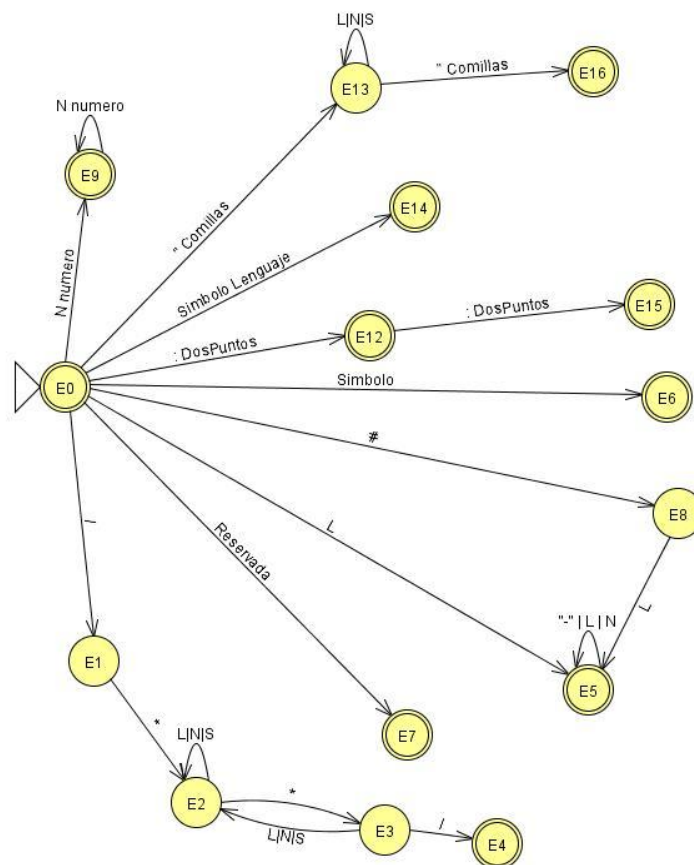


MANUAL TECNICO PROYECTO 1 - ML WEB

Para la realización de los AFD de cada Analizador se utilizó la herramienta JFLAP. Estos AFD fueron contruidos a partir de patrones y símbolos encontrados dentro del lenguaje. A continuación se presentan los AFD contruidos. En el caso de las palabras reservadas se tiene solo un estado (diferente de los estados para reconocer ID) en donde se acepta la palabra reservada directamente en cada lenguaje. Tambien para los simbolos de cada lenguaje, se definio un solo estado, a diferencia de los dobles como por ejemplo ==, !=, etc.

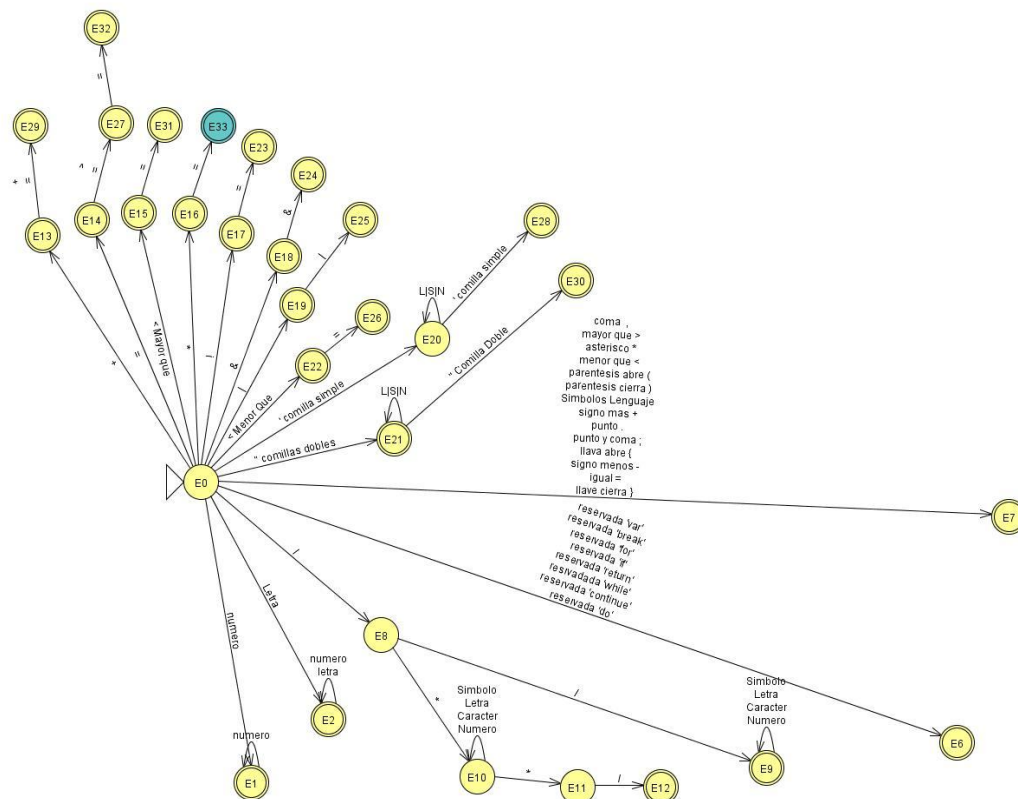
AFD CSS



En el caso del lenguaje CSS las expresiones regulares encontradas dentro de las restricciones son las siguiente:

