



## Taller 2: Abstracción de datos y sintáxis abstracta

**Fundamentos de Lenguajes de Programación / 750095M / Grupo 01 / Prof. Robinson Duque / Monitor Juan Marcos Caicedo / 2019-1**

1. (20 pts) Implementar la interfaz *Bignum* para  $N = 16$  utilizando listas, como se muestra en la diapositiva número 12 de las transparencias del tema de Abstracción de datos. Recuerde que la representación *Bignum* está definida así (para un  $N$  grande):

$$\lceil n \rceil = \begin{cases} () & n = 0 \\ (\text{cons } r[\lceil q \rceil]) & n = qN + r, 0 \leq r < N \end{cases}$$

Donde, por ejemplo, si  $N = 16$ , algunas representaciones son:

- $\lceil 17 \rceil = '(1\ 1) = ((1 \times 16^0) + (1 \times 16^1))$
- $\lceil 4419 \rceil = '(3\ 4\ 1\ 1) = ((3 \times 16^0) + (4 \times 16^1) + (1 \times 16^2) + (1 \times 16^3))$
- $\lceil 3488135 \rceil = '(7\ 8\ 9\ 3\ 5\ 3) = ((7 \times 16^0) + (8 \times 16^1) + (9 \times 16^2) + (3 \times 16^3) + (5 \times 16^4) + (3 \times 16^5))$

**Importante:** En su implementación deben estar las entidades de la interfaz *zero*, *is-zero?*, *successor*, *predecessor* definidas de manera correcta para que la implementación funcione.

La implementación de la interfaz debe funcionar correctamente para el *código cliente* ya definido en clase:

- Suma
- Resta
- Multiplicación
- Potencia
- Factorial

Realice al menos 2 pruebas por cada operación para verificar su correcto funcionamiento.

2. (15pts) Teniendo en cuenta el ejemplo visto en clase de sumas anidadas (que se implementó usando listas y procedimientos), implementar las sumas anidadas usando **datatypes**, y para este caso, crear sus respectivas funciones *parse* y *unparse*.

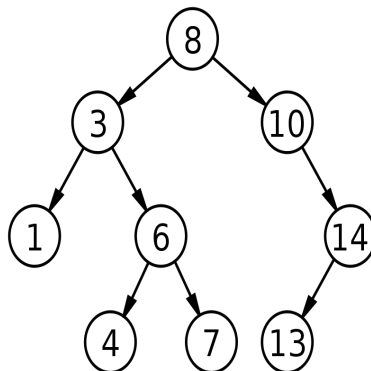
3. (30pts) Implementar el tipo de dato *stack* (pila), que consiste de las entidades:

- *empty-stack*: Definición de pila vacía.
- *push*: Insertar elemento en una pila.
- *pop*: Retira el elemento superior de la pila.
- *top*: Devuelve el elemento superior de la pila. (sin retirarlo)
- *empty-stack?*: Predicado que se encarga de preguntar si la pila está vacía.

El tipo de dato pila debe ser implementado y representado usando:

1. Listas
2. Procedimientos
3. Datatypes

4. (35pts) Un árbol binario de búsqueda es un árbol binario que cumple con la siguiente propiedad de orden para cada uno de sus nodos: el subárbol izquierdo de cualquier nodo (si no está vacío) sólo contiene valores menores que el del nodo, y el subárbol derecho (si no está vacío) sólo contiene valores mayores que el del nodo.



*Figura 1: Ejemplo de Árbol Binario de Búsqueda*

La siguiente es una gramática que representa un árbol binario y con la que se puede representar un árbol binario de búsqueda, tenga en cuenta que la gramática no es suficiente para definir o validar la propiedad de orden en un árbol binario para considerarlo como árbol binario de búsqueda. **Recuerde que una hoja puede representarse como un nodo con árbol-vacío en los hijos izquierdo y derecho.**

```

<árbol-binario> := (árbol-vacío) empty
                 := (nodo) <número> <árbol-binario> <árbol-binario>
  
```

Debe representar e implementar un **Árbol Binario de Búsqueda** en Racket usando:

1. Listas
2. Datatypes

Para cada una de las representaciones, debe realizar:

- Representación de un *árbol-vacío*

- Extractores del valor del nodo, del hijo izquierdo y del hijo derecho
- Predicados: *árbol-vacío?*, *árbol-hoja?*, *árbol-nodo?*
- Una función llamada *validador-orden* que reciba un árbol binario de búsqueda y valide su propiedad de orden, esto es, si en realidad se cumple la propiedad de orden de un árbol binario de búsqueda en el árbol binario dado. De ser así, la función debe retornar *#t*, de lo contrario, *#f*.
- Una función llamada *insertar-elemento* que recibe un árbol binario de búsqueda y un número *n*, si el elemento *n* ya existe en el árbol dado, retorna el árbol, sino, debe insertar el número *n* en el lugar correcto del árbol donde debe estar (teniendo en cuenta la propiedad de orden).

Adicionalmente para la representación de **Datatypes** debe agregar:

- Su respectiva función *unparse*.
- Su respectiva función *parse*.

## Aclaraciones

1. El taller es en grupos de máximo tres (3) alumnos.
2. La solución del taller debe ser subida al campus virtual a más tardar el día **23 de Junio a las 11:59 pm**. Se debe subir al campus virtual en el enlace correspondiente a este taller un archivo comprimido **.zip** que siga la convención *CódigoEstudiante1-CódigoEstudiante2-CódigoEstudiante3-Taller2FLP20191.zip*. Este archivo debe contener los archivos: **bignum16.rkt**, **sumasanidadas-datatypes.rkt**, **pilas-listas.rkt**, **pilas-procedimientos.rkt**, **pilas-datatypes.rkt**, **abb-listas.rkt**, **abb-procedimientos.rkt** y **abb-datatypes.rkt** que contengan el desarrollo de los ejercicios.
3. Para todos los tipos de datos se debe **incluir la gramática que utilizó**.
4. En las primeras líneas de cada archivo deben estar comentados los nombres y los códigos de los estudiantes participantes.
5. Se debe incluir para cada procedimiento un comentario que explique lo que realiza cada función y para qué es empleada.
6. En caso de tener dudas, puede consultar con el profesor **Robinson Duque** los días Jueves de 4 pm a 5 pm (después de clase, en su oficina) o los días Martes de 10 am a 11 am (en su oficina, enviando correo previo para ser atendidos) o consultar con el monitor **Juan Marcos Caicedo** en el horario de atención de los Miércoles de 2 pm a 3 pm (en el Laboratorio del Grupo de Investigación Avispa, 3er piso) y los Jueves de 4 pm a 5 pm (en el mismo lugar).

## Entregas Tardías o por Otros Medios

1. Este taller **sólo se recibirá a través del campus virtual**. Adicionalmente, sólo se evaluarán los documentos solicitados en el punto 2 de la sección anterior. Cualquier otro tipo de correo o nota aclaratoria será descartado. Sólo se aceptan envíos por fuera del horario establecido bajo excusas de fuerza mayor validadas a través de la dirección de la Escuela.

2. Las entregas tarde serán penalizadas así: (-1pt) por cada hora de retraso o fracción. Por ejemplo, si usted realiza su entrega y el campus registra las 12:00 am (i.e., 1seg después de la hora de entrega), usted está incurriendo en la primer hora de retraso. Asegurese con mínimo dos horas de anticipación que el link de carga funciona correctamente toda vez que es posible incurrir en una entrega tardía debido a los tiempos de respuesta.