

Universidad del Valle de Guatemala
Facultad de Ingeniería
Diseño de Lenguajes de Programación



Laboratorio 1

Dilary Sarahí Cruz López – 231010

Repaso de conceptos

1. Defina que hace la función retractar dentro del proceso de reconocimiento de tokens. Explique:

- En que momento se invoca esta función:
 - Cuando se leyó la parte de la cadena antes de hacer el match, su función es mover el puntero y continuar leyendo la cadena pero empezando como desde cero porque ya tiene un match.
- Que sucede con el puntero del buffer al ejecutarse:
 - El lexemebeing reconoce el inicio del autómata y seguirá leyendo lo mismo mientras el forward sigue avanzando en la cadena y leyendo token hasta que se haga match.
- Por que es necesaria para aceptar correctamente un token.
 - Porque el token sigue leyendo caracteres pero el retract hace un stop o pausa para no caer en un exceso de lectura o para que no caiga en otras funciones como fallo.
- Como influye en el comportamiento del autómata finito.
 - Es necesario para que el autómata empiece en el lugar correcto y sea un éxito de reconocimiento.

2. Defina que hace la función fallo en el contexto del análisis léxico.
Explique:

- Que significa que un autómata falle.

- Es cuando el carácter no es reconocido como parte de un token, cambia de autómata y se reinician los punteros.
- Que ocurre cuando no existe una transición válida para el carácter leído.
 - Tira “fallo” y regresa al inicio del autómata para volver a leer el token. Si en dado caso cae en bucle, tira error y no se acepta la cadena.
- Como permite esta función intentar el reconocimiento de otro patrón.
 - Cuando el autómata no reconoce la cadena, reajusta el puntero, al igual que el diagrama.
- Su relación con la estrategia de máxima coincidencia.
 - Reconocer el token de mayor longitud.

3. Defina la función error dentro del proceso de análisis léxico. Su explicación debe incluir:

- En qué situación se produce un error léxico.
 - Cuando no se llega a un estado de aceptación, se reporta el error y se escoge un método de recuperación.
- Que ocurre cuando ningún autómata reconoce la secuencia de entrada.
 - Se decide limpiar para continuar con el flujo porque ya trono y no pertenece a la gramática de tokens.
- Como se reporta el error.

- Se genera un mensaje que indica que ocurrió un error léxico, especificando el símbolo o secuencia no reconocida y su ubicación.
 - Que mecanismos de recuperación pueden aplicarse.
 - Se pueden aplicar distintos mecanismos como eliminar el carácter inválido y continuar con el siguiente, descartar la secuencia de caracteres hasta encontrar un delimitador válido
4. Provea un ejemplo como el que se vió en clase.
- El lenguaje reconoce operadores como `<` y palabras reservadas como `while`. Si la entrada es `<+while`, el analizador léxico primero reconoce correctamente el símbolo `<` como operador relacional. Al encontrar la secuencia `+while`, el autómata intenta reconocer el símbolo `+`, pero al no formar parte de una secuencia válida junto con lo que sigue, puede producirse un error dependiendo de la gramática definida. Si la secuencia completa no coincide con ningún patrón válido, el analizador no alcanza un estado de aceptación y se genera un error léxico. En ese momento, se reporta el error indicando la secuencia no reconocida y se aplica un mecanismo de recuperación, como descartar el símbolo inválido y continuar con el análisis, permitiendo luego reconocer correctamente la palabra `while` como identificador o palabra reservada.

Problema 2

```
dilar@DESKTOP-4NUURCE:/mnt/c/Users/dilar/OneDrive/Desktop
/Semestre7/Diseño/Lab1$ flex lexer.l
dilar@DESKTOP-4NUURCE:/mnt/c/Users/dilar/OneDrive/Desktop
/Semestre7/Diseño/Lab1$ gcc lex.yy.c -o lexer -lfl
dilar@DESKTOP-4NUURCE:/mnt/c/Users/dilar/OneDrive/Desktop
/Semestre7/Diseño/Lab1$ ./lexer
dilar@DESKTOP-4NUURCE:/mnt/c/Users/dilar/OneDrive/Desktop
/Semestre7/Diseño/Lab1$ ./lexer
int x = 10;
IDENTIFICADOR: int
IDENTIFICADOR: x
ASIGNACION
ENTERO: 10
ERROR LEXICO: ;
|
```

- Punto 6
 - El analizador léxico se probó con código desde Sublime Text, donde reconoció correctamente identificadores, números enteros y flotantes, operadores aritméticos y relacionales, además de descartar comentarios y reconocer cadenas con secuencias de escape. Esto demuestra que el lexer funciona correctamente en lenguajes similares a Java.
- Punto 7
 - En Sublime Text el resaltado de sintaxis se logra mediante reglas de coloreado definidas en el editor. Para implementar algo similar con nuestro lexer, sería necesario generar tokens y asociarlos a diferentes estilos de color. Sin embargo, Flex solo realiza análisis léxico y no maneja directamente la presentación visual, por lo que el coloreado depende del editor.

- Punto 8
 - Una herramienta similar a Flex es ANTLR, que permite generar analizadores léxicos y sintácticos a partir de una gramática formal. A diferencia de Flex, ANTLR también construye el parser y soporta múltiples lenguajes como Java y Python.

Link del repositorio: https://github.com/dils1809/Dise-o_de_Lenguajes