

CPLP - 1er parcial - Parciales viejos

05/05/2014.....	2
1).....	2
2).....	2
3).....	2
4).....	3
01/07/2022 - Tema 2.....	4
Ejercicio 1.....	4
Ejercicio 2.....	5
Ejercicio 3.....	5
28/4/2023.....	6
Ejercicio 1.....	6
Ejercicio 2.....	6
Ejercicio 4.....	7

05/05/2014

1)

05_05_2014_Ejercicio_1

2)

Identificador	Nro. línea	Momento alocación (l-valor)	R-valor	Alcance	Tiempo vida
parcial	1	-	-	2-26	1-26
p	4	automático	indef. (basura)	5-26	1-26
q	4	automático	indef. (basura)	5-26	1-26
pint	5	automático	nil	6-26	1-26
pint^	5	dinámico	indef. (basura)	6-26	23-26
auxiliar	6	-	-	7-26	6-13
aux	7	automático	indef. (basura)	8-13	6-13
i	7	automático	indef. (basura)	8-13	6-13
v	7	semi-dinámico	indef. (basura)	8-13	6-13
valor	14	-	-	15-26	14-19
h	15	automático	indef. (basura)	16-19	14-19

3)

$G = (N, T, S, P)$

$N = \{ \langle \text{funcion} \rangle, \langle \text{bloque_indentado} \rangle, \langle \text{sentencia} \rangle, \langle \text{letra} \rangle, \langle \text{digito} \rangle, \langle \text{identificador} \rangle, \langle \text{parametros} \rangle, \langle \text{indentacion} \rangle, \}$

T = {0, ..., 9, 'a', ..., 'z', 'A', ..., 'Z', '(', ')', ',', ' ', '_', ':'}

S = {<funcion>}

P = {

<funcion> ::= 'def ' <identificador> '(' [<parametros>] ':' NUEVA_LINEA

<bloque_indentado>

<bloque_indentado> ::= {<indentacion> <sentencia>}+

<sentencia> ::= (<sentencia_if> | <sentencia_while> | <sentencia_for> |

<sentencia_try_except> | <asignacion> | <funcion> | <llamada_a_funcion> | ...)

<indentacion> ::= ' '

<parametros> ::= <parametro> {', ' <parametro>}*

<parametro> ::= <identificador> []

<identificador> ::= ('_' | <letra>){('_' | <letra> | <digito>)}

<digito> ::= (0 | ... | 9)

<letra> ::= ('a' | ... | 'z' | 'A' | ... | 'Z')

}

4)

- a) Qué verga es un lenguaje expresivo 🤔, pero me huele a falso.
- b) Falso, es estática en cuanto a su l-valor si se aloca en memoria en compilación y se desaloca luego de que termina el programa (no cuando finaliza la ejecución sino después).
- c) Falso.
- d) Verdadero(? si con puesta a punto se hace referencia a que en la interpretación se ejecuta de una y en la compilación hay que esperar a que compile xd.

01/07/2022 - Tema 2

Ejercicio 1

a) $G = (N, T, S, P)$

$N = \{ \langle \text{try_except} \rangle, \langle \text{bloque_indentado} \rangle, \langle \text{bloque_except} \rangle, \langle \text{bloque_else} \rangle, \langle \text{bloque_finally} \rangle, \langle \text{sentencia} \rangle, \langle \text{identificador} \rangle, \langle \text{indentacion} \rangle, \langle \text{digito} \rangle, \langle \text{letra} \rangle \}$

$T = \{ 'a', \dots, 'z', 'A', \dots, 'Z', 0, \dots, 9, ':', '_', ' ' \}$

$S = \{ \langle \text{try_except} \rangle \}$

$P = \{$

$\langle \text{try_except} \rangle ::= \text{'try:' NUEVA_LINEA}$

$\langle \text{bloque_indentado} \rangle$

$\{ \langle \text{bloque_except} \rangle \}^+$

$[\langle \text{bloque_else} \rangle]$

$[\langle \text{bloque_finally} \rangle]$

$\langle \text{bloque_except} \rangle ::= \text{'except '}[\langle \text{identificador} \rangle] \text{' NUEVA_LINEA}$

$\langle \text{bloque_indentado} \rangle$

$\langle \text{bloque_else} \rangle ::= \text{'else:' NUEVA_LINEA}$

$\langle \text{bloque_indentado} \rangle$

$\langle \text{bloque_finally} \rangle ::= \text{'finally:' NUEVA_LINEA}$

$\langle \text{bloque_indentado} \rangle$

$\langle \text{bloque_indentado} \rangle ::= \{ \langle \text{indentacion} \rangle \langle \text{sentencia} \rangle \}^+$

$\langle \text{indentacion} \rangle ::= ' '$

$\langle \text{sentencia} \rangle ::= (\langle \text{sentencia_if} \rangle \mid \langle \text{sentencia_while} \rangle \mid \langle \text{sentencia_for} \rangle \mid$

$\langle \text{asignacion} \rangle \mid \langle \text{try_except} \rangle \mid \langle \text{funcion} \rangle \mid \langle \text{llamada_a_funcion} \rangle \mid \dots)$

$\langle \text{identificador} \rangle ::= ('_' \mid \langle \text{letra} \rangle) (('_' \mid \langle \text{letra} \rangle \mid \langle \text{digito} \rangle))^*$

$\langle \text{letra} \rangle ::= ('a' \mid \dots \mid 'z' \mid 'A' \mid \dots \mid 'Z')$

$\langle \text{digito} \rangle ::= (0 \mid \dots \mid 9)$

$\}$

b) [Link al GitHub](#)

