

2.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 Final Análisis Semántico 12 de Enero 2017

Enunciado:

PROCESADORES DE LENGUAJES Y COMPILADORES

12 de enero de 2017

Observaciones: 1. Las calificaciones se publicarán hacia el 23 de enero.
2. La revisión será hacia el 25 de enero.
3. En la web se avisará de las fechas exactas.
4. La duración de este examen es de 40 minutos.

3. Un fragmento de un lenguaje que permite realizar declaraciones de variables e inicialización de variables tiene la siguiente Gramática Sintáctica:

```
P → D P | I P | λ
D → T id
T → int | real | bool | matrix [cte-entera, cte-entera ] of T
I → init id V
V → E | E , V
E → cte-entera | cte-real | E + E | E and E | id [V] | id | true
```

Donde:

- Existe un único ámbito de definición de variables. La declaración previa de variables es obligatoria.
- Las matrices (matrix) tienen dos dimensiones. No se permite una matriz cuyos elementos sean otras matrices. El índice de las distintas dimensiones de la matriz siempre empieza por 1.
- La expresión de tipos para las matrices es: *array (1..númFilas, 1..númColumnas, tipoElemento)*
- No existe conversión automática de tipos en ningún caso.
- La operación suma (+) se aplica a enteros y reales y la operación and solo a lógicos.
- La sentencia init inicializa la variable id con el valor de V.
- La inicialización de las matrices deberá ser con una lista con el número exacto de valores para toda la matriz, separados por comas.
- Es posible inicializar una variable varias veces.
- El tipo entero ocupa 2 bytes, el real 4 y el lógico 1.

Se pide diseñar una **Definición Dirigida por la Sintaxis** para realizar el **Análisis Semántico** de este fragmento de lenguaje. Se deberán describir brevemente cualquier función auxiliar que se utilice.

Ejemplo de Programa válido:

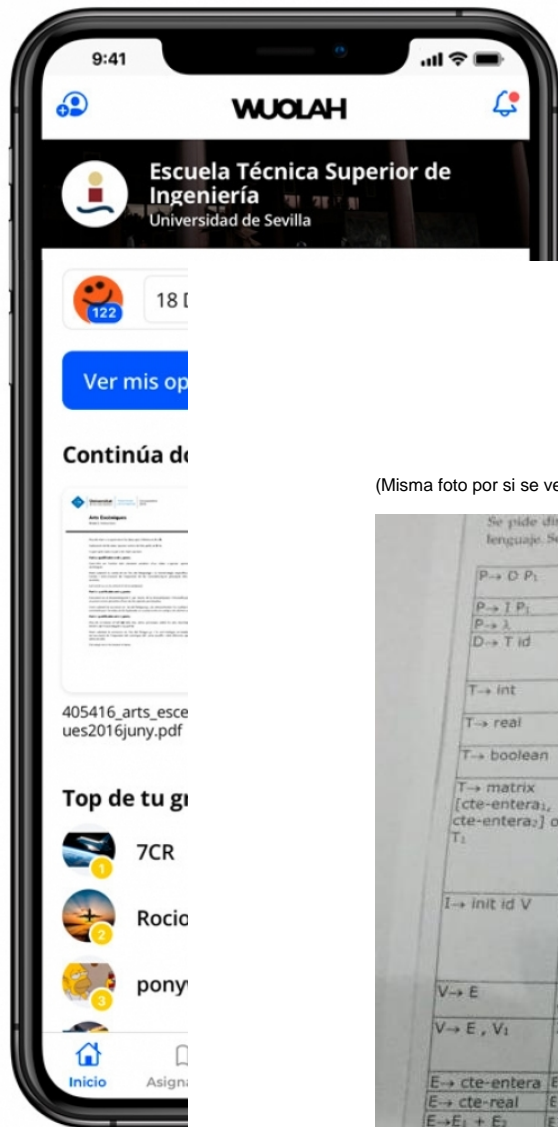
```
real a
int b
matrix [2, 3] of int mat
init a 1.73
init mat 7, 6, 8, 3, 6, 3
init b mat[1, 3 + b]
```

Solución:

P → D P ₁	zdecl := true zdecl := false
P → I P ₁	
P → λ	
D → T id	InsertaTipoTS (id.ent, T.tipo) InsertaDespTS (id.ent, despl) desp := desp + T.ancho // desp: variable del Procesador con la dirección de la variable
T → int	T.tipo := Entero T.ancho := 2
T → real	T.tipo := Real T.ancho := 4
T → boolean	T.tipo := Lógico T.ancho := 1
T → matrix [cte-enters ₁ , cte-enters ₂] of T ₁	If (T ₁ .tipo != array(1..n,1..m,t)) Then If (cte-enters ₁ =0 or cte-enters ₂ =0) Then Error ("Los índices de una matriz no pueden ser cero") Else T.tipo := array(1..cte-enters ₁ .val, 1..cte-enters ₂ .val, T ₁ .tipo) T.ancho := cte-enters ₁ .val * cte-enters ₂ .val * T ₁ .ancho Else Error ("No se permiten matrices de matrices")
I → init id V	I.tipo := If (BuscaTipoTS(id.ent)=array(1..n,1..m,t) and V.tipo=t and V.nval=n*m) Then tipo-ok Else If BuscaTipoTS(id.ent)=V.tipo Then tipo-ok Else tipo-error
V → E	V.tipo := E.tipo V.nval := 1
V → E, V ₁	If (V ₁ .tipo=E.tipo) Then V.tipo := E.tipo; V.nval := V ₁ .nval + 1 Else V.tipo := tipo-error
E → cte-enters	E.tipo := Entero
E → cte-real	E.tipo := Real
E → E ₁ + E ₂	E.tipo := If (E ₁ .tipo=E ₂ .tipo and E ₁ .tipo∈{Entero,Real}) Then E ₁ .tipo Else tipo-error
E → E ₁ and E ₂	E.tipo := If (E ₁ .tipo=E ₂ .tipo and E ₁ .tipo=Lógico) Then Lógico Else tipo-error
E → id [V]	E.tipo := If (BuscaTipoTS(id.ent)=array(1..n,1..m,t) and V.tipo=entero and V.nval=2) Then t Else tipo-error
E → id	E.tipo := BuscaTipoTS (id.ent)
E → true	E.tipo := Lógico

Las funciones utilizadas son:

- BuscaTipoTS: Recupera el tipo de un identificador en la Tabla de Símbolos
- InsertaTipoTS: Introduce el tipo de un identificador en la Tabla de Símbolos
- InsertaDespTS: Introduce la dirección de memoria de una variable en la Tabla de Símbolos
- Error: Envía un mensaje de error al gestor de errores



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



(Misma foto por si se ve mejor)

Se pide diseñar una estructura de datos para el lenguaje. Se deberán describir brevemente cuantos...

P → D P _i	zdecl := true zdecl := false
P → J P _i	
P → J	
D → T id	InsertaTipoTS (id.ent, T.tipo) InsertaDespTS (id.ent, desp) desp := desp + T.ancho // desp: variable del Procesador con la dirección de la variable
T → int	T.tipo := Entero T.ancho := 2
T → real	T.tipo := Real T.ancho := 4
T → boolean	T.tipo := Lógico T.ancho := 1
T → matrix [cte-enters ₁ , cte-enters ₂] of T ₁	If (T ₁ .tipo != array(1..n, 1..m, t)) Then If (cte-enters ₁ =0 or cte-enters ₂ =0) Then Error ("Los índices de una matriz no pueden ser cero") Else T.tipo := array(1..cte-enters ₁ .val, 1..cte-enters ₂ .val, T ₁ .tipo) T.ancho := cte-enters ₁ .val * cte-enters ₂ .val * T ₁ .ancho Else Error ("No se permiten matrices de matrices")
I → init id V	I.tipo := If (BuscaTipoTS(id.ent)=array(1..n, 1..m, t) and V.tipo=t and V.nval=n*m) Then tipo-ok Else If BuscaTipoTS(id.ent)=V.tipo Then tipo-ok Else tipo-error
V → E	V.tipo := E.tipo V.nval := 1
V → E, V ₁	If (V ₁ .tipo=E.tipo) Then V.tipo := E.tipo; V.nval := V ₁ .nval + 1 Else V.tipo := tipo-error
E → cte-enters	E.tipo := Entero
E → cte-real	E.tipo := Real
E → E ₁ + E ₂	E.tipo := If (E ₁ .tipo=E ₂ .tipo and E ₁ .tipo={Entero, Real}) Then E ₁ .tipo Else tipo-error
E → E ₁ and E ₂	E.tipo := If (E ₁ .tipo=E ₂ .tipo and E ₁ .tipo=Lógico) Then Lógico Else tipo-error
E → id [V]	E.tipo := If (BuscaTipoTS(id.ent)=array(1..n, 1..m, t) and V.tipo=entero and V.nval=2) Then t Else tipo-error
E → id	E.tipo := BuscaTipoTS (id.ent)
E → true	E.tipo := Lógico

Las funciones utilizadas son:

- BuscaTipoTS: Recupera el tipo de un identificador en la Tabla de Símbolos
- InsertaTipoTS: Introduce el tipo de un identificador en la Tabla de Símbolos
- InsertaDespTS: Introduce la dirección de memoria de una variable en la Tabla de Símbolos
- Error: Envía un mensaje de error al gestor de errores