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

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

donde la función `Fibo` se define como:

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

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 $wa \in S$

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

Ejercicio 5 (Principio de inducción primitiva)

Sea un alfabeto $\Sigma = \{\bullet, \circ, \triangle, \square\}$ 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 $xwx \in PAL$

a. Proporcione 3 elementos de PAL.
b. Proporcione 3 elementos de Σ^* que no pertenezcan a PAL.
c. 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 $b\alpha bc \in \Delta$ III Si $\alpha \in \Delta$, entonces $b\alpha ba \in \Delta$

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

Para probar que un lenguaje no está incluído 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 = $\{\varepsilon, a, aa, aaa, aaaa, \ldots\}$

L2 = $\{\varepsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\}$

L3 = $\{b, aba, aabaa, aaabaaa, aaaabaaaa, \ldots\}$

L4 = $\{\varepsilon, ab, abab, ababab, abababab, \ldots, ba, baba, bababa, babababa, \ldots\}$

L5 = $\{\varepsilon, a, ab, aba, abab, ababa, ababab, \ldots\}$

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 $\Gamma = \Delta$.

I $\varepsilon \in \Gamma$	I $\varepsilon \in \Delta$
II Si $\alpha \in \Gamma$, entonces $a\alpha \in \Gamma$	II Si $\alpha \in \Delta$, entonces $\alpha a \in \Delta$
III Si $\alpha \in \Gamma$, entonces $b\alpha \in \Gamma$	III Si $\alpha \in \Delta$, entonces $\alpha b \in \Delta$

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.

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

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 $Q = T$

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 $\mathbb{N}$:

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

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

Ejercicio 12 (Funciones sobre tiras)

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

a. Defina las siguientes funciones.

$$\begin{aligned}
\texttt{largo} &: \Sigma^* \to \mathbb{N} \\
\texttt{duplicar} &: \Sigma^* \to \Sigma^* \\
\texttt{invertir} &: \Sigma^* \to \Sigma^*
\end{aligned}
\qquad
\begin{aligned}
\texttt{largo} &: \Sigma^+ \to \mathbb{N} \\
\texttt{último} &: \Sigma^+ \to \Sigma \\
\texttt{principio} &: \Sigma^+ \to \Sigma^* \\
\texttt{primero} &: \Sigma^+ \to \Sigma
\end{aligned}
\qquad
\begin{aligned}
\texttt{resto} &: \Sigma^+ \to \Sigma^* \\
\texttt{duplicar} &: \Sigma^+ \to \Sigma^+ \\
\texttt{invertir} &: \Sigma^+ \to \Sigma^+
\end{aligned}$$

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 sentido[1] tienen como dominio a Σ^+.

Los siguientes ejemplos aclaran el significado de las funciones pedidas.

$$\begin{aligned}
\texttt{largo}(abcdc) &= 5 \\
\texttt{resto}(abcdc) &= bcdc \\
\texttt{duplicar}(abc) &= aabbcc
\end{aligned}
\qquad
\begin{aligned}
\texttt{primero}(abcdc) &= a \\
\texttt{último}(abcdc) &= c
\end{aligned}
\qquad
\begin{aligned}
\texttt{principio}(abcdc) &= abcd \\
\texttt{invertir}(abcdc) &= cdcba
\end{aligned}$$

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

[1]En algunos casos, podemos totalizar la función: por ejemplo, podemos definir $\texttt{principio}(\varepsilon) := \varepsilon$. 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:

$$\begin{array}{l} \texttt{append} : \Sigma^* \times \Sigma^* \to \Sigma^* \\ \texttt{append}(\varepsilon, w) \quad = w \\ \texttt{append}(au, w) = a\texttt{append}(u, w) \end{array}$$

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

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

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 $\texttt{invertir}\colon \Sigma^* \to \Sigma^*$ (ej: $\texttt{invertir}(abc)\colon = cba$)
b. Demuestre usando inducción en Σ^* que $(\bar{\forall} w \in \Sigma^*)\ \texttt{invertir}(wx) = x\ \texttt{invertir}(w)$
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 $(\bar{\forall} w \in A)\ \texttt{invertir}(w) = w$

Ejercicio 15 (Más funciones sobre tiras)

a. Identifique la siguiente función.

$$\begin{array}{l} f : \Sigma^* \times \Sigma \to \{0, 1\} \\ f(\varepsilon, x) \quad = 0 \\ f(xw, y) = \text{si } x = y \text{ entonces } 1 \text{ sino } f(w, y) \end{array}$$

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

$$\begin{array}{r} \texttt{cant} : \Sigma^* \times \Sigma \to \mathbb{N} \\ \texttt{copiar} : \Sigma^* \times \mathbb{N} \to \Sigma^* \\ \texttt{sacar\_de\_la\_izquierda} : \Sigma^* \times \mathbb{N} \to \Sigma^* \\ \texttt{primera\_posición} : \Sigma^* \times \Sigma \to \mathbb{N} \end{array}$$

Los siguientes ejemplos aclaran el significado de las funciones pedidas.

