

3.pdf



Anónimo



Procesadores de Lenguajes



3º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid**



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Examen Análisis Semántico 30 Junio de 2017

Enunciado

3. Dado el siguiente fragmento de gramática:

$E \rightarrow E_1 \text{ out } \{ E_2, E_3 \} \mid E_1 + E_2 \mid E_1 \text{ nand } E_2 \mid \text{id}$
 $S \rightarrow \text{if } E \text{ then } S_1 \mid \text{loop } E \text{ } S_1 \text{ endloop} \mid S_1 ; S_2$

El lenguaje tiene las siguientes características:

- El lenguaje tiene los tipos entero, lógico y cadena
- El lenguaje no tiene conversión automática de tipos
- Todas las variables se han de declarar antes de ser usadas
- El operador lógico out devuelve falso si el número E_1 es mayor que E_2 y menor que E_3
- El operador lógico nand representa la operación *not and*
- El operador + realiza una suma entre enteros y una concatenación entre cadenas
- La sentencia loop funciona de la siguiente manera: se evalúa la expresión E; si es cierta o si es un número mayor que cero, se ejecuta el cuerpo del loop y se vuelve a evaluar la expresión; si es falsa o vale cero o menos, se sale del loop
- La sentencia if se ejecuta si la expresión es cierta

Se pide diseñar el **Analizador Semántico** mediante una **Definición Dirigida por la Sintaxis**, explicando brevemente las funciones y atributos utilizados.

Solución

Examen julio-2017. Análisis Semántico	
$E \rightarrow E_1 \text{ out } \{ E_2, E_3 \}$	$E.\text{tipo} := \text{if } E_1.\text{tipo} = E_2.\text{tipo} = E_3.\text{tipo} = \text{entero} \text{ then lógico else tipo_error}$
$E \rightarrow E_1 + E_2$	$E.\text{tipo} := \text{if } E_1.\text{tipo} = E_2.\text{tipo} \in \{\text{entero}, \text{cadena}\} \text{ then } E_1.\text{tipo} \text{ else tipo_error}$
$E \rightarrow E_1 \text{ nand } E_2$	$E.\text{tipo} := \text{if } E_1.\text{tipo} = E_2.\text{tipo} = \text{lógico} \text{ then lógico else tipo_error}$
$E \rightarrow \text{id}$	$E.\text{tipo} := \text{BuscaTipoTS}(\text{id.entrada})$
$S \rightarrow \text{if } E \text{ then } S_1$	$S.\text{tipo} := \text{if } E.\text{tipo} = \text{lógico} \text{ then } S_1.\text{tipo} \text{ else tipo_error}$
$S \rightarrow \text{loop } E \text{ } S_1 \text{ endloop}$	$S.\text{tipo} := \text{if } E.\text{tipo} \in \{\text{lógico}, \text{entero}\} \text{ then } S_1.\text{tipo} \text{ else tipo_error}$
$S \rightarrow S_1 ; S_2$	$S.\text{tipo} := \text{if } S_1.\text{tipo} = \text{tipo_ok} \text{ then } S_2.\text{tipo} \text{ else tipo_error}$

Atributos y funciones usadas:

- E tipo: atributo que almacena el tipo de una construcción
- E BuscaTipoTS: devuelve el tipo asociado al identificador que recibe como parámetro
- E entrada: atributo que guarda la posición en la TS donde se encuentra un identificador