

Escuela Superior de Cómputo

Autómatas del analizador léxico C

Compiladores

Cruz Cruz Jorge
Feregrino Bolaños Antonio
Gonzales Rodríguez Florencio Javier
6-3-2013

Introducción

Dada la complejidad de realizar un analizador léxico, se tiene que dividir en varias partes, cada una controlada por un autómata. Los que empleamos para llevar a cabo el nuestro fueron los siguientes. Los ordenamos según la precedencia (mayor a menor)

Consideraciones

Dado el tamaño de algunos autómatas, se ha simplificado la notación en algunos casos mediante el uso de agrupaciones.

Autómata que reconoce identificadores malformados

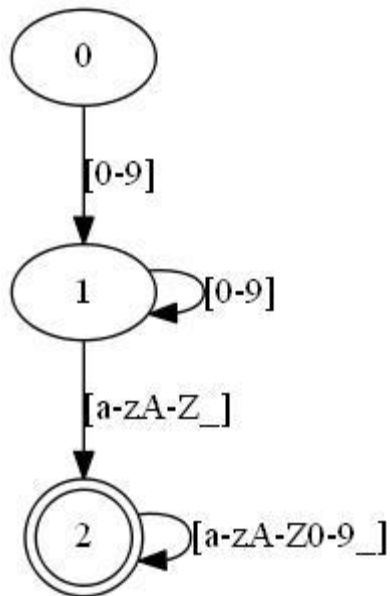
Este autómata se encarga de encontrar errores en los identificadores, reconocerá cadenas del tipo:

- 4fx
- 444tw_
- 8yx

Expresión regular

$[0-9]^+[a-zA-Z0-9_]^+$

Representación gráfica



Autómata que reconoce números flotantes

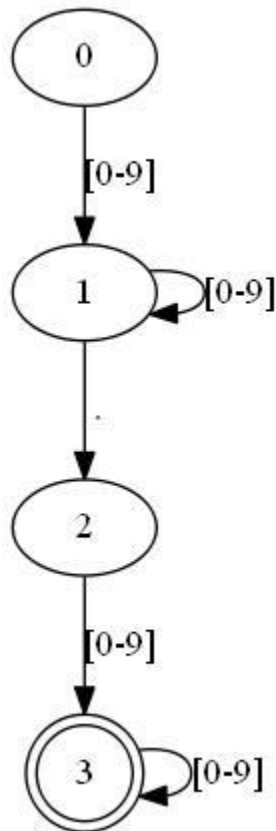
Encargado de interceptar las expresiones de tipo

- 0.11111
- 90.32
- 100.0001

Expresión regular

$[0-9]^+ \cdot [0-9]^+$

Representación gráfica



Autómata que reconoce números enteros

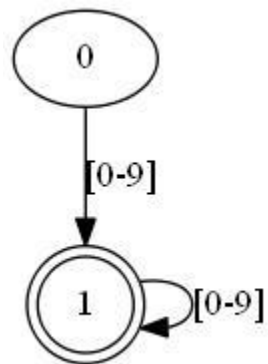
Importante para el analizador, se encarga de validar las expresiones:

