(\text{succ}, 1) = 3, f(^2, 5) = 26$$

Consideremos una función $f : \Sigma \rightarrow \Sigma^*$. Decimos que w se obtiene por reemplazo a partir de u usando f , cuando w es el resultado de reemplazar cada letra de u de acuerdo a la función f .

- a. Defina la función **reemplazo** : $\Sigma^* \times (\Sigma \rightarrow \Sigma^*) \rightarrow \Sigma^*$. Esa función debe cumplir que, si $\Sigma = \{a, b\}$ y f es tal que $f(a) = aba$ y $f(b) = bba$, entonces **reemplazo**(*abab*, f) = *ababbaababba*.
- b. Encuentre una función f adecuada para poder probar la siguiente propiedad, con **duplicar** definida en Ejercicio 12:

$$(\forall w \in \Sigma^*) \text{duplicar}(w) = \text{reemplazo}(w, f)$$

Ejercicio 18

- a. Defina el conjunto $\mathcal{L}_{\mathbb{N}}$ de las listas de naturales con el símbolo | como separador y [] como la lista vacía. Use a los naturales como conjunto base cuando sea necesario.

Ejs: [] es la lista vacía. 15|2|31|[] es una lista que contiene al 15, 2 y 31 en ese orden.

- b. Defina la función **Reduce** : $(\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}) \times \mathbb{N} \times \mathcal{L}_{\mathbb{N}} \rightarrow \mathbb{N}$. Esta función recibe una función binaria de naturales en naturales, un natural y una lista, devolviendo un natural que es la aplicación de la función sobre todos los elementos de la lista. El natural es el valor a devolver en el caso de la lista vacía y usualmente es el neutro de la operación.

Ejemplos
$Reduce(*, 1, []) = 1$
$Reduce(*, 1, 2 3 4 []) = 2 * 3 * 4 * 1 = 24$
$g(x, y) = x * (y + 1)$
$Reduce(g, 0, 2 3 4 []) = g(2, g(3, g(4, 0))) = 32$

- c. Demuestre que $f(l) = Reduce(+, 0, l)$ devuelve la suma de todos los naturales contenidos en la lista l o 0 si la lista es vacía.

Ejercicio 19 (Números binarios)

Considere $\Sigma = \{0, 1\}$ y las siguientes definiciones inductivas del conjunto de los números binarios B_D y B_I .

<p>I Si $x \in \Sigma$, entonces $x \in B_D$</p> <p>II Si $w \in B_D$ y $x \in \Sigma$, entonces $xw \in B_D$</p>	<p>I Si $x \in \Sigma$, entonces $x \in B_I$</p> <p>II Si $w \in B_I$ y $x \in \Sigma$, entonces $wx \in B_I$</p>
--	--

- a. Defina recursivamente la función $eval : B_D \rightarrow \mathbb{N}$ que devuelve el natural representado por un número binario. Es decir, $eval(10100) = 20$ y $eval(000100) = 4$.
- b. Defina recursivamente la función $eval : B_I \rightarrow \mathbb{N}$ que devuelve el natural representado por un número binario. Es decir, $eval(10100) = 20$ y $eval(000100) = 4$. Para resolver este problema, recuerde que el binario 10100 representa $1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$ (O lo que es lo mismo por la regla de Horner, $((1 \times 2 + 0) \times 2 + 1) \times 2 + 0$).
- c. Defina recursivamente la función $par : B_I \rightarrow \{0, 1\}$ que devuelve 1 si la cantidad de unos en el número es par, y 0 si es impar.

Ejercicio 20

Considere los siguientes conjuntos:

- $P = \{p_i \mid i \in \mathbb{N}\}$
- $\Sigma = P \cup \{\oplus, \otimes, \}, \{\}$
- El conjunto \mathcal{L} definido inductivamente por las siguientes reglas:
 - I Si $p_i \in P$, entonces $p_i \in \mathcal{L}$
 - II Si $\varphi \in \mathcal{L}$ y $\psi \in \mathcal{L}$, entonces $(\varphi \oplus \psi) \in \mathcal{L}$
 - III Si $\varphi \in \mathcal{L}$ y $\psi \in \mathcal{L}$, entonces $(\varphi \otimes \psi) \in \mathcal{L}$

- a. Dada una función:

$$v_1 : P \rightarrow \{0, 1\}$$

$$v_1(p_i) = \text{si } i \text{ es par entonces } 1 \text{ sino } 0$$

Por ejemplo: $v_1(p_0) = 1$ y $v_1(p_1) = 0$

Defina una función recursiva $eval_1 : \mathcal{L} \rightarrow \{0, 1\}$, considerando que se evalúa \oplus como el máximo, \otimes como el mínimo, y cada $p_i \in P$ como $v_1(p_i)$.

Por ejemplo: $eval_1(((p_0 \otimes p_1) \oplus p_0)) = 1$

- b. Defina una función recursiva $eval_2 : \mathcal{L} \times (P \rightarrow \{0, 1\}) \rightarrow \{0, 1\}$, donde se agrega la interpretación de los elementos de P a la función $eval_1$.

