



Taller 2: Abstracción de datos Parte 1

Get the power of the abstraction!

Fundamentos de Lenguajes Programación

Carlos Andres Delgado S
carlos.andres.delgado@correounivalle.edu.co

Marzo de 2023

1. Reglas

- El taller tiene como fecha límite de entrega el Lunes 17 de Abril de 2022 a las 23:59, hora de Colombia, a partir de allí se aplica una penalización que se acumula de 0.15 por hora o fracción de demora.
- Realizar en grupos de 2 a 4 personas, en caso de presentar individualmente o un grupo de 2 o 3 personas no permitan el ingreso de alguien que lo requiera, se sancionará con 0.5 en la nota del taller por incumplimiento de esta regla.
- Debe usarse recursión para resolver los problemas, no se permiten funciones como map o sort que resuelven el problema directamente
- Debe usarse sintaxis de Racket paréntesis en caso de entregar en papel, en computador debe ser funcional en lenguaje Racket.
- Las funciones deben definirse con **lambda**
- Entregue un sólo archivo comprimido. No entregue archivos comprimidos dentro de archivos comprimidos.
- Debe entregar el código fuente organizado en carpetas dentro del primer nivel del archivo comprimido, no cree una jerarquía compleja difícil de revisar.
- No se permite copiar código de Internet ni de sus compañeros. Si se encuentra código copiado el taller será anulado por completo a todas las partes involucradas.
- Debe implementar el taller en Dr Racket y recuerde que debe usar lambda para especificar sus funciones.
- Las primeras líneas de cada archivo deben contener los nombres y códigos de los estudiantes.

```
;Autores: Alvaro Uribe 1902321. Gustavo Petro, 1954545, Francia Marquez,  
↪ 1914547
```

1.1. Indicadores de logro

1. **IL 1.1.4** Implementa Tipos Abstractos de Datos utilizando dos componentes: una interfaz y una implementación recursivos
2. **IL 1.1.5** Utiliza estrategias con listas, procedimientos y datatypes para representar tipos de dato definidos por una gramática
3. **IL 1.1.6** Construye un árbol de sintaxis abstracta a partir de una representación concreta en una gramática BNF

2. Enunciado

Un registro es un tipo de dato, es una composición de datos simples y complejos bajo una estructura definida por llaves, por ejemplo un punto en un plano cartesiano de 2D se puede ver como

```
{x:10 , y:20}
```

En el cual la llave x tiene el valor 10 y la llave y tiene el valor 20, otro ejemplo podría ser la descripción de una persona

```
{nombre:"Maria", apellido:"Lopez", edad:23, hobbies:["Nadar","Cantar","Conquistar  
↪ el mundo", "Estudiar FLP", "Defender el cambio"]}
```

Como se puede observar también podemos tener datos complejos dentro de las llaves, como es el caso de hobbies, que tiene una lista de cadenas de caracteres.

En este taller vamos a implementar registros bajo las siguientes restricciones:

- La llave es un símbolo sin espacios
- Los datos que puede contener son número, símbolo, lista de números y lista de símbolos

Entregue un archivo por cada representación:

- Representación de los registros usando listas
- Representación de los registros usando procedimientos
- Representación de los registros usando datatypes

```
<record>
    ::= ' ( )
        empty-record
    ::= <symbol> <element> <record>
        non-empty-record(key, elm, record)

<element>
    ::= <simple-item>
        simple-item(datum)
    ::= <list-item>
        list-item(datum)
```

```

<simple-item>
  ::= <number>
    item-num(datum)
  ::= <symbol>
    item-sym(datum)

<list-item>
  ::= '()
    empty-list-item()
  ::= <simple-item> <list-item>
    non-empty-list-item(elm, lst)

```

Símbolo es cualquier **palabra** sin espacios y **numero** es cualquier número natural. La información que se encuentra en el recuadro son los nombres que deben llevar los constructores y las variantes en el datatype.

Así mismo, si se desea implementar un extractor su nombre está compuesto por el nombre de la variante y el nombre del campo, por ejemplo:

- non-empty-record->record
- item-num->datum

Los predicados usan el nombre del constructor, pero al final agregan un ?, por ejemplo empty-record?, non-empty-list-item?.

Le objetivo del taller es implementar los registros usando representación basada en listas, procedimientos y datatypes.

2.1. Funciones sobre el TAD

Diseñe las siguientes funciones para cada representación:

1. **get-numbers**: Retorna una lista con todos los números que se encuentran dentro de la estructura, esta lista debe ser plana
2. **get-symbols**: Retorna una lista con todos los símbolos que se encuentran dentro de la estructura, esta lista debe ser plana

2.2. Representación basada en listas

Debe implementar los constructores y observadores (predicados y extractores) respetando los nombres indicados en la gramática usando una representación basada en listas.

