Taller 2. Seminario de Matemáticas

1. El siguiente juego se juega con frijolitos. El juego comienza con los frijolitos agrupados en dos montones no necesariamente del mismo tamaño. Una movida del juego consiste en que uno de los jugadores elimina un montón y divide el montón que queda en dos montones (no necesariamente del mismo tamaño). Los jugadores se turnan en hacer sus movidas y gana el último en hacer una movida válida (sacar un montón y dividir el otro). Demuestre que, si uno de los montones iniciales tiene una cantidad par de frijolitos, entonces el jugador que tiene el primer turno puede ganar sin importar cómo juegue el adversario, es decir, existe para el jugador que inicia una estrategia ganadora. Use inducción por curso de valores sobre la cantidad de frijolitos que tiene el montón par inicial. Note como la demostración le define la estrategia ganadora.

2. Defina utilizando inducción estructural las cadenas de caracteres:

   a. Sea $\boldsymbol{\Sigma}$ un alfabeto. Usando inducción estructural defina el conjunto $\boldsymbol{\Sigma^*}$ de todas las ***cadenas*** sobre el alfabeto Σ.
   b. Defina una función que calcule el largo de una cadena (ℓ).
   c. Defina la función de concatenación que toma dos cadenas y devuelve la cadena resultante de colocar la segunda cadena en secuencia de la primera cadena.
   d. Demostrar que para todo par de cadenas w1,w2 que pertenecen a Σ* ℓ(w1w2) = ℓ (w1) + ℓ(w2).
   e. El inverso w−1 de una cadena w es la cadena que consiste de los símbolos de w en orden inverso. De una definición recursiva del inverso de una cadena.
   f. Demuestre usando inducción estructural que
   $(w1w2)-1 = w1^{-1}w2^{-1}$

3. La notación prefija permite definir expresiones aritméticas en la que el operador se escribe siempre antes que los operandos. Por ejemplo

   \+ 10 25 Expresa la suma el número 10 con el número 25

   \+ 4 * 2 6 Expresa la suma de 4 y el productos de 6 por 2.

a. Defina inductivamente el conjunto $\mathcal{P}_{\mathbb{N}}$ de todas las expresiones aritméticas en notación infija. Las expresiones contienen números naturales y los símbolos “+” y “*”.

b. Defina los operadores #op y #num para $\mathcal{P}_{\mathbb{N}}$ y demuestre que se cumple #op = #num – 1.

c. Usando esta definición explique paso a paso si estas expresiones están o no bien formadas:

   i. * * + 4 5 * 3 + 7 1 9

   ii. * + + 5 + + 6 7 8 * 9 1

d. En la notación prefija es posible evaluar expresiones sin la necesidad de usar paréntesis para establecer precedencia de operadores. Defina inductivamente la función eval: $\mathcal{P}_{\mathbb{N}} \rightarrow \mathbb{N}$ que evalúa una expresión aritmética prefija cualquiera. Por ejemplo:

eval(+ * 2 + 10 2 4) = 28

4. Usando Scheme implemente las funciones asociadas a las listas vistas en clase (las función head y sub ya están definidas en Scheme como cat y cdr respectivamente).
5. Usando Scheme programe la función InsertSort que ordena una lista de números naturales así:
   a. Defina la función ***insert*** que recibe una lista ordenada y un número natural y entrega una lista con el número natural insertado en la posición de que le corresponda de acuerdo al orden. Ejemplo:

      (insert ‘(1 4 5 10) 7) → (1 4 5 7 10)

   b. Use la función *insert* para definir la función *InsertSort* que ordena una lista desordenada. Ejemplo:

      (insertSort ‘(3 7 9 8 2 1)) → (1 2 3 7 8 9)