



Validación de un autómata en Java

Resumen

El programa tiene que validar los diferentes tipos de recorridos de programa que están compuestos por algunas palabras reservadas así como sintaxis, aquí se muestran algunos ejemplos del funcionamiento que va hacer el programa el bloque izquierdo de código es como va a estar validado en el programa ósea la palabra ON;_ $\leftarrow +$ es igual a ENCENDER; ya que el “_” es como si fuera un espacio y “ $\leftarrow +$ ” es como si fue uno o varios saltos de líneas.

En dado caso que el usuario no ingresa correctamente cada línea de código con lo que se indica va a mostrar un mensaje de error, por eso es la tabla de transiciones y errores

Modo de uso

El archivo `EDF.class` es el ejecutable. Para hacer uso del mismo se usa la siguiente sintaxis, desde una consola o CMD:

```
java EDF ejemplo1
```

`ejemplo1` representa el nombre de un archivo donde viene la sintaxis del lenguaje que se debe de validar en el autómata.

```
ENCENDER;  
PARPADEAR(p1);  
VEL={1};  
VEL={2};  
TIME_S={10};  
LICUAR(2,10);  
PARPADEAR(p1);  
APAGAR;
```

Ejemplo de una corrida

Si se ejecuta el siguiente archivo llamado `pruebadefinitiva.txt` :

```
ENCENDER;  
PARPADEAR(p1);  
VEL={1};  
APAGAR;
```

Se usa este comando:

```
java DFA pruebadefinitiva.txt
```

El resultado será el siguiente:

```
Iniciando el análisis...  
-----  
Contenido del archivo:  
ENCENDER;  
PARPADEAR(p1);  
VEL={1};
```

```
APAGAR;
```

```
-----  
Cadena de tokens generada: ON-pc-es-en-PP-pi-pl-pf-pc-es-en-VL-ig-li-1-lf-pc-es-en-OFF-pc-es-en-  
Ok  
-----
```

```
Análisis finalizado. Sintaxis válida.
```

Ejemplo de una sintaxis incorrecta

El siguiente archivo contiene una sintaxis incorrecta:

```
ENCENDER;  
PARPADEAR(pl);  
VEL={1};  
TIME_S={5};  
LICUAR(1,5);  
STOP(1,5);  
PARPADEAR(pl);  
LICUAR(1,5);  
STOP(pl);  
APAGAR;
```

La sintaxis en el comando `STOP` se está intentando agregar un argumento invalidó (pl en vez de dos números separados por coma).

Si ejecutamos `java DFA ejemploincorrecto.txt` nos arroja el siguiente resultado:

```
Iniciando el análisis...
```

```
-----  
Contenido del archivo:
```

```
ENCENDER;  
PARPADEAR(pl);  
VEL={1};  
TIME_S={5};  
LICUAR(1,5);  
STOP(1,5);  
PARPADEAR(pl);  
LICUAR(1,5);  
STOP(pl);  
APAGAR;
```

```
-----  
Cadena de tokens generada: ON-pc-es-en-PP-pi-pl-pf-pc-es-en-VL-ig-li-1-lf-pc-es-en-TI-ig-li-5-lf-pc-es-en-LC-pi-1-co-5-pf-pc-es-en-ST-pi-1-  
Exception in thread "main" java.lang.Error: E12 - Error de sintaxis: se esperaba un valor entre el 1,2 o 3  
    at DFA.delta(DFA.java:341)  
    at DFA.process(DFA.java:503)  
    at DFA.main(DFA.java:528)
```

Si vemos detenidamente en el cuerpo del error, vemos que se está llamando a nuestra tabla de errores la cual definimos previamente junto a la tabla de transiciones.

Explicación del código

El un solo archivo que contiene todos los métodos y clases. Algunos de los principales componentes son:

- Clase DFA
 - (DFA (Deterministic Finite Automaton)) Es la clase madre donde todos los demás están dentro.
- Clase Parser
 - `parseFile` Se encarga de leer los archivos externos.
 - `tokenize` Transforma el texto obtenido en pequeños trozos para un posterior análisis.
- Variable `ErrorTable`
 - Una variable que tiene los datos de errores.

- Delta
 - Es el método que hace el recorrido por todo el autómata.