**COMPILADOR DECAF – DAVID CORZO & IAN JENATZ**

**FASE 1 – SCANNER Y LEXER:**

DOCUMENTACIÓN DE FUNCIONES GLOBALES:

1. def add\_edge:
   * Esta función chequea el diccionario de estados para poder insertar una llave y valor, primero chequea si la llave que corresponde a épsilon existe en el diccionario, si no existe hace un update agregando la llave y el valor (que es un set), si existe simplemente agrega al set correspondiente. Si la llave no es épsilon simplemente se hace un update al número como valor puesto a que sólo épsilon puede tener varias aristas, todo lo demás tiene aristas únicas.
2. def num\_label\_maker:
   * Generador para poder tener nombres estados únicos dentro del diccionario de estados, pero produce números únicos para nombrar los estados. Usados en el diccionario de estados.
3. def letter\_label\_maker:
   * Generador para poder tener nombres estados únicos dentro del diccionario de estados, lo mismo que la anterior, pero produce combinaciones de letras en lugar de números para nombrar los estados. Usados en estados de DFA.

DOCUMENTACIÓN DE CLASES USADAS:

1. class enfa:
   * Esta clase sirve como un 'struct' en el que se almacena el estado inicial y final del NFA. Sólo almacena valores y los métodos sólo sirven para imprimir el 'struct'
2. class dfa:
   * Sirve la exactamente la misma función que la clase anterior, pero se utiliza para el DFA.
3. class NFA:
   * Esta clase toma una expresión regular y devuelve un épsilon-nfa. Contiene los siguientes métodos:
     1. infix\_to\_postfix: Este método usa el atributo de clase 'self.infix', que es una expresión regular en forma normal, la transformamos a una postfix por conveniencia del algoritmo. Luego procede a hacer una expresión regular que está en forma de postfix. Ejemplo: a+b+(c+d) -> cd+a+b+
     2. thompson\_construction:
     3. regex\_or:
     4. regex\_character:
     5. regex\_concatenation:
     6. regex\_question\_mark:
     7. regex\_kleene:
4. class DFA:
5. class scanner:
6. class error\_msg:

**FASE 2 - PARSER:**

**FASE 3 - SEMANTIC:**

**FASE 4 – GENERACIÓN DE CÓDIGO:**