- 24212
- 990231
- 1

Expresión regular

$[0-9]^+$

Representación gráfica



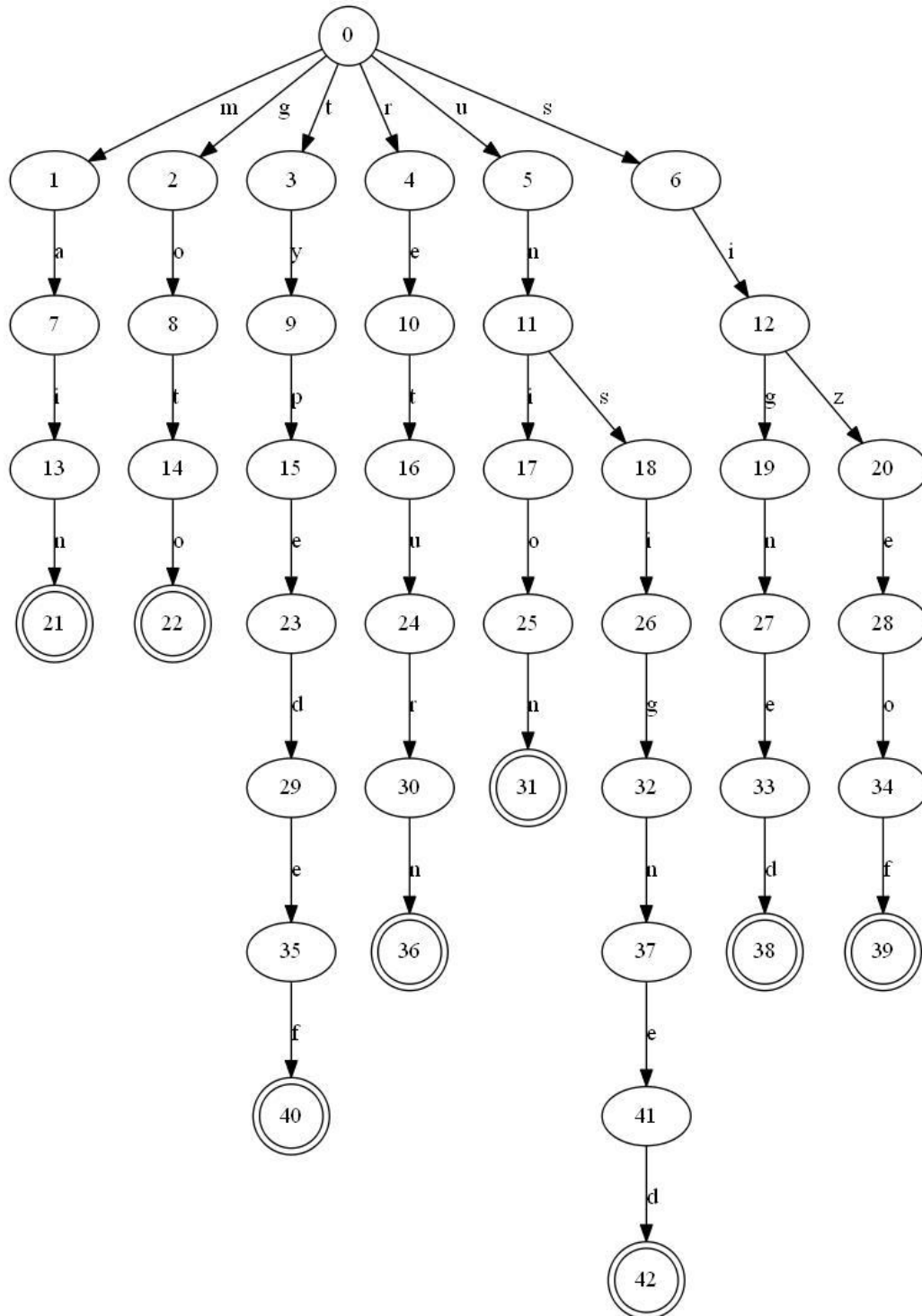
Autómata que reconoce palabras reservadas

Realiza el trabajo de reconocer algunas palabras reservadas de C, estas son fijas.

Expresión regular

(main)|(goto)|(return)|(signed)|(sizeof)|(typedef)|(union)|(unsigned)

