.UBAfiuba
FACULTAD DE INGENIERÍA

Lenguajes regulares

Teoría de Algoritmos I (75.29 / 95.06)
Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Lenguajes regulares

Un lenguaje es regular

si existe algún autómata finito que lo reconozca.

Existen algunas operaciones que se pueden aplicar a los lenguajes regulares llamados operaciones regulares

Cuyo resultado es también un lenguaje regular

Operaciones regulares

Sean A, B dos lenguajes regulares,

Union: $A \cup B = \{x / x \in A \text{ o } x \in B\}$

Concatenación: $A \circ B = \{xy / x \in A, y \in B\}$

Estrella (star): $A^* = \{x_1 x_2 \ldots x_k / k \geq 0 \text{ y cada } x_i \in A\}$

Ejemplo:

A={auto, avion} y B={rojo, verde, azul}

$A \cup B$ = {auto, avion,rojo, verde, azul}

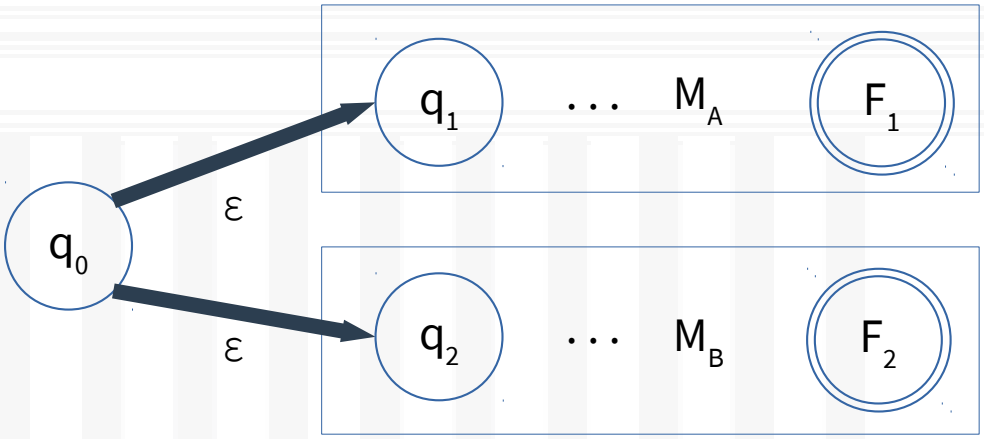
$A \circ B$ = {autorojo, autoverde, autoazul, avionrojo, avionverde, avionazul}

A^* = {ε, auto, avion, autoauto, avionavion, autoavion, avionauto, ...}

Union

Sean A,B automatas finitos

Queremos ejecutar simultáneamente la maquina A y B y ver si alguna de ellas termina en estado de aceptación



Creamos un nuevo autómata

Que incluya A y B

Con un estado inicial y una transición sin lectura al estado inicial de A y B

Concatenación

Sean A, B autómatas finitos

Queremos que una se ejecute inmediatamente luego de la otra para ver si el resultado termina en estado de aceptación

Creamos un nuevo autómata

Que incluya A y B

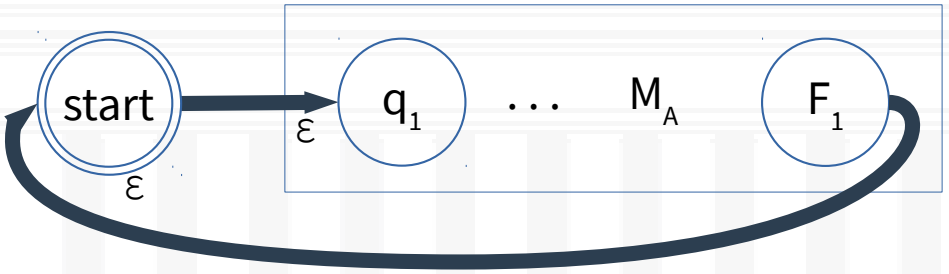
Se agrega transición con símbolo ε entre los estados de aceptación de A y el estado de inicio de B

Los estados de aceptación de A dejan de serlo.

Estrella

Sea A un autómata finito

Queremos se ejecute nuevamente inmediatamente al llegar al estado de aceptación



Creamos un nuevo autómata

Con un nuevo estado inicial y al mismo tiempo de aceptación

Se agrega transición con símbolo ε entre el nuevo estado y el estado de aceptación de A

Se agrega transición con símbolo ε entre los estados de finalización de A y el nuevo estado

Los estados de aceptación de A dejan de serlo

Expresiones regulares

Se conoce como “expresiones regulares”

al uso de operaciones regulares para construir expresiones describiendo lenguajes

Cualquier expresión regular se puede resolver con un AFD (o AFND)

Por lo tanto son equivalentes en su poder de expresión (y reconocimiento)

Expresiones regulares: Definición formal

R es una expresion regular si es:

1. a para algún a en el alfabeto Σ
2. ε (string vacio)
3. Ø (ausencia de string)
4. (R1 ∪ R1)
5. (R1 o R2)
6. (R1)*

(Con R1 y R2 expresiones regulares)

Lenguajes NO regulares

Lenguajes que un AF no puede reconocer.

Ej: $B = \{0^n 1^n / n \geq 0\}$

El AF carece de memoria

Operaciones que requieren “recordar” algo (salvo particularidades) no son posibles.

Un lenguaje tiene que cumplir una propiedad para ser regular

Utilizaremos un teorema conocido “pumping lemma” para verificarlo

Pumping lemma (Lema del “bombeo”)

Si un lenguaje no tiene la propiedad

Entonces no es regular

Todos los strings de un lenguaje regular

pueden "bombearse" si son al menos tan largos que un cierto valor especial, llamado longitud de bombeo

Cada string contiene una sección

que puede repetirse cualquier cantidad de veces con el string resultante aun perteneciendo al lenguaje

Pumping lemma: Definición

Si A es un lenguaje regular,

Existe un numero p (longitud de bombeo),

donde

si s es cualquier string $\in$ A de longitud al menos p,

Entonces s puede ser divido en

3 partes s=xyz

Que satisfacen las condiciones:

1)Para cada $i \geq 0$, $xy^iz \in A$,

2)$|y| > 0$

3)$|xy| \leq p$

.UBAfiuba

FACULTAD DE INGENIERÍA

Presentación realizada en Julio de 2020