

Definición de estados léxicos

Podemos definir estados que se asocian a un conjunto de expresiones regulares, de forma que se cumpla:

- ❑ Solo puede haber un estado activo en un instante dado.
- ❑ La entrada actual solo podrá tener la estructura de uno de los lenguajes representados por el estado activo.

Sintaxis

En la sección de definiciones, definimos **n** estados de la siguiente forma :

%state <nombre_1>,.....<nombre_n>

En la sección de expresiones regulares:

<nombre_i>

ER1 { ----- }

Definición de estados léxicos

O, si hay mas de una E. Regular para el mismo estado

```
<nombre_i>{  
    ER1 { ----- }  
    .  
    .  
    ERn { ----- }  
}
```

- ❑ Hay un estado definido por defecto y es el estado activo al comienzo. El nombre de este estado es YYINITIAL.
- ❑ Para cambiar el estado activo se debe ejecutar el método yybegin(nombre), cuyo argumento es el nombre del estado que queremos que esté activo.
- ❑ Si la definición de una E. Regular no tiene asociado ningún estado, el comportamiento del sistema permite que lo que hay en la entrada pueda ser identificado con la estructura de ese lenguaje.

Definición de estados léxicos

O, si hay mas de una E. Regular para el mismo estado

```
<nombre_i>{  
    ER1 { ----- }  
    .  
    .  
    ERn { ----- }  
}
```

- ❑ Hay un estado definido por defecto y es el estado activo al comienzo. El nombre de este estado es YYINITIAL.
- ❑ Para cambiar el estado activo se debe ejecutar el método yybegin(nombre), cuyo argumento es el nombre del estado que queremos que esté activo.
- ❑ Si la definición de una E. Regular no tiene asociado ningún estado, el comportamiento del sistema permite que lo que hay en la entrada pueda ser identificado con la estructura de ese lenguaje.

Definición de estados léxicos

Ejemplo: Detectar la posición par o impar de los naturales de un fichero

```
%%
```

```
%class pares  
%public  
%switch  
%standalone  
%state pares
```

```
%%
```

```
<pares> [0-9]+ { System.out.println("Par : "+yytext());  
                yybegin(YYINITIAL) ;}  
<YYINITIAL> [0-9]+ { System.out.println("impar : "+yytext());  
                    yybegin(pares) ; }  
. {}
```

Ejercicios propuestos

Ejercicios propuestos

1. Revisar los problemas de la práctica 1 para ver en qué casos es útil aplicar la noción de estado. Rehacer dichos problemas usando estados.
2. Dado un fichero de texto, Definir un corrector del fichero que ponga en mayúscula cualquier letra en minúscula después de un punto.
3. Realizar una especificación en JFlex para que, dado un entero en la entrada, se nos diga si el entero es o no múltiplo de tres.