



Facilitador: Tomás J. Concepción Miranda

Indicaciones

Se debe realizar un informe de laboratorio, en el que se detalle, para cada problema, el desarrollo de la solución (no solo la respuesta). Este trabajo puede ser realizado de forma individual, en grupos de dos (2) o en grupos de tres (3). **Envíe su informe en Moodle, en formato PDF, así como el fuente del programa en el bloque correspondiente antes de las 11:55 p.m. del 15 de julio del 2023.**

Enunciados

Los siguientes problemas hacen uso del lenguaje Python. Los archivos `lab5.py` y `lab5_funciones.py` se encuentra junto a estas instrucciones en el eCampus. El archivo `lab5_funciones.py` contiene funciones se deben completar según el enunciado de los problemas. `lab5.py` sirve como un bosquejo para el desarrollo del laboratorio. Este script importa las funciones de `lab5_funciones.py`, es decir, que `lab5.py` puede usar las funciones escritas en `lab5_funciones.py`.

Tip: Reutilizen las funciones de laboratorios pasados de esta asignatura para usarlos en este laboratorio.

Problema 1: Complete la función `crear_op_bin_alea`. Esta función acepta como argumento un conjunto A , y retorna una operación binaria aleatoria sobre A . (7 puntos)

Problema 2: Complete la función `evaluar_op`. Esta función acepta como argumento una operación binaria f y una lista de operandos en un tupla¹, y retorna el resultado de evaluar la lista de operandos con la operación binaria de por medio, *e.g.* si la tupla es $(1, 2, 3)$, entonces se va evaluar como si fuera $1 * 2 * 3$. La operación es *asociativa a la izquierda*, por lo que, si hay varias operaciones seguidas y no hay parentesis, la evaluación se realiza de izquierda a derecha dos a dos, *e.g.* $a * b * c * d = ((a * b) * c) * d$. (7 puntos)

Problema 3: Complete la función `es_conmutativa`. Esta acepta como argumento un conjunto A y una operación binaria f sobre A , y retorna **True** si la operación binaria es conmutativa, de lo contrario retorna **False**. (8 puntos)

¹Recuerden que las tuplas puede ser de diferentes tamaños, por ejemplo 2, 3, 5 u 80

Problema 4: Complete la función `es_asociativa`. Esta acepta como argumento un conjunto A y una operación binaria f sobre A , y retorna una operación binaria asociativa aleatoria sobre A . (8 puntos)

Problema 5: Complete la función `crear_semigrupo_alea`. Esta acepta como argumento un conjunto A , y retorna `True` si la operación binaria es asociativa, de lo contrario retorna `False`. (8 puntos)

Problema 6: Complete la función `tiene_e_identidad`. Esta acepta como argumento un conjunto A , una operación binaria f sobre A y un argumento opcional `obtener_e_identidad` que por defecto es `False`, y retorna `True` si la operación binaria tiene un elemento de identidad, de lo contrario retorna `False`. Si `obtener_e_identidad=True`, adicionalmente devuelve el elemento de identidad, o devuelve `None` si la operación binaria no tiene elemento de identidad. (8 puntos)

Problema 7: Complete la función `crear_monoide_alea`. Esta acepta como argumento un conjunto A , y retorna un monoide sobre A . (8 puntos)

Problema 8: Complete la función `tiene_inversos`. Esta acepta como argumento un conjunto A , una operación binaria f sobre A y un argumento opcional `obtener_e_identidad` que por defecto es `False`, y retorna `True` si la operación binaria tiene un inverso para cada elemento de A , de lo contrario retorna `False`. (8 puntos)

Problema 9: Complete la función `crear_grupo_alea`. Esta acepta como argumento un conjunto A , y retorna un grupo aleatorio sobre A . Tip: establezca un elemento de identidad aleatorio y luego llene las operaciones en acorde a esto. (8 puntos)