Representación gráfica



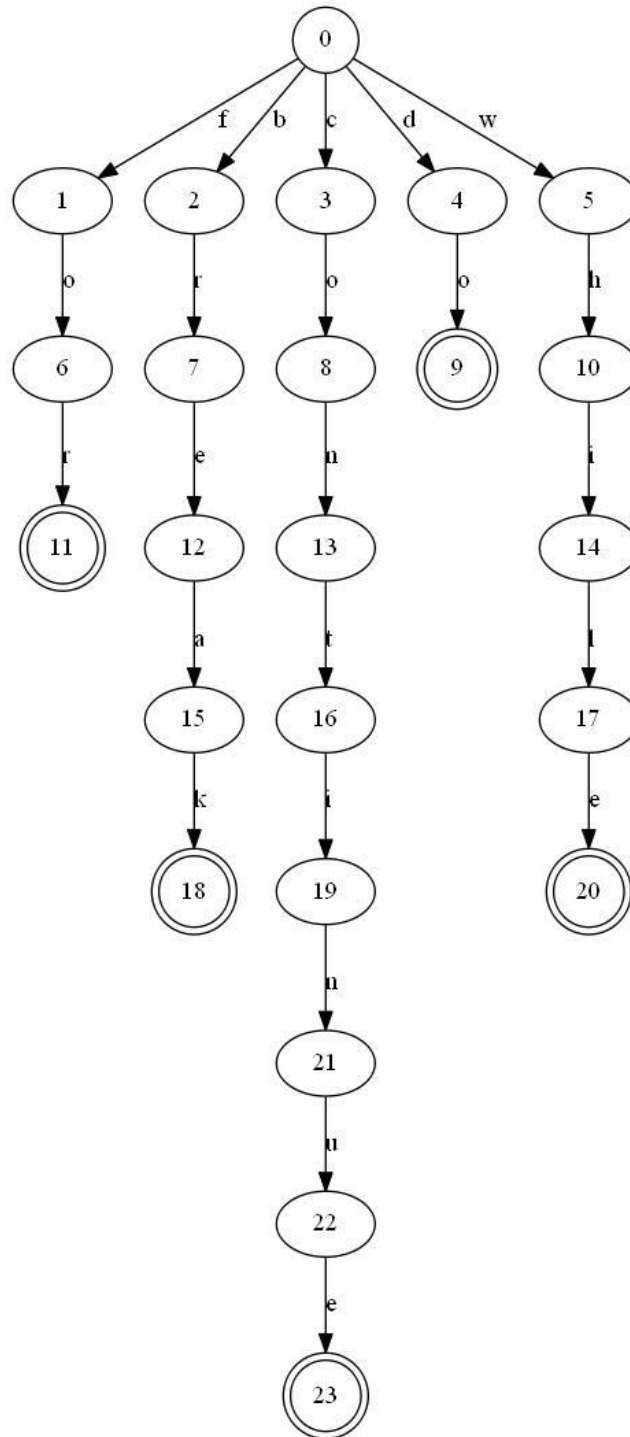
Autómata que reconoce estructuras repetitivas

Al ser estas palabras reservadas la expresión regular para validarlas es un poco “estática”

Expresión regular

`(for)|(break)|(continue)|(do)|(while)`

Representación gráfica



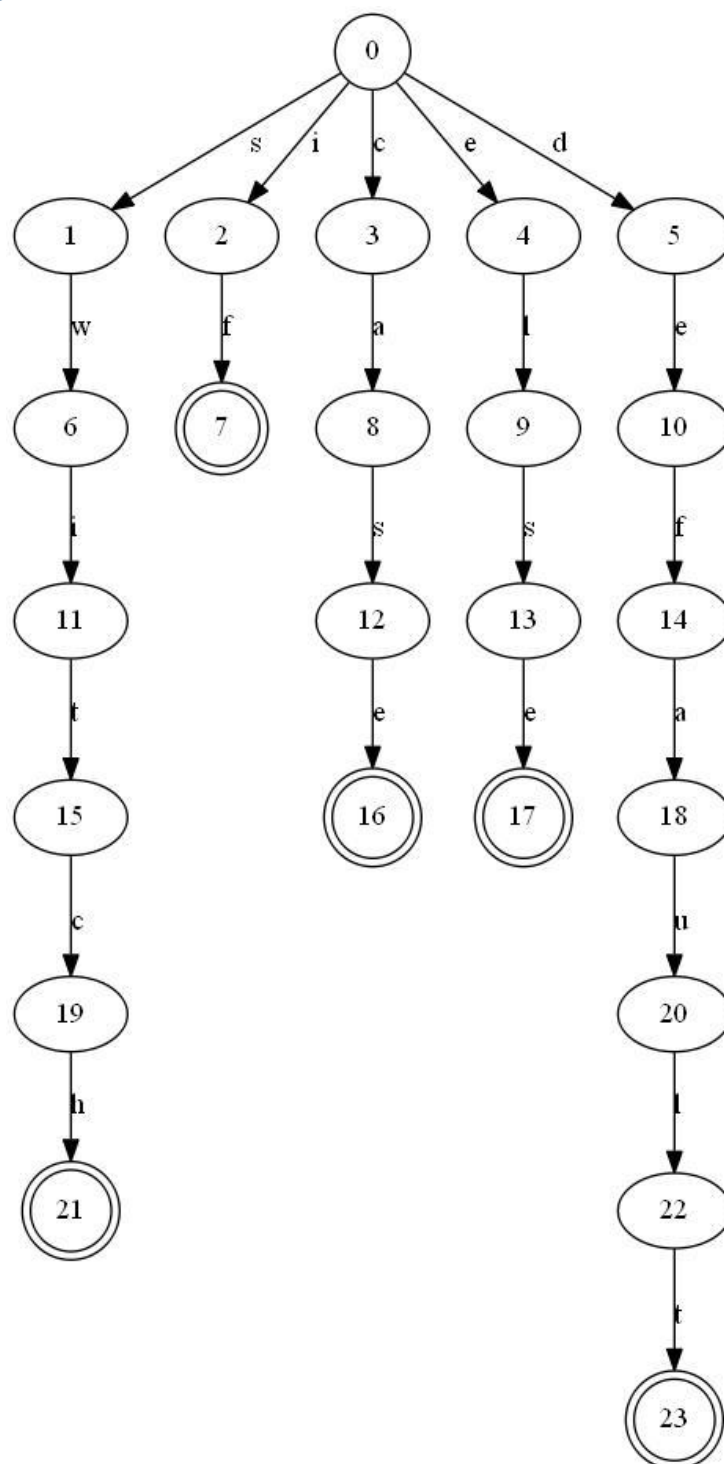
Autómata que reconoce estructuras selectivas

