



## Primer examen parcial - Parte 2 60 % parcial

### Fundamentos de lenguajes de programación

Duración: 2 horas

Carlos Andres Delgado S, Ing \*

17 de Octubre 2015

Nombre: \_\_\_\_\_  
Código: \_\_\_\_\_

**Importante:** Suba al campus virtual el interpretador modificado. Incluya comentarios en cada función que construya, en los comentarios deben ir casos de prueba. Utilice el interpretador definido en clase que consta de evaluación de expresiones condicionales, procedimientos y procedimientos recursivos. Descargue del campus virtual el interpretador **Interpretador procedimiento recursivos**

#### 1. Añadiendo primitivas [15 puntos]

Añada al lenguaje las primitivas de números  $\text{pow}(x,y)$ ,  $\text{div}(x,y)$ ,  $\text{mod}(x,y)$ . Estas primitivas realizan las operaciones: *eleva  $x$  a la  $y$ , dividir  $x$  entre  $y$  y  $y$  modulo  $y$  de  $x$ .*

- (7 puntos) Describa los cambios que deben realizarse a la especificación de la gramática y al tipo de dato primitivo
- (8 puntos) Describa los cambios que deben realizarse al procedimiento `applyprimitive`

#### 2. Añadiendo operaciones al lenguaje [40 puntos]

Su misión si decide aceptarla es agregar dos operaciones que se le solicitan. Para cada caso debe explicar los cambios que realizó en la gramática y funciones del lenguaje.

- (25 puntos) Agregue la operación *list* al lenguaje, esta toma cualquier número de argumentos y retorna un valor expresado que es la lista de valores, por ejemplo:

```
1 let x = 4
2 in list(x, -(x,1), -(x,3))
```

Debe retornar: '(4 3 1)

- (15 puntos) Añada al lenguaje la operación *print*, esta toma un argumento, lo imprime y retorna 1.

#### 3. Añadiendo registros al interpretador [45 puntos]

Hasta el momento los valores expresados del lenguaje se componen de números, booleanos, procedimientos. En este ejercicio se le propone extender el conjunto de valores expresados del lenguaje para manejar registros. De esta manera se utilizará el tipo de dato *registro* definido previamente. Para esto es necesario añadir al interpretador del curso algunas variaciones:

- (12 puntos) Es necesario introducir la expresión de creación de registros: Un registro se debe crear indicando una etiqueta y un conjunto de campos seguidos de valores. Por ejemplo:

```
registro(campo1:3 campo2:4 campo3:10) y
unregistrovacio().
```

Así mismo se debe incorporar la expresión de acceso a los campos de un registro. Para esto se utilizara la notación `.Rec.campo` donde `Rec` debe ser una expresión que tenga como valor un registro y `campo` debe ser un símbolo.

Indique la BNF para registros y los cambios que deben realizarse en la gramática del interpretador.

- (20 puntos) Es necesario definir las acciones a realizar por el evaluador de expresiones cuando encuentra las variantes para creación de registros y para acceso a los campos de un registro. Realice esta modificación, asegurando que sólo se pueden acceder a campos de valores tipo *registro*. Intentar acceder a un campo de un dato cuyo valor no sea de este tipo debe producir un error.
- (8 puntos) Por último, a veces es necesario verificar si dos registros dados son iguales. Implemente la primitiva `eqv-records?` que recibe dos registros y determina si son iguales. Describa los cambios que debe realizar en la función `apply-primitive`.

\*carlos.andres.delgado@correounivalle.edu.co