

PRÁCTICA FINAL (2015)

(grupos con 3 personas)

Objetivo:

El objetivo las prácticas de TAC es construir un analizador léxico y sintáctico de un lenguaje que se especifica a principio de curso, usando las herramientas Jflex y Cup. La práctica se realizará por módulos, y de forma incremental en las distintas sesiones de prácticas, según se avance en las sesiones de teoría.

En cada sesión, se explicará, en su caso, la parte de la herramienta y/o práctica que se deberá realizar para la siguiente sesión y se revisará la práctica propuesta en la sesión anterior. No es obligatorio seguir esta secuencia pero es muy recomendable

Enunciado de la práctica:

La herramienta SELFA es el resultado de un proyecto fin de carrera que desarrolla un software web para la definición de expresiones regulares, gramáticas, autómatas finitos y autómatas a pila, así como para la realización de diferentes operaciones sobre estos elementos. La herramienta se puede encontrar en :

<http://portal.esi.uclm.es/selfa/indexCompilar.php>

El objetivo de esta práctica es diseñar, usando Jflex y Cup, un analizador léxico y sintáctico para un subconjunto del lenguaje usado por selfa. El fichero HowToCastellano.pdf define todos los componentes de selfa.

Los componentes del lenguaje de la práctica son:

- Definición de expresiones regulares (Ers).
- Definición de autómatas finitos (Afs).
- Definición de gramáticas.
- Definición de dos operaciones para Ers (a determinar para cada grupo).
- Definición de dos operaciones para Afs (a determinar para cada grupo).
- Definición de dos operaciones para gramáticas (a determinar para cada grupo).
- Posibilidad de tener comentarios en el texto, del tipo /* */

Las características de este lenguaje son:

- El lenguaje se estructura como un bloque de definiciones (ERs y/o Afs) seguido de un bloque de operaciones sobre los elementos definidos
- La definición de los componentes de selfa y sus operaciones viene especificada en el documento: [HowToCastellano.pdf](#)

Ejemplo de texto que pertenece a un lenguaje:

```
regex e2 { 'hola' }
automaton prueba{

    states q0,q1,q2;
    alphabet a,b,c;
    initial q0;
    final q2;
    transition{

        q0,a = q1;
        q1,b = q1;
        q1,c = q2;
    }
}
regex e1 { (a*)b }

grammar gr{
    terminal 0,1;
    nonterminal S,A;
    axiom S;
    productions{
        S := 0 S;
        S := 1 A;
        S := 0;
        S := 1;
        A := 1 A | $;
    }
}
print(prueba);
exp1 = FatoRE(prueba);

print(e1);

print(e2);
exp2 = alternation(e1,e2);

/* OPERACIONES */

print(gr);

a = RGtoFA(gr);
```