

UTN FRBA – SSL – Examen Final – 2025-05-20

Apellido, Nombre:		Legajo:		Nota:	
-------------------	--	---------	--	-------	--



- Resuelva el examen en tinta y en esta hoja; no se aceptan documentos adicionales.
- Durante el examen no se responden consultas; si lo necesita, escriba hipótesis de trabajo, las cuales también se evalúan.

1. (1 punto) Indique la única afirmación falsa sobre la utilización de BNF para el desarrollo de los LP:

- ☐ Especifica la precedencia operadores
- ☐ Especifica la asociatividad de los operadores.
- ☐ Especifica los elementos terminales del lenguaje.
- ☐ Especifica el orden de evaluación de los operandos.
- ☐ Puede ser implementada mediante la herramienta bison (yacc)

2. (1 punto) Indique la única afirmación falsa sobre GR:

- ☐ Sus producciones pueden tener a derecha ϵ .
- ☐ El lenguaje generado puede ser descrito por una regex.
- ☐ Todos los lenguajes finitos pueden ser generados por GR.
- ☐ Sus producciones pueden tener a derecha dos elementos no terminales.
- ☐ El lenguaje generado es cerrado respecto de la operación concatenación consigo mismo.

3. (1 punto) ¿Por qué no se puede desreferenciar un puntero nulo?

4. Dado:

```
int a[7]={42,21,14}, *p=a;
int *buscar(int buscado, const int *arreglo_donde_buscar, size_t
    tamaño_arreglo); //retorna puntero a encontrado o nullptr
```

a. (1 punto) Indique y justifique la pragmática (porqué se usa) de const:

b. Indique y justifique la semántica las siguientes expresiones sintácticamente correctas:

i. (1 punto) Asignación

- `p = a`
- `a = p`

ii. (1 punto) sizeof

- `sizeof a`
- `sizeof p`

iii. (2 puntos) Subindicación

- `*(p+2)`
- `a[7]`

iv. (2 puntos) Invocación

- `buscar(7,a, 7)`
- `p - buscar(14,a, 7)`

v. (Punto Extra) 2[a]

1. Una Resolución

1. Especifica el orden de evaluación de los operandos. Porque es un concepto semántico que describe comportamiento y no estructura como lo hace BNF.
2. Sus producciones pueden tener a derecha dos elementos no terminales. Por definición de GR.
3. Porque `nullptr` no apunta a ningún objeto ni función, por lo tanto desreferenciarlo da un comportamiento indefinido.
4.
 - a. Porque se quiere denotar que la función no modifica los valores de el arreglo_donde_buscar.
 - b.
 - i.
 - Es correcto porque en una expresión el identificador de un arreglo se degrada a la dirección de memoria del primer elemento del arreglo.
 - Es un error semántico porque `a` no es un *l-value* modificable.
 - ii.
 - Es el tamaño en bytes ocupado por la variable arreglo.
 - Es el tamaño en bytes de la variable puntero.
 - iii.
 - `*(p+2)` accede al tercer elemento, equivale a `a[2]`.
 - Es comportamiento indefinido porque `a[7]` referencia a la posición de memoria que le sigue al arreglo; al desplazarse la cantidad total de elementos cae afuera del arreglo.
 - iv.
 - `buscar(7, a, 7)` retorna `nullptr` porque no lo va a encontrar, los cuatro últimos valores del arreglo se inicializan implícitamente en cero.
 - `p - buscar(14, a, 7)` usa aritmética de punteros (i.e., escala en función del tamaños de los objetos apuntos) y es la diferencia o distancia de punteros; como 14 está en el elemento con índice 2, por diferencia de punteros 2 es el resultado de la expresión.
 - v. `a[i] == *(a+1)`, y la suma es conmutativa.

v1.1.0-beta.1+2025-05-21