Universidad de La Laguna. Escuela Técnica Superior de Ingeniería Informática Tercero del Grado de Informática

PROCESADORES DE LENGUAJES. CONVOCATORIA DE JUNIO

06/06/2017 2 páginas

Nombre:		
Alu:	GitHub Id:	

1. Escriba un PEGJS que recibe como entrada una expresión regular como a(b.c)+|abc y produce como salida un árbol de análisis sintáctico:

En esta representación del árbol AST hemos usado un formato muy simple mediante arrays: el primer elemento del array indica el tipo de nodo (por ejemplo 'ALT' para indicar una alternativa o unión) y el resto de elementos en el array son los hijos de ese nodo.

No tiene por que ser esta la representación que use para su AST: cualquier representación razonable que respete la prioridad y asociatividad de las operaciones y facilite el recorrido del AST será considerada correcta.

Las expresiones regulares que consideramos vienen definidas por las siguientes reglas:

- 1. Si re1y re2 son dos regexps entonces re1 | re2 es una regexp
- 2. La concatenación de regexps es una regexp
- 3. Los cierres de regexps re como
- re?
- re*

- re+ son regexps
- 4. Si re es una regexp entonces (re) entre paréntesis es también una regexp
- 5. El punto '.', el '^' y el '\$' son regexps
- 6. Los caracteres que no son metasimbolos (esto es, distintos de '*', '+', '|', etc.) son expresiones regulares
- 7. Los metasímbolos (como '*', '+', '|', etc.) cuando son escapados '*' se consideran caracteres y son también expresiones regulares
- 8. Si no ha superado la segunda parte deberá incluir en su lenguaje de regexps las clases o conjuntos:
- Una clase es una secuencia de items entre corchetes: [items] es una regexp
- Los items pueden ser bien carácteres o bien rangos char char