Ejercicio 2

Nombre	Momento alocación	R-valor	Alcance	Tiempo vida
y	automático	0	2-11 16-23	1-23
s	automático	null	3-11	1-11
s*	dinámico	basura	3-11	11
main	-	-	5-11	4-11
var3	estático	0	6-11	<1-23>
m	automático	basura	8-11	4-11
n	automático	basura	8-11	4-11
var2	automático	basura	10	9-10
auxi	estático	0	13-23	<1-23>
b	automático	0	7-11 14-20	1-23
funcion2	-	-	15-23	14-18
funcion3	-	-	20-23	19-23
b	automático	basura	21-23	19-23

Ejercicio 3

- a) Falso, eso es la sobrecarga, pero que verga tiene que ver con CPLP? 🤔
- b) Que verga es la unión discriminada. Pero me la juego por verdadero

28/4/2023

Ejercicio 1

a) $G = (N, T, S, P)$

N =

$\{ \langle url \rangle \langle protocolo \rangle \langle dominio \rangle \langle ruta \rangle \langle caracter_valido_ruta \rangle \langle caracter_valido_dominio \rangle \langle letra \rangle \}$

$T = \{ 'a', \dots, 'z', 'A', \dots, 'Z', 0, \dots, 9, '-', '.', '/', ':', '&', '?', '=' \}$

$S = \{ \langle url \rangle \}$

$P = \{$

$\langle url \rangle ::= \langle protocolo \rangle ':' \langle dominio \rangle [\langle ruta \rangle]$

$\langle protocolo \rangle ::= 'http' ['s']$

$\langle dominio \rangle$ $::=$

$['www.'] \{ \{ \langle caracter_valido_dominio \rangle \}^+ '.' \}^* \{ \langle caracter_valido_dominio \rangle \}^+$

$\langle ruta \rangle ::= \{ '/' \} \{ \langle caracter_valido_ruta \rangle \}^+$

$\langle caracter_valido_ruta \rangle ::= (\langle letra \rangle \mid '-' \mid \langle digito \rangle \mid \& \mid ? \mid = \mid .)$

$\langle caracter_valido_dominio \rangle ::= (\langle letra \rangle \mid '-' \mid \langle digito \rangle)$

$\langle letra \rangle ::= ('a' \mid \dots \mid 'z' \mid 'A' \mid \dots \mid 'Z')$

$\langle digito \rangle ::= (0 \mid \dots \mid 9)$

$\}$

b) Link al GitHub

Ejercicio 2

Nombre	Momento alocación	R-valor	Alcance	Tiempo vida
main	-	-	3-12	2-12
a	automático	basura	5-12	2-12
n	automático	basura	5-12	2-12

p	automático	basura	5-7.3 8-12	2-12
vec1	automático	basura	6-12	2-12
c	automático	basura	7-12	2-12
Uno	-	-	7.1-12	7-7.5
vec2	semidinámico	basura	7.3-7.5	7-7.5
p	automático	nil	7.4-7.5	7-7.5
p^	dinámico	basura	7.4-7.5	7.4.4

Ejercicio 4

- a) Es importante contar con criterios de evaluación para los lenguajes, ya que estos nos van a permitir conocer las ventajas y desventajas en diferentes puntos de la implementación del lenguaje, como simplicidad, ortogonalidad, abstracción o eficiencia de tiempo y espacio.

Dos criterios relevantes para un lenguaje son:

- Ortogonalidad, es decir, la capacidad de, a partir de un pequeño conjunto de instrucciones básicas, poder construir sentencias complejas y consistentes. Este criterio es relevante porque de no cumplirse puede llevar a inconsistencias que dificulten al desarrollador (un ejemplo de esto es C, que no puede retornar arrays o funciones, de una función, por lo que hay que buscar otras formas de hacerlo como retornar un puntero a un array o función)
- Abstracción: es la cualidad de un lenguaje que permite al programador definir y usar estructuras u operaciones complejas (lo copié de la teoría je), sin tener que darle importancia a la forma en que las mismas están implementadas, ya que esos detalles le son invisibles al programador. Es relevante en un lenguaje puesto que permite desarrollar aplicaciones muy complejas pero entendibles, al hacer uso de funciones que, si bien son complejas, se las accede de una manera simple (ya que el lenguaje permite abstraernos de su implementación)

- b) Tipo:

- Constante: la estabilidad de la ligadura de tipo de una constante es estática
- Variable estática: la estabilidad de la ligadura de tipo de una variable estática puede ser dinámica (si el lenguaje lo permite, es decir, si es dinámicamente tipado)

Alcance:

- Constante y variable estática: desde la siguiente línea donde se declaran

Momento de asignación (del l-valor):

- Constante: su momento de asignación es automático: se asigna memoria al asignarse la memoria de la unidad que la contiene, y se desaloca cuando el programa finaliza su ejecución
- Variable estática: su momento de asignación es estático: se asigna memoria en tiempo de compilación y se mantiene su valor para las próximas ejecuciones

R-valor:

- Constante: la estabilidad de la ligadura de una constante con su r-valor es estática, el mismo no puede variar en ejecución
- Variable estática: la estabilidad de la ligadura de una variable estática con su r-valor es dinámica, el mismo puede variar en ejecución

Ambos conceptos tienen algunas similitudes, pero también diferencias bastante importantes como lo son el momento de asignación (automático vs estático) y la estabilidad de la ligadura, tanto de tipo como de r-valor (estática vs dinámica (en el caso del tipo, es dinámica si el lenguaje es dinámicamente tipado))

c) La semántica estática analiza me duermo y me tengo q ir a comer :)