$$\begin{array}{rr} \texttt{cant}(abcdc, c) = 2 & \texttt{copiar}(ab, 3) = ababab \\ \texttt{sacar\_de\_la\_izquierda}(abcdc, 3) = dc & \texttt{primera\_posición}(abcdc, c) = 3 \\ \texttt{sacar\_de\_la\_izquierda}(abcdc, 7) = \varepsilon & \texttt{primera\_posición}(abcdc, e) = 6 \end{array}$$

Ejercicio 16 (Orden)

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

a. Defina la función $\texttt{entre} : \Sigma^* \times \Sigma \times \Sigma \to \{0, 1\}$ 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, $\texttt{entre}(\text{jose}, b, t) = 1$ y $\texttt{entre}(\text{mama}, b, z) = 0$.

b. Defina la función `insord` $: \Sigma^* \times \Sigma \to \Sigma^*$ que inserta ordenadamente una nueva letra en una tira ya ordenada. Por ejemplo, $\texttt{insord}(abcde, d) = abcdde$. 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:

$$f(succ, 1) = 3, f(^2, 5) = 26$$

Consideremos una función $f : \Sigma \to \Sigma^*$. 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` $: \Sigma^* \times (\Sigma \to \Sigma^*) \to \Sigma^*$. Esa función debe cumplir que, si $\Sigma = \{a, b\}$ y f es tal que $f(a) = aba$ y $f(b) = bba$, entonces $\texttt{reemplazo}(abab, f) = ababbaababba$.

b. Encuentre una función f adecuada para poder probar la siguiente propiedad, con `duplicar` definida en Ejercicio 12:

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

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 $Reduce : (\mathbb{N} \times \mathbb{N} \to \mathbb{N}) \times \mathbb{N} \times \mathcal{L}_{\mathbb{N}} \to \mathbb{N}$. 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 (Numerales binarios)

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

I Si $x \in \Sigma$, entonces $x \in B_D$ II Si $w \in B_D$ y $x \in \Sigma$, entonces $xw \in B_D$	I Si $x \in \Sigma$, entonces $x \in B_I$ II Si $w \in B_I$ y $x \in \Sigma$, entonces $wx \in B_I$

a. Defina recursivamente la función `eval` : $B_D \to \mathbb{N}$ que devuelve el natural representado por un numeral binario. Es decir, `eval`$(10100) = 20$ y `eval`$(000100) = 4$.

b. Defina recursivamente la función `eval` : $B_I \to \mathbb{N}$ que devuelve el natural representado por un numeral 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, $(((1 \times 2 + 0) \times 2 + 1) \times 2 + 0) \times 2 + 0)$.

c. Defina recursivamente la función `par` : $B_I \to \{0, 1\}$ que devuelve 1 si la cantidad de unos en el numeral es par, y 0 si es impar.

Ejercicio 20

Considere los siguientes conjuntos:

- $P = \{p_i \ / \ i \ \in \ \mathbb{N}\}$
- $\Sigma = P \cup \{\oplus, \otimes,), (\}$
- 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:

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

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

Defina una función recursiva $eval_1 : \mathcal{L} \to \{0, 1\}$, 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 $eval_2 : \mathcal{L} \times (P \to \{0, 1\}) \to \{0, 1\}$, 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 la siguientes afirmaciones son verdaderas o falsas para cualquier función v:

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

Ejercicio 21 (Otros esquemas)

Considere las siguientes listas de ecuaciones[2]

$$
\begin{array}{ccc}
g(0) = 1 & h(0) = 1 & j(0) = 1 \\
g(1) = 0 & h(1) = 0 & j(2n+2) = 2 \times j(n) - 1 \\
g(n+2) = g(n) + \min\{g(n), g(n+1)\} & h(n+2) = h((n+2)\ mod\ 2) & j(2n+1) = 2 \times j(n) - 1
\end{array}
$$

a. Determine si son formas correctas de definir funciones. Justifique.

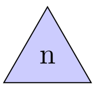
b. Pruebe que las primeras dos columnas definen una misma función probando que cada una de ellas es igual a la siguiente función:

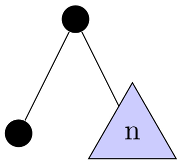
$$
\begin{array}{ll}
f : \mathbb{N} \to \{0,1\} & \\
f(2n) & = 1 \\
f(2n+1) & = 0
\end{array}
$$

Ejercicio 22 (Árboles binarios)

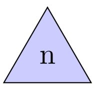
Considere las siguientes reglas de construcción de árboles binarios (sin información):

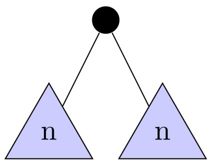
1. $\bullet \in \mathcal{L}$

2. Si 

$\in \mathcal{L}$, entonces 

$\in \mathcal{L}$

3. Si 

$\in \mathcal{L}$, entonces 

$\in \mathcal{L}$

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.

[2] Escribimos $n\ mod\ m$ 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 incluído en el lenguaje N_3.
  II. El lenguaje N_2 está incluído en el lenguaje N_4.
  III. El lenguaje N_3 está incluído en el lenguaje N_1.
  IV. El lenguaje N_3 está incluído 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 $f : N_1 \to N_2$ 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 $f : N_2 \to N_3$ 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 $f : N_3 \to N_4$ 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 $f : N_3 \to N_2$ 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.