Estáticas, similares a las estructuras repetitivas

Expresión regular

(switch)|(case)|(else)|(if)|(default)

Representación gráfica

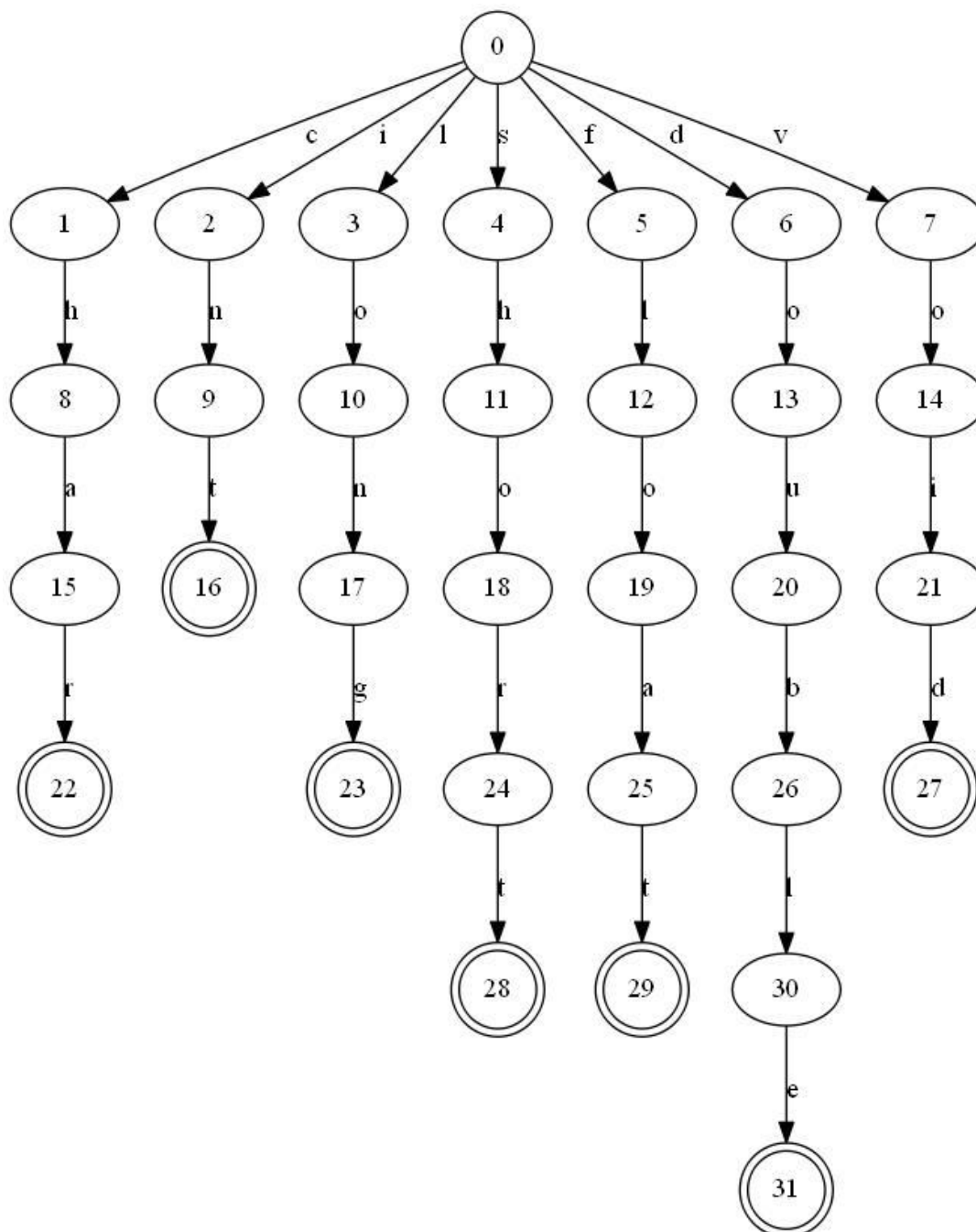


Autómata que reconoce tipos de dato

Expresión regular

(char)|(int)|(long)|(short)|(float)|(double)|(void)

