Documento anónimo

2.pdf

Exámenes 1 Parcial Resueltos

3° Procesadores de Lenguajes

Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

Examen Análisis Léxico 26 de Octubre de 2016

Enunciado:

ANÁLISIS LÉXICO Y TABLA DE SÍMBOLOS

Primer examen. 26 de octubre de 2016

Observaciones: 1. Las calificaciones se publicarán hacia el 15 de noviembre.
2. La revisión será hacia el 17 de noviembre.
3. En la web se avisará de las fechas exactas.
4. La duración de este examen es de 40 minutos.

Se tiene un lenguaje de programación con las siguientes características:

- Existen declaraciones de funciones y de variables (globales). Sus nombres deben empezar por letra y pueden ir seguidos por cualquier cantidad de letras y dígitos. Los nombres pueden llevar también dólares ($) en su interior (pero no pueden ni empezar ni finalizar por dólar).
- Una función se declara poniendo la palabra reservada `function`, el tipo que retorna, su nombre y los tipos y nombres de los parámetros formales encerrados entre paréntesis y separados por comas. Los parámetros siempre se pasan por valor. A continuación vienen las declaraciones de variables (locales) y las sentencias encerradas entre las palabras reservadas `begin` y `end`.
- Una variable se declara poniendo la palabra reservada `var`, su nombre y su tipo (representado mediante las palabras reservadas `int` o `real`). El tipo entero ocupa 2 bytes y el real 4.
- Las sentencias son asignaciones (mediante `:=`) de expresiones a variables.
- Los operandos de las expresiones pueden ser variables, llamadas a funciones u otras expresiones. Las expresiones pueden llevar paréntesis.
- Se pueden realizar operaciones aritméticas sobre datos del mismo tipo. Los operadores disponibles son la suma (+), resta (-) y menos unario (-).
- Las llamadas a funciones se escriben poniendo los parámetros actuales entre paréntesis y separándolos por comas.

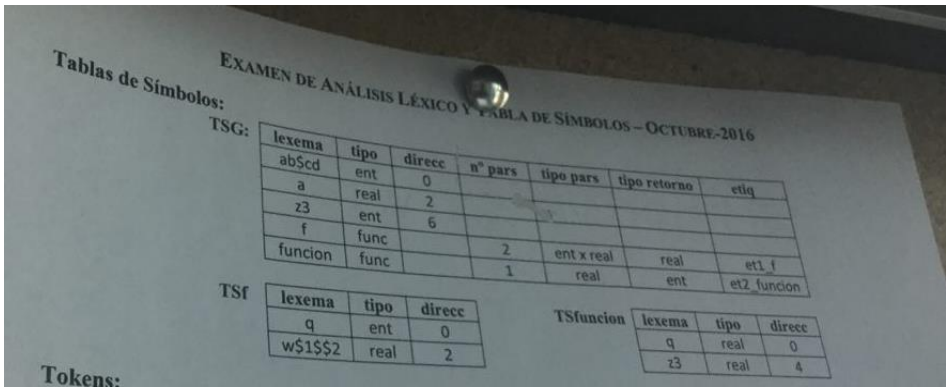
Teniendo en cuenta que los distintos elementos del lenguaje pueden ir separados por blancos, tabuladores o saltos de línea y que no hay distinción entre mayúsculas y minúsculas, se pide:

a. Construir las **Tablas de Símbolos** que generaría un Compilador para el ejemplo.
b. Diseñar un **Analizador Léxico** para este lenguaje (*Tokens*, Gramática Regular, Autómata Finito Determinista y Acciones Semánticas), que introduzca toda la información posible en la Tabla de Símbolos.

Ejemplo de un fragmento de programa en este lenguaje:

```
Var ab$cd    Int
Var    a Real
Var z3       Int
Function real f (int q, real w$1$$2)
Begin
 ab$cd:=-z3-ab$cd+q
 a := w$1$$2
End
Function int funcion (real q) var z3 real
begin z3 := - (q - f(-ab$cd, z3 + z3)) end
```

Solución:



EXAMEN DE ANÁLISIS LÉXICO Y TABLA DE SÍMBOLOS – OCTUBRE-2016

Tablas de Símbolos:

TSG:

lexema	tipo	direcc	nº pars	tipo pars	tipo retorno	etiq
ab$cd	ent	0				
a	real	2				
z3	ent	6				
f	func		2	ent x real	real	et1_f
funcion	func		1	real	ent	et2_funcion

