

## UTN FRBA – SSL – Examen Final – 2019-12-04

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:

- ☐ Recibe lexemas y genera tokens.
- ☐ `ungetc` devuelve al flujo el último token leído.
- ☐ Un token puede ser generado por diferentes lexemas.
- ☐ En C, `printf("ho1a")`, es siempre léxicamente correcto.
- ☐ Se puede determinar el operador que representa el lexema asterisco (\*).

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

- ☐ Yacc es un parser.
- ☐ Un parser procesa una secuencia de tokens.
- ☐ En un árbol de derivación, las hojas son tokens.
- ☐ En C, la expresión `a+b+c` tiene un único árbol de derivación.
- ☐ En C, `printf("ho1a")` es siempre sintácticamente correcto.

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

- ☐ Es parte del compilador.
- ☐ Detecta un error en `printf("%d")`.
- ☐ Vincula con las funciones de biblioteca.
- ☐ Detecta inconsistencias de tipo de dato.
- ☐ En C, `printf("ho1a")` es siempre semánticamente correcto.

4. Dado

$$L = \{a^n b^n \mid n > 0\}$$

- (2 puntos) Escriba un BNF que lo genere.
- (2 puntos) Codifique en C el o los PAS asociados:

5. (Punto Extra) Dibuje el árbol de expresión para `f(x).a[i]`

## 1. Una Resolución

1. ☐  
☐  
☒  
☒  
☐
2. ☐  
☒  
☒  
☒  
☒
3. ☒  
☐  
☐  
☒  
☐
4. a.  $S \rightarrow aTb, T \rightarrow aTb | \epsilon$   
ó  
 $S \rightarrow aSb, S \rightarrow ab$   
b.  

```
void S(void){
    Match(A);
    if(A==GetNextToken())
        S();
    Match(B);
}
```
5.  $(( ( ( f ) ( x ) ) ) . a ) [ ( i ) ] )$