Representación gráfica

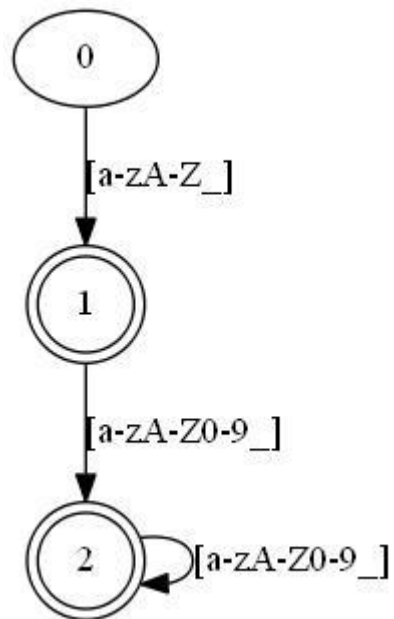


Autómata que reconoce los identificadores

Expresión regular

$[a-zA-Z_]+[a-zA-Z0-9_]^*$

Representación gráfica

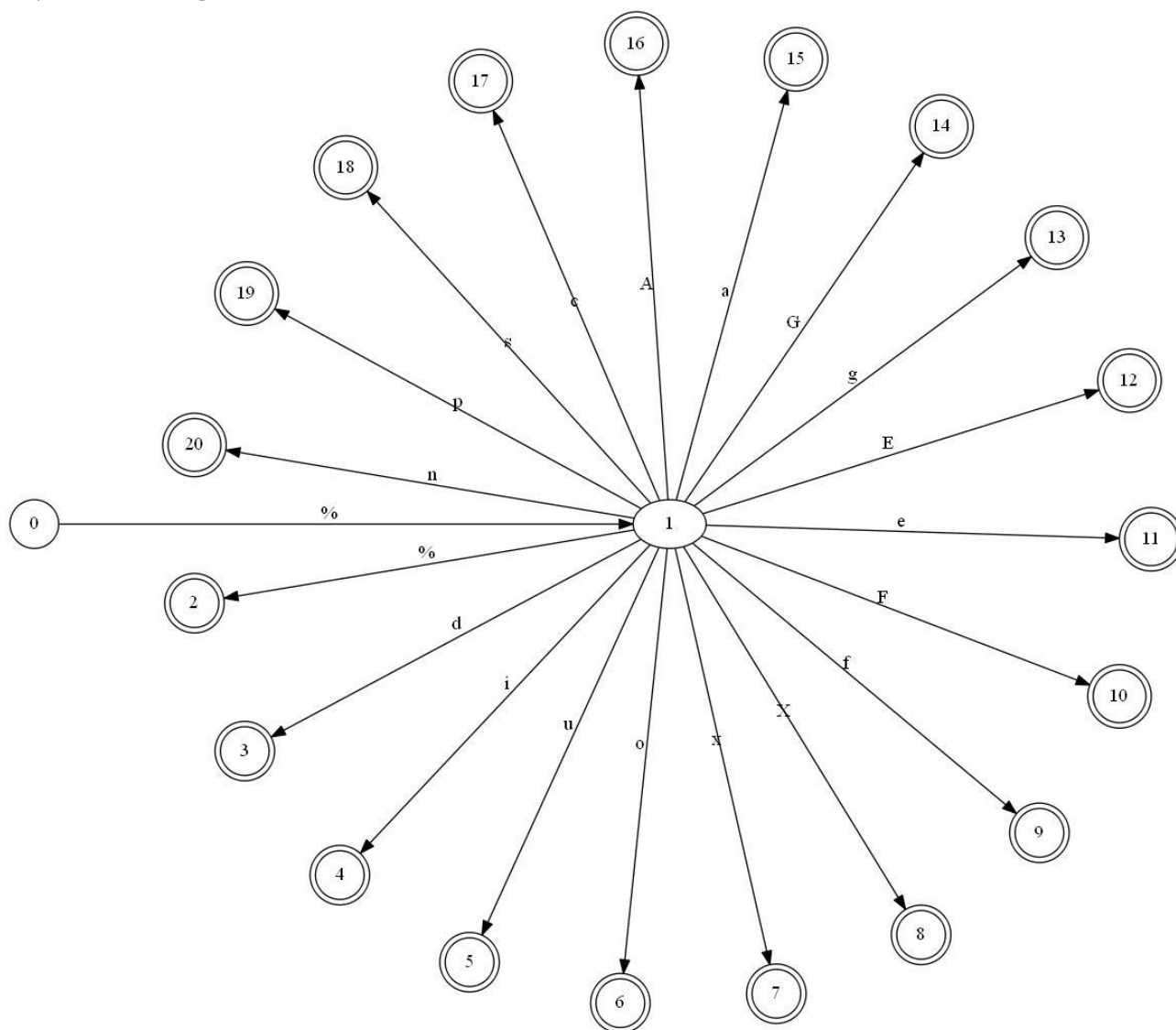


Autómata que reconoce especificadores de formato

Expresión regular

`%(d|i|u|o|x|X|f|F|e|E|g|G|a|A|c|s|p|n|%)`

Representación gráfica

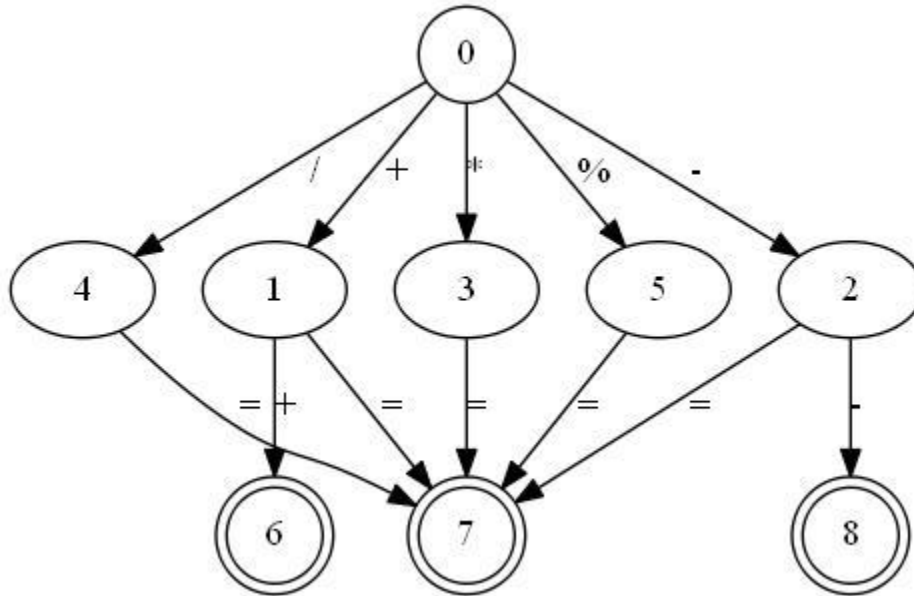