- Para el ID la expresion regular $(L | \#)(L|D)^*$, en donde L = Letra, D = Dígito (numero), esto por que un ID podria ser definido con un #id o simplemente sin el simbolo #.
- Para los comentarios se definió solamente multilinea: `/"" (L|N|S)* ""/` en donde L = Letra , N = Numero, S= Simbolo, ya que en un comentario puede incluirse cualquier carácter y sera valido.
- Para el numero se definio la expresion $(N)^+$ ya que para que sea reconocido como un numero este debe contener al menos un digito.
- En el caso de las cadenas se definio $(\text{""})(L|N|S)^*(\text{""})$, siendo L=Letra, N=Numero, S=Simbolo, ya que en una cadena se puede almacenar cualquier tipo de carácter, en este caso.

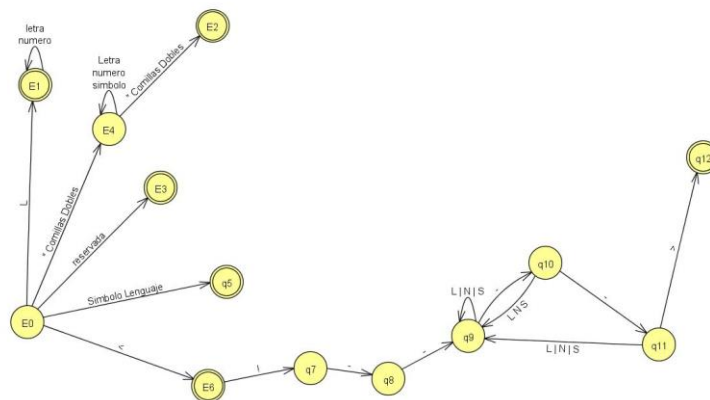
AFD JS



El analizador del Lenguaje JS, siendo el mas especificado se obtvieron muchos simbolos de lenguaje y palabras reservadas, consecuencia se generaron muchos mas estados.

- Para el ID la expresion regular (L)(L|D|_)*, en donde L = Letra, D = Dígito (numero), esto por que un ID podria ser definido con un guion en medio y este seria reconocido.
- Para los comentarios se definió solamente multilinea: /(“*” | /) (L|N|S)* (“*” | /)/ en donde L = Letra , N = Numero, S= Simbolo, ya que en un comentario puede incluirse cualquier carácter y sera valido. En este caso, el lenguaje permite dos tipos de comentarios, uno multilinea y otro unilinea, estos son definidos por medio del (“*” | /) colocados al inicio y al final de la expresion regular.
- Para el numero se definio la expresion (N)+ ya que para que sea reconocido como un numero este debe contener al menos un dígito.
- En el caso de las cadenas se definio (“)(L|N|S)* (“) , siendo L=Letra, N=Numero, S=Simbolo, ya que en una cadena se puede almacenar cualquier tipo de carácter, en este caso.

AFD HTML



Para el analizador del HTML se trabajo ignorando lo que hay fuera de los simbolos <> , por ejemplo ignorado <no ignorado> ignorado, y gurdandolo como un tipo de token diferente. Esto para posteriormente realizar la correccion de errores.

- Para el ID la expresion regular (L)(L|D)*, en donde L = Letra, D = Dígito (numero).
- Para los comentarios se definió: <!--(L|S|N)*--> en donde L = Letra , N = Numero, S= Simbolo, debiendo cumplir con este patron, ya que si no, no es reconocido como multilinea.
- En el caso de las cadenas se definio (")(L|N|S)*(") , siendo L=Letra, N=Numero, S=Simbolo, ya que en una cadena se puede almacenar cualquier tipo de carácter, en este caso.

Para cada lenguaje se definio una lista de tokens definidas con una clase enumerada, esto para tener mejor organizados los tokens y poder realizar analizadores sintacticos posteriormente.

Al realizar la recuperacion de los archivos se tomo en cuenta el path inicial de cada entrada. Se uso la misma lista de tokens, guardando espacios, tabuladores y saltos de linea como un tipo de token .DESCONOCIDO, y asi poder realizar una cadena limpia de errores.