Inducción Estructural

Semestre 2023-1

Desafío 14

Odin Miguel Escorza Soria **Tania Michelle Rubí Rojas**

Para cada uno de los siguientes ejercicios, **justifica ampliamente** tu respuesta:

1. Una cadena de caracteres w es palindroma si es de la forma $w = uu^R$, donde u^R es la cadena u escrita de atrás hacia delante.

   - **Define recursivamente** el conjunto de las cadenas palíndromas.
   - **Demuestra**, usando **inducción estructural**, que todas las cadenas palíndromas definidas anteriormente tiene un número par de símbolos.

2. La función `agrega` se define como sigue:

$$\texttt{agrega}(e, [a_1, \ldots, a_n]) = [a_1, \ldots, a_n, e]$$

   Por otro lado, la función `concat` se define como sigue:

$$\texttt{concat}([a_1, \ldots, a_n], [b_1, \ldots, b_n]) = [a_1, \ldots, a_n, b_1, \ldots, b_n]$$

   Realiza lo siguiente:

   - **Define recursivamente** las funciones `agrega` y `concat`.
   - **Demuestra**, usando **inducción estructural** y tus funciones recursivas, que

   [tcbox raise base]`agrega(e, concat(`l_1, l_2`)) = concat(`l_1`, agrega(e, `l_2`))`

3. El conjunto de paréntesis bien balanceados es el siguiente:

$$P = \{(), (()), ((())), (())(()), \ldots\}$$

   Realiza lo siguiente:

   - **Define recursivamente** el conjunto de paréntesis bien balanceados.
   - **Demuestra**, usando **inducción estructural** y tu función recursiva, que todos los elementos del conjunto P contienen un número igual de paréntesis izquierdos que derechos.

4. La función `cuenta` se define como sigue:

$$\texttt{cuenta}(s) = \text{ el número de unos que tiene } s$$

   donde s es una cadena binaria. **Realiza** lo siguiente:

   - **Define recursivamente** la función `cuenta`.
   - **Demuestra**, usando **inducción estructural** y tu función recursiva, que

   [tcbox raise base]`cuenta(st) = cuenta(s) + cuenta(t)`

5 La función `reversa` se define como sigue:

$$\texttt{reversa}([a_1, \ldots, a_n]) = [a_n, \ldots, a_1]$$

Por otro lado, la función `fusion` se define como sigue:

$$\texttt{fusion}([a_1, \ldots, a_n], [b_1, \ldots, b_n]) = [b_1, \ldots, b_n, a_1, \ldots, a_n]$$

- **Define recursivamente** las funciones `reversa` y `fusion`.
- **Demuestra**, usando **inducción estructural** y tus funciones recursivas, que:

[tcbox raise base]$\texttt{reversa(fusion(}l_1, l_2\texttt{)) = fusion(reversa(}l_1\texttt{), reversa(}l_2\texttt{))}$

6 La función `reversa` se define como sigue:

$$\texttt{reversa}([a_1, \ldots, a_n]) = [a_n, \ldots, a_1]$$

Por otro lado, la función `misterio` está definida recursivamente como sigue:

$$\text{misterio}([], l_2) = l_2$$
$$\text{misterio}((x : xs), l_2) = \text{misterio}(xs, (x : l_2))$$

- **Define recursivamente** la función `reversa`.
- ¿Qué hace la función `misterio`?
- **Ejecuta** la función `misterio` con dos listas no triviales.
- **Demuestra**, usando **inducción estructural** y las funciones recursivas anteriores, que:

[tcbox raise base]$\texttt{reversa(}l\texttt{) = misterio(}l\texttt{, []))}$

7 Sea A una fórmula de $\mathcal{LPROP}$ cuyos únicos conectivos lógicos son $\wedge, \vee, \neg$.

- **Define recursivamente** la función `swap`, la cual hace los siguientes intercambios en la fórmula A:

$\wedge$ por $\vee$ $\quad$ $\vee$ por $\wedge$ $\quad$ p por $\neg p$

donde p es una variable proposicional.

- **Demuestra**, usando **inducción estructural** y tu función recursiva, que

[tcbox raise base]$\neg A \equiv$ `swap(A)`

8 El conjunto de cadenas S está definido recursivamente como sigue:

i. $1 \in S$
ii. Si $s \in S$, entonces $0s, 1s \in S$.
iii. No hay nada en S que no sean objetos definidos con las reglas anteriores.