Autómata que reconoce operadores de incremento/decremento

Expresión regular

$(++)|(--)|(+|-|*|/|\%)=$

Representación gráfica

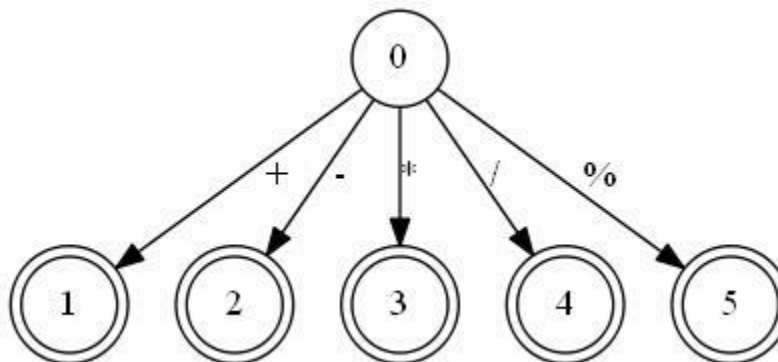


Autómata que reconoce operadores aritméticos

Expresión regular

$+|-|*|/|\%$

Representación gráfica

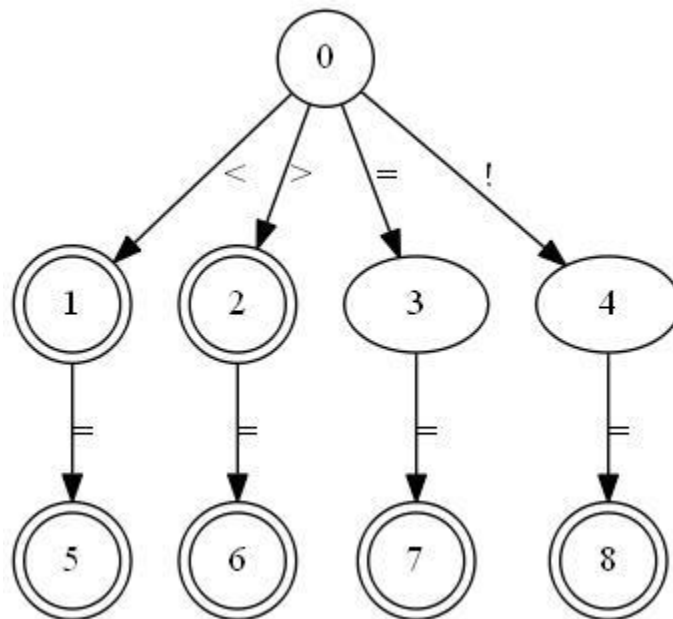


Autómata que reconoce operadores relacionales

Expresión regular

<|>|(>=)|(<=)|(!=)|(==)