Por ejemplo: $eval_2(((p_0 \otimes p_1) \oplus p_0), v_1) = 1$

- c. Indique si las siguientes afirmaciones son verdaderas o falsas para cualquier función v :

$$(\forall \varphi, \psi \in \mathcal{L}) eval_2((\varphi \oplus \psi), v) = 1 \Rightarrow eval_2(\varphi, v) = 1$$

$$(\forall \varphi, \psi \in \mathcal{L}) eval_2((\varphi \oplus \psi), v) = 1 \Leftarrow eval_2(\varphi, v) = 1$$

$$(\forall \varphi, \psi \in \mathcal{L}) eval_2((\varphi \otimes \psi), v) = 1 \Rightarrow eval_2(\varphi, v) = 1$$

$$(\forall \varphi, \psi \in \mathcal{L}) eval_2((\varphi \otimes \psi), v) = 1 \Leftarrow eval_2(\varphi, v) = 1$$

Ejercicio 21 (Otros esquemas)

Considere las siguientes listas de ecuaciones²

$g(0) = 1$	$h(0) = 1$	$j(0) = 1$
$g(1) = 0$	$h(1) = 0$	$j(2n + 2) = 2 \times j(n) - 1$
$g(n + 2) = g(n) + \min \{g(n), g(n + 1)\}$	$h(n + 2) = h((n + 2) \bmod 2)$	$j(2n + 1) = 2 \times j(n) - 1$

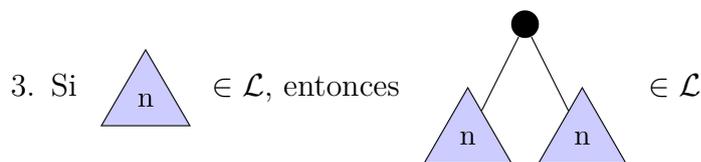
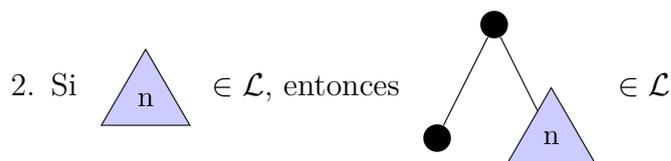
- a. Determine si son formas correctas de definir funciones. Justifique.
- b. Pruebe que las primeras dos columnas definen una misma función probando que cada una de ellas es igual a la siguiente función:

$$\begin{aligned}
 f : \mathbb{N} &\rightarrow \{0, 1\} \\
 f(2n) &= 1 \\
 f(2n + 1) &= 0
 \end{aligned}$$

Ejercicio 22 (Árboles binarios)

Considere las siguientes reglas de construcción de árboles binarios (sin información):

1. $\bullet \in \mathcal{L}$



Se definen inductivamente las siguientes lenguajes:

- el lenguaje N_1 , definido inductivamente por las cláusulas 1 y 2.
- el lenguaje N_2 , definido inductivamente por las cláusulas 1 y 3.
- el lenguaje N_3 , definido inductivamente por las cláusulas 1 y 2 y 3.
- el lenguaje N_4 , definido inductivamente por las cláusulas 2 y 3.

²Escribimos $n \bmod m$ para indicar el resto de la división de n entre m .

- a. Indique si las siguientes afirmaciones son verdaderas o falsas:
- I. El lenguaje N_1 está incluido en el lenguaje N_3 .
 - II. El lenguaje N_2 está incluido en el lenguaje N_4 .
 - III. El lenguaje N_3 está incluido en el lenguaje N_1 .
 - IV. El lenguaje N_3 está incluido en el lenguaje $N_1 \cup N_2 \cup N_4$.
- b. Cada árbol de los lenguajes anteriores codifica (o representa) al natural correspondiente a la altura del mismo; por ejemplo, la cláusula base de N_1 codifica el cero.
- I. En caso que sea posible, construya una función $f : N_1 \rightarrow N_2$ siguiendo el ERP correspondiente que convierta las codificaciones de naturales en N_1 a sus correspondientes codificaciones en N_2 .
 - II. En caso que sea posible, construya una función $f : N_2 \rightarrow N_3$ siguiendo el ERP correspondiente que convierta las codificaciones de naturales en N_2 a sus correspondientes codificaciones en N_3 .
 - III. En caso que sea posible, construya una función $f : N_3 \rightarrow N_4$ siguiendo el ERP correspondiente que convierta las codificaciones de naturales en N_3 a sus correspondientes codificaciones en N_4 .
 - IV. En caso que sea posible, construya una función $f : N_3 \rightarrow N_2$ siguiendo el ERP correspondiente que convierta las codificaciones de naturales en N_3 a sus correspondientes codificaciones en N_2 .

Observe que no todos los lenguajes definidos son libres, por lo que debe tener cuidado al usar el ERP.