Utiliza **inducción estructural** para **demostrar** que cada cadena en S termina en 1.

9 El conjunto de cadenas T está definido recursivamente como sigue:

i. $a \in T$
ii. Si $t \in T$, entonces $ta, tb \in T$
iii. No hay nada en T que no sean objetos definidos con las reglas anteriores.

Utiliza **inducción estructural** para **demostrar** que cada cadena en T comienza con una a

10 El conjunto de cadenas S está definido recursivamente como sigue:

i. $\epsilon \in S$
ii. Si $s \in S$, entonces $bs, sb, saa, aas \in S$.
iii. No hay nada en S que no sean objetos definidos con las reglas anteriores.

Utiliza **inducción estructural** para **demostrar** que cada cadena en S tiene un número par de $a's$.

11 El conjunto de cadenas T está definido recursivamente como sigue:

i. $1, 2, 3, 4, 5, 6, 7, 8, 9 \in T$
ii. Si $t, u \in T$, entonces $t0, tu \in T$
iii. No hay nada en T que no sean objetos definidos con las reglas anteriores.

Utiliza **inducción estructural** para **demostrar** que ninguna cadena en T representa un entero con un cero principal.

12 El conjunto de cadenas S está definido recursivamente como sigue:

i. $1, 3, 5, 7, 9 \in S$
ii. Si $s, t \in S$, entonces $st, 2s, 4s, 6s, 8s \in S$.
iii. No hay nada en S que no sean objetos definidos con las reglas anteriores.

Utiliza **inducción estructural** para **demostrar** que cadena en S representa un entero impar.

13 El conjunto de cadenas T está definido recursivamente como sigue:

i. $\epsilon \in T$
ii. Si $x \in T$, entonces $1x0, 0x1 \in T$
iii. No hay nada en T que no sean objetos definidos con las reglas anteriores.

Utiliza **inducción estructural** para **demostrar** que cadena en T contiene un número igual de ceros y unos.

14 Un conjunto de árboles binarios está definido recursivamente como sigue:

i. Si $a \in A$, entonces `hoja(a)` es un árbol.
ii. Si t_1, t_2 son árboles, entonces `mk`(t_1, t_2) es un árbol.
iii. Éstos y sólo éstos son árboles.

Resuelve lo siguiente:

- **Define recursivamente** las funciones `nh` y `nni` que regresen el número de hojas y el número de nodos internos de este tipo de árboles, respectivamente.
- **Demuestra**, usando **inducción estructural** y las funciones recursivas anteriores, que:

[tcbox raise base]`nh(t) = nni(t) + 1`

15 Sea $\mathcal{T}$ un árbol binario. Utiliza **inducción estructural** para **demostrar** que si la altura de $\mathcal{T}$ es n, entonces el mínimo número de nodos en $\mathcal{T}$ es de n.

16 Sea $\mathcal{T}$ un árbol binario. Utiliza **inducción estructural** para **demostrar** que si la altura de $\mathcal{T}$ es n, entonces el máximo número de nodos internos es de $2^{n-1} - 1$.

17 ¿Cuál o cuáles son las similitudes y diferencias entre la inducción matemática y la inducción estructural?

18 ¿Es posible realizar una demostración por inducción dentro de otra demostración por inducción? En caso afirmativo, **muestra** un ejemplo. En caso contrario, **explica** por qué.

19 La función `sp` está definida como sigue:

$$\texttt{sp(k, } [a_1, \ldots, a_n]\texttt{)} = a_1^k + \ldots + a_n^k$$

Por otro lado, la función reversa está definida como sigue:

$$\texttt{reversa(}[a_1, \ldots, a_n]\texttt{)} = [a_n, \ldots, a_1]$$

- **Define recursivamente** las funciones `sp` y `reversa`.
- **Demuestra**, usando **inducción estructural** y las funciones recursivas anteriores, que:

[tcbox raise base]$\texttt{sp}(l)\ =\ \texttt{sp}(\texttt{reversa}(l))$

20 Sea A una fórmula en $\mathcal{LPROP}$.

- **Define recursivamente** las funciones `nc` y `np`, las cuales regresan el número de conectivos lógicos en A y el número de variables proposicionales en A, respectivamente.
- **Demuestra**, usando **inducción estructural** y las funciones recursivas anteriores, que:

[tcbox raise base]$\texttt{nc} \leq \texttt{np}$