

Introducción

Monday, January 15, 2024 2:33 PM

Hay dos partes importantes para crear un compilador

1. Análisis
2. Síntesis

El análisis consta de la parte léxica (donde se buscan los identificadores, los signos y números), la parte sintáctica (donde se identifican que se cumplan las reglas definidas) y el análisis semántico (donde se distinguen los tipos de datos y las operaciones). Finalmente se reduce el código y se traduce al lenguaje destino.

Los lenguajes de programación se dividen en cinco niveles

1. Primera generación: Son 1 y 0
2. Segunda generación: los lenguajes ensambladores
3. Tercera generación: lenguajes de alto nivel que se parecen más al lenguaje humano
4. Cuarta generación: Lenguajes de propósitos específicos como SQL
5. Quinta generación: lenguajes basados en lógica como Lisp

En el preprocesamiento se eliminan

- Comentarios
- Espacios en blanco
- Tabuladores
- Nuevas líneas
- Caracteres de separación

Token = (Unidad léxica, atributo)

Patrón = Descripción de un lexema

Lexema = Secuencia de caracteres del programa fuente

Buffer = Se utiliza para transmitir y analizar

Conceptos

Wednesday, January 17, 2024 3:09 PM

Alfabeto: Conjunto finito de símbolos

Cadena: Secuencia finita de símbolos de un alfabeto

- Prefijo: Cadena que se obtiene de eliminar uno o más elementos de la derecha de una cadena
- Sufijo: Cadena que se obtiene de eliminar uno o más elementos de la izquierda de una cadena
- Subcadena: Cadena que se obtiene de eliminar un prefijo y un sufijo
- Subsecuencia: Cadena obtenida de la eliminación de símbolos no necesariamente contiguos

Error Léxico: Error cuando no se cumplen las reglas del autómata o la expresión regular (ñ)

Error sintáctico: Errores en las palabras o reglas del código (Punto y coma. Ortografía)

Unión: $X \cup Y = Y \cup X$

Concatenación: $XY \neq YX$

$L = \{0,1\}$

L^* :

- $L^0 = \{\epsilon\}$
- $L^1 = \{0,1\}$
- $L^2 = \{00,01,10,11\}$

L^* = Cerradura de Kleen (while)

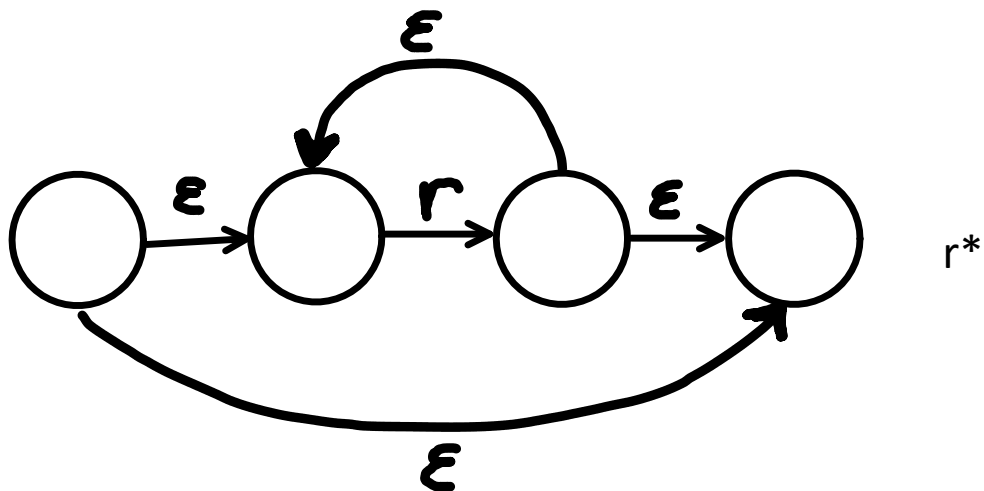
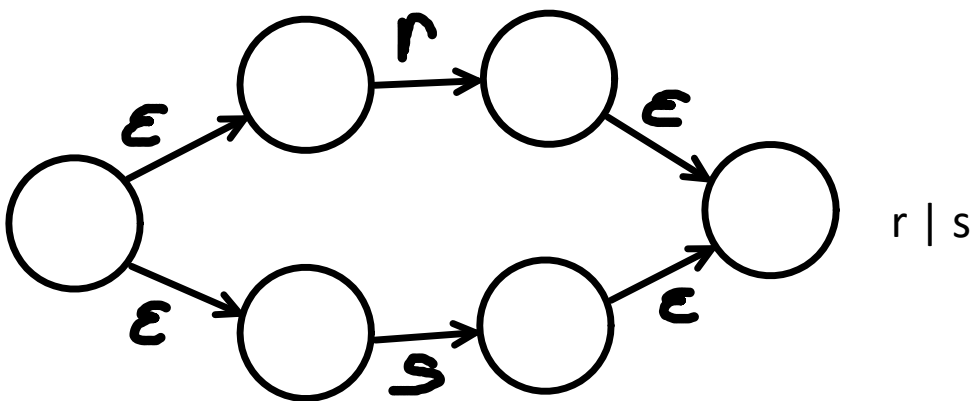
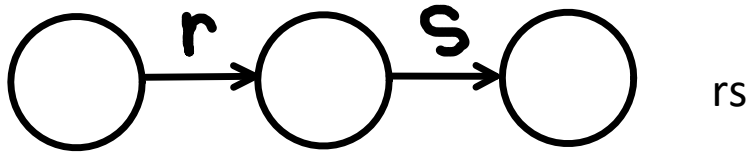
L^+ = Cerradura positiva (Do while)

