

---

# **Compiladores:**

## **Sesión 4, Análisis Léxico**

---

**Pontificia Universidad Javeriana Cali**  
**Ingeniería de Sistemas y Computación**  
**Prof. Gloria Inés Alvarez V.**

# Definiciones regulares

- Facilitan la escritura y la interpretación de expresiones regulares largas o complejas.
- Es una notación que permite dar nombre a una expresión regular para usarla en otra.

Ejemplo.

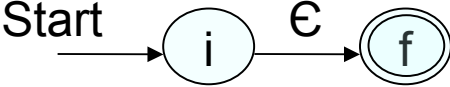
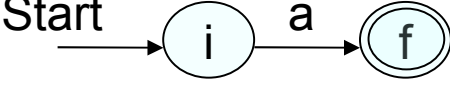
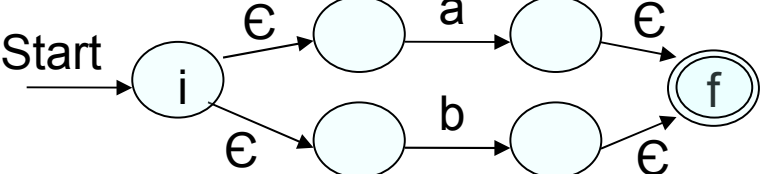
Mayúsculas -> A | B | C | .....| Z

Minúsculas -> a | b | c | ...| z

Nombre -> Mayúsculas (Minúsculas)\*

# Algoritmo de Construcción de Thompson

- A partir de una expresión regular es posible e manera automática construir un NFA:

Exp.Regular	NFA
$\epsilon$	 <p>Start → i → <math>\epsilon</math> → f</p>
a (a esta en $\Sigma$ )	 <p>Start → i → a → f</p>
a b	 <p>Start → i → <math>\epsilon</math> → (a b) → <math>\epsilon</math> → f</p>

# Algoritmo de Construcción de Thompson

Exp.Regular	NFA
ab	<p>The NFA for the regular expression 'ab' consists of three states: a start state 'i', an intermediate state, and a final state 'f'. The start state 'i' is indicated by an arrow labeled 'Start'. There is a transition from 'i' to the intermediate state labeled 'a', and a transition from the intermediate state to 'f' labeled 'b'. The final state 'f' is indicated by a double circle.</p>
$r^*$	<p>The NFA for the regular expression <math>r^*</math> consists of four states: a start state 'i', two intermediate states, and a final state 'f'. The start state 'i' is indicated by an arrow labeled 'Start'. There is a transition from 'i' to the first intermediate state labeled <math>\epsilon</math>. There is a transition from the first intermediate state to the second intermediate state labeled 'r'. There is a transition from the second intermediate state back to the first intermediate state labeled <math>\epsilon</math>. There is a transition from the second intermediate state to the final state 'f' labeled <math>\epsilon</math>. The final state 'f' is indicated by a double circle.</p>

# Simulación en dos pilas de un NFA

Dada una cadena de entrada  $X$ , y un NFA  $N$ , determinar si  $N$  acepta  $X$ :

$S = \epsilon\text{-closure}(\{S_0\})$

$a = \text{nextchar}()$

While  $a \neq \text{EOF}$

$S = \epsilon\text{-closure}(\text{move}(S, a))$

$a = \text{nextchar}()$

If  $S \cap F \neq \text{vacío}$

    return "Yes"

Else return "No"

Se implementa usando 2 pilas:

Una para el conjunto actual de estados (Move)

Otra para el conjunto de nuevos estados (Clausura)

# Algoritmo de Construcción de Subconjuntos

Dado un NFA construir un DFA:

- Operaciones:
  - $\epsilon$ -Closure( $S_i$ ): conjunto de estados del NFA alcanzables desde el estado  $S_i$ , con transiciones  $\epsilon$
  - $\epsilon$ -Closure( $T$ ), donde  $T$  es un conjunto de estados del NFA: conjunto de estados del NFA alcanzables desde el estado  $S$  en  $T$ , con transiciones  $\epsilon$
  - Move( $T, a$ ), donde  $T$  es un conjunto de estados del NFA, y  $a$  es un símbolo del alfabeto: conjunto de estados del NFA a los cuales hay una transición con el símbolo  $a$ , desde un estado  $S$  en  $T$ .

# Algoritmo de Construcción de Subconjuntos

Algoritmo: genera la tabla de transición Dtran

Dstates =  $\epsilon$ -closure( $S_0$ )

Mientras haya un estado T no marcado en Dstates:

    Marcar T;

    Para cada símbolo de entrada a

$U = \epsilon$ -closure(Move(T,a))

        Si U no está en Dstates entonces

            Adicionar U a Dstates sin marcar

        Dtran[T,a] = U

# Consideraciones de Tiempo y Espacio

Dada una expresión regular  $r$  y una cadena de entrada  $x$ :

- NFA:
  - Algoritmo de construcción de Thompson:
    - Tiempo  $O(|r|)$
    - Tabla de transición  $O(|r|)$(NFA tiene a lo sumo  $2|r|$  estados y 2 transiciones por cada estado)
  - Simular el NFA con el algoritmo de 2 pilas:  $O(|r|^*|x|)$



# Consideraciones de Tiempo y Espacio

Dada una expresión regular  $r$  y una cadena de entrada  $x$ :

- DFA:
  - Algoritmo de construcción de Thompson:
    - Tiempo  $O(|r|)$
  - Algoritmo de Construcción de subconjuntos:
    - Tiempo
    - Tabla de transición  $O(2^{|r|})$
  - Simular el DFA:  $O(|x|)$