Ingeniería Informática

PROCESADORES DE LENGUAJE

19 de Junio de 2018

Nombre: DNI:

1. (1.5 punto) **Análisis Léxico**

   Dada la especificación léxica siguiente, diseña el autómata finito necesario para analizar léxicamente la entrada.

Exp. regular	Elem. Léxico
<>	distinto
<	menor
>	mayor
+	opsuma
−	opsuma
++	incremento
−−	decremento
−>	desref
−−−>	dobleref
:=	asig

   Después indica claramente la segmentación en tokens que realizaría tu analizador léxico sobre la siguiente cadena de entrada:

```
<<>+++-> ----->
:==:- --->
```

   Indica, asimismo, qué caracteres son devueltos a la entrada (si los hay) por el analizador léxico antes de retornar cada uno de los tokens.

2. (1.5 puntos) **Análisis descendente**

   Demuestra que la siguiente gramática es LL(1) y construye su tabla de análisis sintáctico LL(1).

   $A \Longrightarrow B\ C$
   $B \Longrightarrow b\ c\ B \mid \lambda$
   $C \Longrightarrow c\ C \mid \lambda$

   Después haz la traza del análisis de un analizador descendente basado en la tabla obtenida anteriormente de las dos siguientes cadenas de entrada: "c" y "λ". Indica claramente la configuración de la pila, la entrada restante y la acción realizada en cada iteración del algoritmo de análisis sintáctico.

3. (1.5 puntos) **Análisis ascendente** Dada la siguiente gramática:

1) S $\Longrightarrow$ A b C
2) S $\Longrightarrow$ λ
3) A $\Longrightarrow$ λ
4) C $\Longrightarrow$ λ

Diseña el autómata aceptor de prefijos viables y haz la traza del análisis ascendente de las cadenas "b" y "λ"

4. (1.5 puntos) **ETDS**

   Dada la siguiente gramática, construye un esquema de traducción dirigida por la sintaxis (ETDS) que cuente e imprima el número de símbolos "b" que aparecen en la entrada justo a continuación de un símbolo "a".

   1) S $\Longrightarrow$ a b A
   2) A $\Longrightarrow$ a b c A
   3) A $\Longrightarrow$ c b c A
   4) A $\Longrightarrow$ λ

   Por ejemplo, ante una entrada como "ababccbc", el ETDS debe imprimir el valor 2; la impresión se debe hacer en la regla del no terminal S.

   El ETDS no puede utilizar ninguna variable global: toda la información debe pasarse a través de atributos. Indica claramente de qué tipo es cada uno de los atributos que utilices y cuál es su cometido.

5. (1.5 puntos) **Generación de código**

   Dada la gramática de la instrucción "REPEAT-UNTIL", completarla de forma que se genere el código correspondiente a la instrucción.

```
InstRepeat -> repeat { Bloque } until ( Condicion );  }
Bloque -> ...
Condicion -> ...
```

   Considérese que la función $Bloque$ devuelve un conjunto de instrucciones, que podrían incluir la traducción de un "break" o "continue". La función $Condicion$ devolverá el código necesario para ejecutar la condición y sus saltos.

   Asímismo, se dispone del método getNewLabel(), que devuelve una nueva etiqueta y del método getNewTemp() que devuelve una referencia a una nueva variable temporal.

6. (1.5 puntos) **Optimización** Define "Bloques Básicos", el algoritmo de partición y la construcción del grafo.

7. (1 puntos) **Gestión de la memoria** Describe el procedimiento de llamada y retorno de una función en una gestión de memoria de pila.