ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

Grado en Ingeniería Informática

PROCESADORES DE LENGUAJE

1 de septiembre de 2017

Nombre: DNI:

- No está permitido entregar ejercicios escritos a lápiz.

---

1. (2.5 puntos) **Análisis léxico.**

   a) Describir la "Creación de un AFN a partir de una expresión regular".

   b) Describir, lo más detalladamente posible, el algoritmo de "Creación de un AFD a partir de un AFN".

   c) Describir las características del algoritmo de "Creación de un AFD a partir de una expresión regular".

   d) Describir el algoritmo de "Minimización de estados de un AFD".

   e) Para la siguiente expresión regular: $(00 \mid 0^*1)$:

   - Calcular el AFN asociado.
   - Convertir el anterior AFN en AFD.
   - Minimizar el AFD obtenido.

2. (3 puntos) **Análisis ascendente**

   Para el algoritmo LR(0):

   a) Describir cómo se obtiene "la colección canónica de elementos" y el "autómata reconocedor de prefijos viables".

   b) Describir la construcción de la tabla de análisis LR(0).

   c) Describir la diferencia con el algoritmo SLR.

   d) Para la siguiente gramática:

   $$\begin{aligned} S &\rightarrow L\ ,\ S \\ S &\rightarrow L \\ L &\rightarrow (\ L\ ) \\ L &\rightarrow a\ ,\ L \\ L &\rightarrow \epsilon \end{aligned}$$

Calcula todo lo necesario para hacer el análisis SLR de la cadena

$$((), (a, a))$$

e indica si esta está generada por la gramatica o no.

3. (2.5 puntos) **ETDS**

Dada la siguiente gramática define el lenguaje de los números romanos menores de 100:

S	$\rightarrow$	Decenas Unidades
Decenas	$\rightarrow$	Dec
Decenas	$\rightarrow$	**XL**
Decenas	$\rightarrow$	**L** Dec
Decenas	$\rightarrow$	**XC**
Dec	$\rightarrow$	Dec **X**
Dec	$\rightarrow$	ϵ
Unidades	$\rightarrow$	Unid
Unidades	$\rightarrow$	**IV**
Unidades	$\rightarrow$	**V** Unid
Unidades	$\rightarrow$	**IX**
Unid	$\rightarrow$	Unid **I**
Unid	$\rightarrow$	ϵ

Escribe un esquema de traducción dirigida por la sintaxis (ETDS), basado en la gramática anterior sin modificar, que permita obtener en un atributo sintetizado de S el valor decimal del número romano definido. Además, el ETDS ha de restringir el número de **X** en Dec y de **I** en $Unid$ a 3, es decir, que si en la entrada aparecen más de 3 veces consecutivas uno de estos símbolos se ha de emitir el error semántico oportuno.

Por ejemplo, para una cadena de entrada como **XXI** la traducción obtenida deberá ser 21 y para la cadena de entrada **VIIII** se obtendrá un error semántico.

Toda la información debe pasarse a través de atributos y no es posible utilizar ninguna variable global. Debes indicar claramente, además, de qué tipo es cada uno de los atributos que utilizas y cuál es su cometido.

4. (2 puntos) **Generación de código y JavaCC**

   La siguiente figura muestra la descripción sintáctica del bucle **REPEAT-UNTIL** en el formato de la herramienta JavaCC.

```
void InstRepeat() :
{}
{
  <REPEAT> <LLAVEAB> ( Instruccion() )* <LLAVECE>
  <UNTIL> <PARAB> Condicion() <PARCE> <PYC>
}
```

   Se pretende enriquecer la gramática anterior para construir el código intermedio de 3 direcciones correspondiente a la instrucción **REPEAT**. Para ello la función $InstRepeat()$ deberá devolver un objeto $Inst$ cuyo campo $code$ contenga dicho código.

   Considere que la función $Condicion()$ devuelve un objeto de la clase $Condition$ con los siguientes campos: $code$, código intermedio que describe la condición; $label\_true$, etiqueta a la que salta el control en el caso de que la condición sea cierta; $label\_false$, etiqueta a la que salta el control en caso de que la condición sea falsa.

   Por su parte, la función $Instruccion()$ devuelve un objeto de la clase $Inst$ cuyo campo $code$ contiene el código intermedio asociado a a dicha instrucción. Asímismo, se dispone del método $getNewLabel()$, que devuelve una nueva etiqueta (es decir, una etiqueta no utilizada en ningún punto del código) y del método $getNewTemp()$ que devuelve una referencia a una nueva variable temporal.