UTN FRBA – SSL – Examen Final – 2018-02-19

Apellido, Nombre:		Legajo:		Nota:	

- Resuelva el examen en tinta y en esta hoja; no se aceptan hojas adicionales.
- Durante el examen no se responde consultas; si lo necesita, escriba hipótesis de trabajo, las cuales también se evalúan.
- Para los ítems de *selección múltiple*, tilde (✓) sólo una opción, la mejor.

1. *(2 puntos)* Escriba la *regex* que representa al LF:

$$L = \left(L_1\right)^C \cap L_2$$

   con L_1 igual al LF *identificadores de C* y L_2 igual al LF *constantes enteras decimales sin sufijo de C*.

2. *(2 puntos)* Calcule $Primero(S)$, siendo S el no terminal $sentIteración$ del BNF de C, es decir, la gramática de las *sentencias de iteración de C*:

3. *(1 punto)* Indique el orden en el cual se ejecutan los siguientes procesos o subprocesos:

   a. ( ) Parser.
   b. ( ) Linker.
   c. ( ) Scanner.
   d. ( ) Backend.
   e. ( ) Preprocesamiento.

4. Dado el fragmento: `{ T v,*p; v =f(1.0 ) ;switch( v.m1){p= v .m2;} }`

   a. *(1 punto)* Analice **léxicamente** y tilde la afirmación **falsa**:

      ❏ Es léxicamente correcto.
      ❏ Contiene dos tokens *punto* (`.`).
      ❏ Los espacios son intrascendentes en el análisis léxico.
      ❏ Al remover todos los espacios la cantidad de tokens varía.
      ❏ Agregar un espacio en cualquier lugar lo mantiene léxicamente correcto.

   b. *(1 punto)* Analice **sintácticamente** y tilde la afirmación **falsa**:

      ❏ Contiene una declaración.
      ❏ Contiene dos invocaciones.
      ❏ Contiene tres expresiones completas.
      ❏ Contiene por lo menos tres sentencias.
      ❏ Contiene por lo menos cinco subexpresiones.

   c. Escriba las declaraciones de los siguientes identificadores para que el fragmento sea **semánticamente** correcto:

      i. *(2 puntos)* `T`:
      ii. *(1 punto)* `f`:

1. Resolución

1. `[1-9][0-9]*`
2. $Primero(S) = \{$ `do`, `for` , `while` $\}$
3.
   a. (3) Parser.
   b. (5) Linker.
   c. (2) Scanner.
   d. (4) Backend.
   e. (1) Preprocesamiento.
4.
   a.
      ✓ Los espacios son intrascendentes en el análisis léxico.
   b.
      ✓ Contiene dos invocaciones.
   c.
      i. `typedef struct s{ int m1; struct s *m2; } T;`
      ii. `T f( double );`