

## UTN FRBA – SSL – Examen Final – 2018-10-03

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. (1 punto) Tilde todas las afirmaciones **verdaderas** con respecto a los algoritmos de MUCH2012 volumen 3:

- ☐ Siempre se puede encontrar un AF para una ER.
- ☐ Siempre se puede encontrar una ER para un AFD.
- ☐ Siempre se puede encontrar un AFD para un AFN.
- ☐ Siempre se puede encontrar una ER para una GIC.
- ☐ Siempre se puede encontrar el AFD mínimo dado un AF.

2. (1 punto) Tilde todas las afirmaciones **verdaderas** con respecto a las GIC:

- ☐ Pueden generar lenguajes finitos.
- ☐ Sus producciones pueden tener  $\epsilon$  en su derecha.
- ☐ Generan LF reconocibles autómatas con pila.
- ☐ Sus producciones pueden tener terminales en su izquierda.
- ☐ Generan LF procesables por parsers con análisis sintáctico descendente recursivo.

3. Analice la siguiente función: `int f(void){return 1<*p++;}`

a. Nivel léxico:

- i. (1 punto) Indique cuantos tokens hay:
- ii. (1 punto) Indique los tres tokens que pertenecen a un LF infinito:
- iii. (1 punto) Indique por lo menos tres tokens que **no** requieran `ungetc` para su detección:

b. Nivel sintáctico, dentro de la sentencia compuesta:

- i. (1 punto) Indique cuantas declaraciones hay:
- ii. (1 punto) Indique el operador con menor precedencia:
- iii. (1 punto) Indique cuántas expresiones, contando las subexpresiones, hay; por ejemplo `40+2` tiene tres expresiones:

c. Nivel semántico:

- i. (1 punto) Agregue antes de la función **una** declaración que haga semánticamente **correcta** a la función:
- ii. (1 punto) Agregue antes de la función **una** declaración que haga semánticamente **incorrecta** a la función:
- iii. (Punto extra) Justifique su anterior respuesta:

## 1. Una Resolución

1.

- ✓ Siempre se puede encontrar un AF para una ER.
- ✓ Siempre se puede encontrar una ER para un AFD.
- ✓ Siempre se puede encontrar un AFD para un AFN.
- ☐ Siempre se puede encontrar una ER para una GIC.
- ✓ Siempre se puede encontrar el AFD mínimo dado un AF.

2.

- ✓ Pueden generar lenguajes finitos.
- ✓ Sus producciones pueden tener a derecha  $\epsilon$ .
- ✓ Generan LF reconocibles autómatas con pila.
- ☐ Sus producciones pueden tener terminales a izquierda.
- ☐ Generan LF procesables por parsers con análisis sintáctico descendente recursivo.

3.

a.

- i. `int f ( void ) { return 1 < * p ++ ; } : 14`
- ii. `{(identificador,f), (constante,1), (identificador, p)}`
- iii. `{{(C, ()), ({})} ó {LPAR, RPAR, LBRAC}`

b.

- i. Ninguna.
- ii. `<`.
- iii. `1 p p++ *p++ 1<*p++: 5`

c.

- i. `int *p; ó int i=0,*p=&i;`
- ii. `void *p; ó struct{double x,y;} p;`
- iii. Ni los puntero `void` ni las estructuras pueden desreferenciarse.

v1.0.0, 2018-10-04