

UTN FRBA – SSL – Examen Final – 2025-02-17

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. Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique su respuesta:

- (1 punto) El *scanner* (*analizador léxico*) recibe un flujo de tokens.
- (1 punto) Para toda GR existe un autómata de pila que reconoce el LF generado por esa GR.
- (1 punto) El alcance de un identificador depende de donde se lo declara.
- (1 punto) La sentencia `for(1;2;3)4;` es correcta sintáctica y semánticamente.

2. Dada la siguiente expresión sintácticamente correcta en C:

`*a**b*c`

- (2 puntos) Durante el análisis léxico, indique cuántos operadores de multiplicación y cuántos de desreferencia se detectan. Justifique:
- (1 punto) Indique en qué etapa de la compilación se genera el árbol de derivación. Justifique:
- (2 puntos) Dibuje el árbol de expresión.
- (1 punto) Escriba una declaración que haga semánticamente correcta a la expresión.

3. (Punto Extra) Los programas en C pueden recibir argumentos desde la línea de comando cuando la definición de *main* tiene esta forma:

```
int main(int argc, char **argv){/*...*/}
```

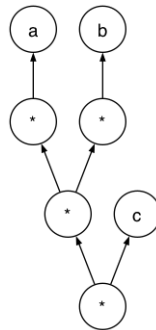
Suponga que, como ejemplo, en la línea de comando se ingresa el siguiente texto:

```
cc -c hello.c -std=c23
```

Dibuje una representación visual (o diagrama de memoria, o diagrama de objetos) de la estructura que generan `argc` y `argv` que contenga como valores los del ejemplo dado.

1. Una Resolución

1.
 - a. F. Genera un flujo de tokens.
 - b. V. Jerarquía de Chomsky.
 - c. V. Por ejemplo: externa, interna, miembro de estructura.
 - d. V. Puede derivarse según la gramática de C y respeta las restricciones semánticas; pero del punto de vista pragmático es un flujo infinito sin efecto visible.
2.
 - a. (2 puntos) Ninguno. No tiene la información sintáctica para diferenciarlo. Son todas apariciones del token *asterisco*.
 - b. (1 punto) Análisis sintáctico o Parsing. Se aplican las reglas gramaticales para derivar el árbol sintáctico.
 - c. (2 puntos) $((*a) * (*b)) * c$



- d. (1 punto) `int *a, *b, c;`
3.
 - Una solución en C:


```
argc = 4;
char *a[]={ "cc", "-c", "hello.c", "-std=c23", nullptr };
argv = a;
```
 - Y otra solución con un diagrama:

