

## UTN FRBA – SSL – Examen Final – 2019-12-11

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 una 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) Tilde todas las afirmaciones **verdaderas** acerca del análisis léxico:

- ☐ Puede implementarse con un AF.
- ☐ Procesa su entrada carácter a carácter.
- ☐ `s printf` es siempre léxicamente correcto.
- ☐ Un lexema dado puede generar diferentes lexemas.
- ☐ Diferencia entre identificadores y palabras clave (keywords).

2. (2 puntos) Tilde todas las afirmaciones **verdaderas** acerca del análisis sintáctico:

- ☐ Un parser procesa expresiones.
- ☐ Un PAS puede invocar al scanner.
- ☐ `s printf` es siempre sintácticamente correcto.
- ☐ El operador de acceso a miembro (.) es conmutativo.
- ☐ En un árbol de derivación, los nodos internos son variables.

3. (2 puntos) Tilde todas las afirmaciones **verdaderas** acerca del análisis semántico:

- ☐ Tiene en cuenta el contexto.
- ☐ Hay un error semántico en `while++`.
- ☐ Detecta errores de programación (bugs).
- ☐ Usa los prototipos para validar las invocaciones.
- ☐ `s printf` es siempre semánticamente correcto.

4. Dada la siguiente gramática:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow 1 \mid 2 \mid 3$

Indique si las siguientes afirmaciones son verdaderas o falsas y justifique su respuesta

- (2 puntos) Puede ser implementada directamente, es decir, sin aplicar ninguna transformación a la gramática, por un parser descendente recursivo.
- (2 puntos) Puede ser implementada directamente, es decir, sin aplicar ninguna transformación a la gramática, por Yacc/Bison.

## 1. Una Resolución

1. ☒  
☒  
☒  
☐  
☒
2. ☐  
☒  
☒  
☐  
☒
3. ☒  
☐  
☐  
☒  
☐
4. a. Falsa. El PDR no soporta recursión a izquierda, debe ser transformada primero una GIC LL(1)  
b. Verdadera. Yacc/Bison admiten recursión a izquierda.

v1.1.0, 2019-12-15