

UTN FRBA – SSL – Examen Final – 2024-07-22

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



- Resuelva el examen en tinta y en esta hoja; no se aceptan documentos adicionales.
- Durante el examen no se responden consultas; si lo necesita, escriba hipótesis de trabajo, las cuales también se evalúan.

1. Para los siguientes conceptos de LF indique una aplicación durante el proceso de compilación o la construcción del compilador:
 - a. (1 punto) AF.
 - b. (1 punto) Algoritmo de Thompson.
 - c. (1 punto) RegEx.
 - d. (1 punto) GIC.
2. Dada la expresión $i++$:
 - a. (1 punto) Analice léxicamente.
 - b. (2 puntos) Declare i para que la expresión sea semánticamente *incorrecta*, *no la inicialice*. Justifique.
 - c. (1 punto) Declare i para que la expresión sea semánticamente *correcta*, *no la inicialice*. Justifique.
 - d. (1 punto) Declare e inicialice i para que la expresión produzca el valor 'B'.
 - e. (1 punto) Indique si la expresión es un *valor-l*. Justifique.
3. (Punto Extra) Explique y ejemplifique con C el concepto de ambigüedad sintáctica.

1. Una Resolución

1.
 - a. Durante el análisis léxico.
 - b. Para crear el analizador léxico.
 - c. Para crear el analizador léxico.
 - d. Para crear el analizador sintáctico.
2.
 - a. ((identificador, i), (++)
 - b. `int i[10];` // el postincremento no está definido para arrays.
 - c. `int i;` // el postincremento sí está definido para escalares.
 - d. `char i='B';` // el postincremento produce como valor su operando.
 - e. La expresión produce el valor antes del incremento y no permite localizar el objeto donde está localizado el nuevo valor incrementado.
3. Una GIC que permite para ciertos textos más de una derivación. Por ejemplo en C, una sentencia `if` anidada en otra y la asociación del `else` (Problema *dangling else*).

v1.1.0-rc.1 2024-07-23