Autómatas

Monday, January 29, 2024

3:20 PM

Reglas de Thomson

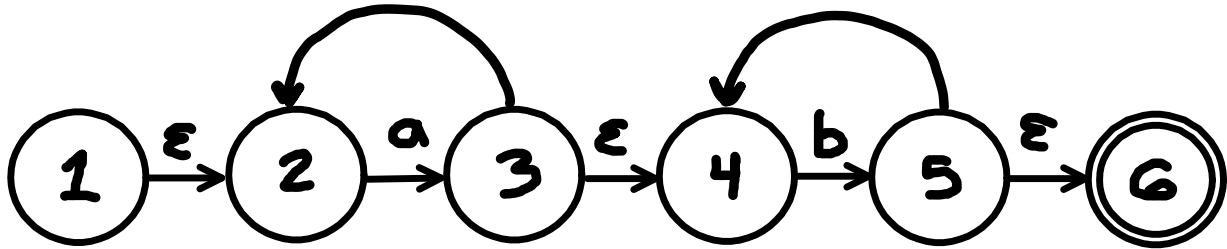


Representan una expresión regular.

Existen los AFN y AFD

Autómata finito no determinista (Con épsilon o sin épsilon) (AFN): Desde un nodo puede ir a distintos estados con la misma etiqueta

Autómata finito determinista (AFD): Una etiqueta solo te puede llevar a un estado.



$A = \{1, 2\}$

$(A, a) = \{2, 3, 4\} = B$

$(A, b) = \{\}$

$(B, a) = \{2, 3, 4\} = B$

$(B, b) = \{4, 5, 6\} = C$

$(C, a) = \{\}$

$(C, b) = \{4, 5, 6\} = C$

	a	b
A	B	-
B	B	C
C	-	C

Manejo de Errores

Tuesday, January 30, 2024 3:22 PM

Objetivos:

- Informar de errores con claridad y recuperarse del error para poder continuar.

Existen métodos de análisis de error sintácticos llamados LL y LR

Estrategias de recuperación:

1. Modo Pánico
 - Quita todos los errores hasta encontrar algo que reconoce como correcto. Parte desde ese punto de reconocimiento.
2. A nivel de frase
 - Corregir con alguna cadena que permita continuar
3. Producciones de error
 -
4. Corrección global
 - Si encuentras un error varias veces corregir todos a lo mismo

Gramáticas

Tuesday, January 30, 2024 3:33 PM

Las reglas de cada idioma indican la sintaxis.

La sintaxis se puede especificar usando gramáticas libres de contexto (notación BNF - Backus-Naur Form) las cuales son fácil de entender y eliminan ambigüedades.

Tener una gramática nos permite integrar fácilmente nuevas construcciones.

Existen tres tipos de analizadores sintácticos para gramáticas:

- Métodos universales
- Métodos descendentes (LL)
- Métodos ascendentes (LR)

Una gramática consiste en

- Terminales (nombre del token o simplemente token)
- No terminales o variables sintácticas
- Un símbolo inicial en la primera producción
- Un conjunto de producciones

Producción:

Factor = id

Factor es no terminal

Id es terminal

Terminales y no terminales

Wednesday, January 31, 2024 2:50 PM

Estos símbolos son terminales:

- Las primeras letras minúsculas del alfabeto como a, b, c
- Los símbolos de operadores como +, *, etc.
- Los símbolos de puntuación como paréntesis, coma, etc.
- Los dígitos del 0 al 9
- Las cadenas en negrita como **id** o **if**, cada una de las cuales representa un solo símbolo terminal.

Estos símbolos son no terminales:

- Las primeras letras mayúsculas del alfabeto como A, B, C.
- La letra S que, al aparecer es, por lo general, el símbolo inicial.
- Los nombres en cursiva y minúsculas, como *expr* o *instr*.
- Al hablar sobre las construcciones

E
E*E
-E*E
-E*(E)
-E*(E+E)
-id*(id+id)

E
E+E
(E)+E
(E)+(E)
(E-E)+(E)
(E-E)+(E*E)
(E-E)+(E*-E)
(id-id)+(id*-id)