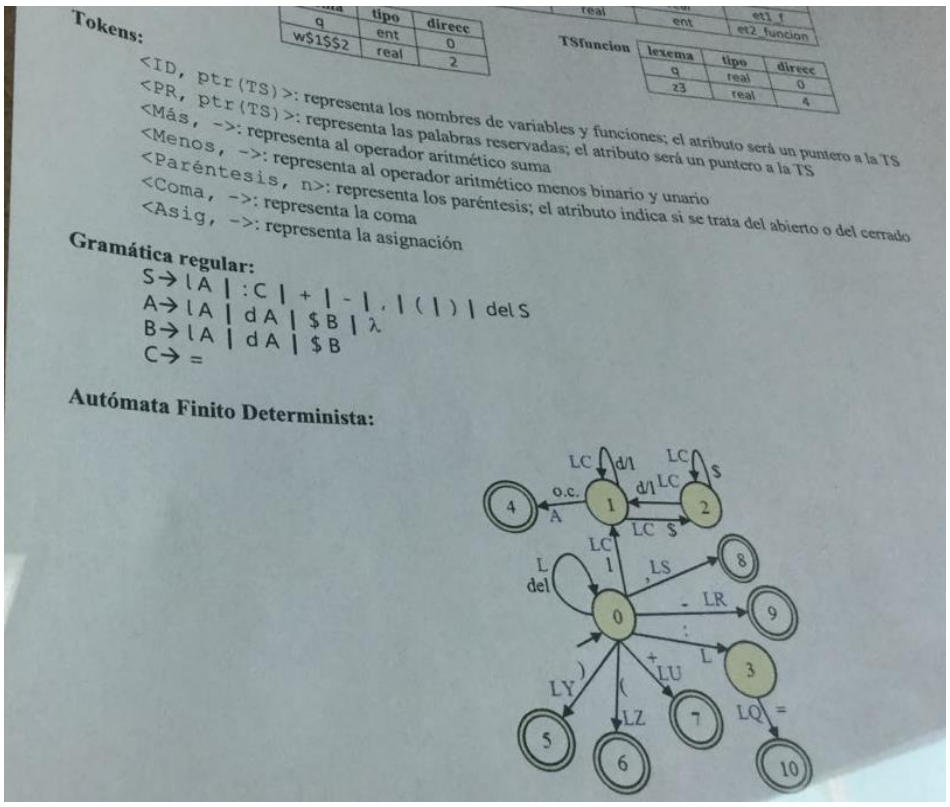
TSf

lexema	tipo	direcc
q	ent	0
w$1$$2	real	2

TSfuncion

lexema	tipo	direcc
q	real	0
z3	real	4

Tokens:



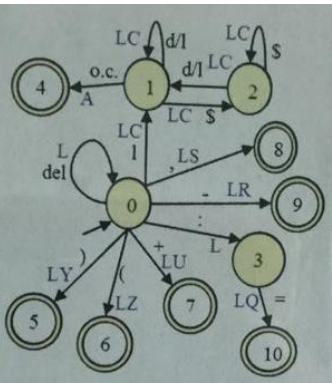
Tokens:

<ID, ptr(TS)>: representa los nombres de variables y funciones; el atributo será un puntero a la TS
<PR, ptr(TS)>: representa las palabras reservadas; el atributo será un puntero a la TS
<Más, ->: representa al operador aritmético suma
<Menos, ->: representa al operador aritmético menos binario y unario
<Paréntesis, n>: representa los paréntesis; el atributo indica si se trata del abierto o del cerrado
<Coma, ->: representa la coma
<Asig, ->: representa la asignación

Gramática regular:

S→ lA | :C | + | - | . | (|) | del S
A→ lA | dA | $B | λ
B→ lA | dA | $B
C→ =

Autómata Finito Determinista:



Acciones Semánticas:

A	p:= buscarTS (palabra)
	If (p ∈ PR) then GenToken (PR, p)
	If (p = NULL) then
	{
	If (ZonaDeclaración) then p:= insertarTS (palabra)
	Else Error ("Identificador no declarado")
	} else If (ZonaDeclaración) Error ("Identificador ya declarado")
	GenToken (ID, p)

C	Concatenar los caracteres en "palabra"
Error	Cualquier otra transición distinta de las recogidas correspondería a un caso de error
L	Leer el siguiente carácter de la entrada
Q	GenToken (Asig, -)
R	GenToken (Menos, -)
S	GenToken (Coma, -)
U	GenToken (Más, -)
Y	GenToken (Paréntesis, 2)
Z	GenToken (Paréntesis, 1)