Si ha implementado correctamente estas funciones, para el caso de procedimientos solo necesita copiarlas y pegarlas.

2.3. Representación basada en procedimientos

Debe implementar los constructores y observadores (predicados y extractores) respetando los nombres indicados en la gramática usando una representación basada en procedimientos.

2.4. Representación usando Datatypes

Debe implementar los constructores y observadores (predicados y extractores) respetando los nombres indicados en la gramática usando define-datatype.

En este caso debe usar **cases** para el desarrollo de la función.

3. Rúbricas de evaluación

Las rubricas que se van a utilizar para la evaluación consideran aspectos relacionados con las reglas del taller, calidad del informe y de realización de las implementaciones. En cada una se explica la asignación de puntos.

Los ejemplos no deben ir comentados y deben estar en el código entregado para ser válidos. Recuerde respetar los nombres propuestos en la gramática e implementar cada una de las producciones indicadas en el enunciado.

Criterio	Nivel 0 (0 pts)	Nivel 1 (5 pts)	Nivel 2 (10 pts)	Nivel 3 (15 pts)
Diseño de representación con listas	No realiza las funciones o estas no se pueden ejecutar	Realiza los constructores y los predicados, sin embargo, hacen falta uno o más tipos de datos	Realiza las funciones correctamente, pero no incluye ejemplos de su funcionamiento o no usa los nombres indicados en la gramática	Realiza las funciones correctamente, respeta sus nombres de acuerdo al enunciado e incluye al menos 5 ejemplos de construcción del tipo de dato de este taller y uso de los observadores
Diseño de representación con procedimientos	No realiza las funciones o estas no se pueden ejecutar	Realiza los constructores y los predicados, sin embargo, hacen falta uno o más tipos de datos	Realiza las funciones correctamente, pero no incluye ejemplos de su funcionamiento o no usa los nombres indicados en la gramática	Realiza las funciones correctamente, respeta sus nombres de acuerdo al enunciado e incluye al menos 5 ejemplos de construcción del tipo de dato de este taller y uso de los observadores

Criterio	Nivel 0 (0 pts)	Nivel 1 (5 pts)	Nivel 2 (10 pts)	Nivel 3 (15 pts)
Diseño de funciones solicitadas	No realiza las funciones o entrega un código que no funciona	Realiza correctamente 1 función, o bien alguna de las funciones no trabaja correctamente. También aplica si el estudiante no usa los extractores y predicados diseñados para la función	Realiza correctamente las dos funciones, pero no incluye ejemplos de su funcionamiento	Realiza correctamente las funciones correctamente e incluye al menos 3 ejemplos de funcionamiento de cada una
Unicidad de las funciones solicitadas entre representación de listas y procedimientos	No realiza las funciones o entrega un código que no funciona o bien las funciones realizadas no son las mismas para ambas representaciones	Una función trabaja correctamente y es la misma para ambas representaciones	Dos funciones trabajan correctamente y son las mismas para ambas representaciones	Las dos funciones trabajan correctamente y son las mismas para ambas representaciones
Diseño de representación con datatypes	No realiza las funciones o estas no se pueden ejecutar	Implementa los datatypes, pero no están todos los tipos de datos necesarios o no se considera el tipo de cada uno de los campos de las variantes	Realiza los datatypes correctamente, pero no incluye ejemplos de su funcionamiento o no usa los nombres solicitados en la gramática	Realiza los datatypes correctamente, incluye 5 ejemplos de construcción de datos usando las funciones que proveen los datatypes y usa los nombres solicitados en la gramática
Diseño de funciones para el TAD con cases	No realiza las funciones o estas no se pueden ejecutar	Realiza la función usando los cases, pero la extracción de los datos no considera el tipo de cada uno de los campos	Realiza 1 función correctamente o no incluye ejemplos de cada una de las funciones	Realiza las 2 funciones correctamente e incluye al menos 3 ejemplos de funcionamiento de cada una

Total 90 puntos.

Su entrega será valorada a partir de las rúbricas consignadas previamente; cada una contiene una lista de criterios y niveles de desempeño. Para obtener la máxima calificación posible de su taller, debe cumplir las especificaciones del nivel más alto para cada criterio.

La nota se calculará con la suma de los puntos asignados al nivel que usted logre alcanzar para cada caso.

$$\text{Nota taller} = 5,0 * \frac{\text{puntos obtenidos}}{90}$$