Representación gráfica

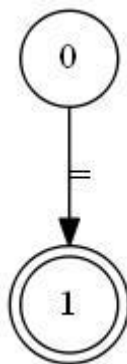


Autómata que reconoce el operador asignación

Expresión regular

=

Representación gráfica

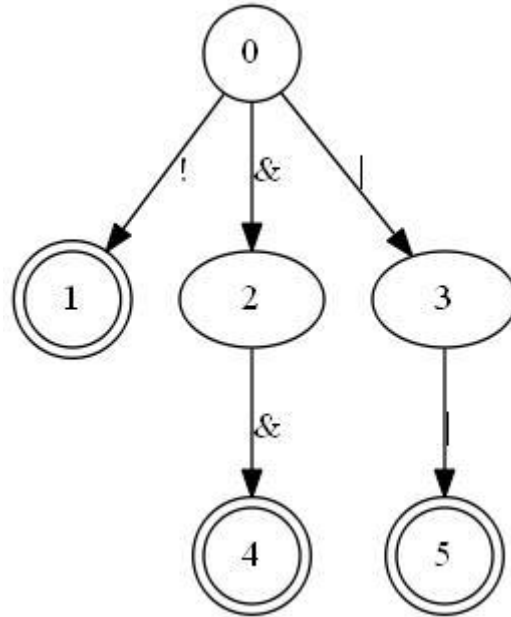


Autómata que reconoce los operadores lógicos

Expresión regular

!(&&)|(|)

Representación gráfica



Autómata que reconoce caracteres especiales

Expresión regular

{ } [] () ; , .

Representación gráfica

