

## UTN FRBA – SSL – Examen Final – 2019-05-23

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



- Resuelva el examen en tinta y en esta hoja; no se aceptan hojas adicionales.
- Para los ítems de *una mejor respuesta*, marcados con un círculo (○), tilde (✓) sólo una opción, la mejor.
- Para los ítems de *respuestas múltiple*, marcados con un caja (□), tilde (✓) todas las respuestas correctas.
- Durante el examen no se responde consultas; si lo necesita, escriba hipótesis de trabajo, las cuales también se evalúan.

1. (2 puntos) Defina formalmente el AF que resulta de la intersección entre las constantes octales sin sufijo de C, las expresiones de C y el LF "números sin signo expresados en base dos".
2. (2 puntos) Tilde todas las afirmaciones **verdaderas** acerca de los algoritmos de MUCH2012 volumen 3:
  - ☐ El complemento de un LR siempre es un LR.
  - ☐ Siempre se puede encontrar una ER para un GR.
  - ☐ Siempre se puede encontrar una GIC para un ER.
  - ☐ Siempre se puede encontrar un AFD para una ER.
  - ☐ El algoritmo de Thompson siempre produce un AFD.
3. (2 puntos) Declare un arreglo y luego dé un ejemplo **semánticamente correcto** de *valor-l no modificable* que use ese arreglo.
4. (2 puntos) Dada la declaración `struct{double x,y;}p`; escriba una expresión **semánticamente incorrecta** utilizando como operandos solo constantes (literales) y los identificadores declarados en la anterior declaración.
5. Dado el siguiente fragmento de código C: `x.1==1LU++LU`
  - a. (2 puntos) Enumere los lexemas en orden de aparición y separados por coma.
  - b. (Punto extra) Justifique si el fragmento es una expresión sintácticamente correcta o no.

## 1. Una Resolución

1.  $M = (\{A, B\}, \{0, 1\}, \{A \Rightarrow 0 \Rightarrow B, B \Rightarrow 0 \Rightarrow B, B \Rightarrow 1 \Rightarrow B\}, 0, \{B\})$

2.   ✓

      ✓

      ✓

      ✓

      □

3.   • Declaración: `int a[7];`

      • Ejemplo: `a`

4. `0 = p.x + p.y`

5.   a. `x, ., 1, ==, 1LU, ++, LU`

      b. Es sintácticamente incorrecta porque no es posible derivar la expresión siguiendo las reglas sintácticas del lenguaje (BNF); más precisamente no se espera una constante luego del operador sufijo de acceso `.`, tampoco se espera una expresión luego del operador sufijo de incremento `++` sufijo. Queda como ejercicio armar el árbol de derivación hasta el punto donde no se puede derivar más.

v1.0.0, 2019-05-25