Teoría de la Computación

Sesión 5

Edgar Andrade, Ph.D.

Matemáticas Aplicadas y Ciencias de la computación

Última revisión: Agosto de 2020

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Expresiones regulares

Equivalencia entre expresiones regulares y lenguajes regulares

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Contenido

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Expresiones regulares

Definición recursiva

Las expresiones regulares están formadas por los siguientes bloques:

1. a para cada símbolo del alfabeto Σ
2. ϵ
3. $\emptyset$

Expresiones regulares

Definición recursiva

Las expresiones regulares están formadas por los siguientes bloques:

1. a para cada símbolo del alfabeto Σ
2. ϵ
3. $\emptyset$

Y por las siguientes reglas. Si R_1 y R_2 son expresiones regulares, entonces también lo son:

1. $(R_1 \cup R_2)$
2. $(R_1 R_2)$
3. R_1^*

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Lenguaje de una expresión regular

Sea R una expresión regular. Decimos que $L(R)$ es el lenguaje correspondiente a R, el cual se define de la siguiente manera:

Definición

1. $L(a) = \{a\}$, donde $a \in \Sigma$
2. $L(\epsilon) = \{\epsilon\}$
3. $L(\emptyset) = \emptyset$
4. $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$
5. $L(R_1 R_2) = L(R_1)L(R_2)$
6. $L(R_1^*) = L(R_1)^*$

Ejemplo

Vamos a calcular la expresión regular $R = (0 \cup 1)0^*$.

Ejemplo

Vamos a calcular la expresión regular $R = (0 \cup 1)0^*$.

$$L((0 \cup 1)0^*) = L(0 \cup 1)L(0^*)$$

Ejemplo

Vamos a calcular la expresión regular $R = (0 \cup 1)0^*$.

$$\begin{aligned} L((0 \cup 1)0^*) &= L(0 \cup 1)L(0^*) \\ &= (L(0) \cup L(1))L(0^*) \end{aligned}$$

Ejemplo

Vamos a calcular la expresión regular $R = (0 \cup 1)0^*$.

$$\begin{aligned} L((0 \cup 1)0^*) &= L(0 \cup 1)L(0^*) \\ &= (L(0) \cup L(1))L(0^*) \\ &= (L(0) \cup L(1))L(0)^* \end{aligned}$$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Ejemplo

Vamos a calcular la expresión regular $R = (0 \cup 1)0^*$.

$$\begin{aligned}
L((0 \cup 1)0^*) &= L(0 \cup 1)L(0^*) \\
&= (L(0) \cup L(1))L(0^*) \\
&= (L(0) \cup L(1))L(0)^* \\
&= (\{0\} \cup \{1\})\{0\}^* = \{0, 1\}\{0\}^*
\end{aligned}$$

Ejemplo

Vamos a calcular la expresión regular $R = (0 \cup 1)0^*$.

$$\begin{aligned} L((0 \cup 1)0^*) &= L(0 \cup 1)L(0^*) \\ &= (L(0) \cup L(1))L(0^*) \\ &= (L(0) \cup L(1))L(0)^* \\ &= (\{0\} \cup \{1\})\{0\}^* = \{0, 1\}\{0\}^* \end{aligned}$$

Entonces el lenguaje que describe la expresión regular es:

$$L(R) = \{00^k, 10^k, k \geq 0\}.$$

Convenciones

Convenimos en que $R^+ = RR^*$. Observe que R^+ contiene las cadenas que están formadas por *al menos una* cadena de R. Con esta convención tenemos $R^+ \cup \epsilon = R^*$.

Convenciones

Convenimos en que $R^+ = RR^*$. Observe que R^+ contiene las cadenas que están formadas por *al menos una* cadena de R. Con esta convención tenemos $R^+ \cup \epsilon = R^*$.

También convenimos en que $R^k = \underbrace{RR \ldots R}_{k \text{ veces}}$.

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único1}\}$;

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único1}\}$;
- $(10^+)^* =$

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único1}\}$;
- $(10^+)^* = \{w, w \text{ cada 1 está seguido de por lo menos un 0}\}$;

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único1}\}$;
- $(10^+)^* = \{w, w \text{ cada 1 está seguido de por lo menos un 0}\}$;
- $(0 \cup \epsilon)(1 \cup \epsilon) =$

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único1}\}$;
- $(10^+)^* = \{w, w \text{ cada 1 está seguido de por lo menos un 0}\}$;
- $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$;

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único} 1\}$;
- $(10^+)^* = \{w, w \text{ cada } 1 \text{ está seguido de por lo menos un } 0\}$;
- $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$;
- $1^*\emptyset =$

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único1}\}$;
- $(10^+)^* = \{w, w \text{ cada 1 está seguido de por lo menos un 0}\}$;
- $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$;
- $1^*\emptyset = \emptyset$;

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único1}\}$;
- $(10^+)^* = \{w, w \text{ cada 1 está seguido de por lo menos un 0}\}$;
- $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$;
- $1^*\emptyset = \emptyset$;
- $\emptyset^* =$

Más ejemplos

- $0^*10^* = \{w, w$ tiene un único1$\}$;
- $(10^+)^* = \{w, w$ cada 1 está seguido de por lo menos un 0$\}$;
- $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$;
- $1^*\emptyset = \emptyset$;
- $\emptyset^* = \{\epsilon\}$.

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único1}\}$;
- $(10^+)^* = \{w, w \text{ cada 1 está seguido de por lo menos un } 0\}$;
- $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$;
- $1^*\emptyset = \emptyset$;
- $\emptyset^* = \{\epsilon\}$.
- $\Sigma^*010\Sigma^* =$

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Más ejemplos

- $0^*10^* = \{w, w \text{ tiene un único}1\}$;
- $(10^+)^* = \{w, w \text{ cada 1 está seguido de por lo menos un } 0\}$;
- $(0 \cup \epsilon)(1 \cup \epsilon) = \{\epsilon, 0, 1, 01\}$;
- $1^*\emptyset = \emptyset$;
- $\emptyset^* = \{\epsilon\}$.
- $\Sigma^*010\Sigma^* = \{w, w \text{ contiene la subpalabra } 010\}$;

Diferencia entre ϵ y $\emptyset$

Sea R una expresión regular. ¿Cuáles de las siguientes proposiciones son verdaderas y cuales falsas?:

- $R \cup \emptyset = R$

Diferencia entre ϵ y $\emptyset$

Sea R una expresión regular. ¿Cuáles de las siguientes proposiciones son verdaderas y cuales falsas?:

- $R \cup \emptyset = R$ ☞ Verdadero

Diferencia entre ϵ y $\emptyset$

Sea R una expresión regular. ¿Cuáles de las siguientes proposiciones son verdaderas y cuales falsas?:

- $R \cup \emptyset = R$ ☞ Verdadero
- $R\epsilon = R$

Diferencia entre ϵ y $\emptyset$

Sea R una expresión regular. ¿Cuáles de las siguientes proposiciones son verdaderas y cuales falsas?:

- $R \cup \emptyset = R$ ☞ Verdadero
- $R\epsilon = R$ ☞ Verdadero

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Diferencia entre ϵ y $\emptyset$

Sea R una expresión regular. ¿Cuáles de las siguientes proposiciones son verdaderas y cuales falsas?:

- $R \cup \emptyset = R$ ☞ Verdadero
- $R\epsilon = R$ ☞ Verdadero
- $R \cup \epsilon = R$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Diferencia entre ϵ y $\emptyset$

Sea R una expresión regular. ¿Cuáles de las siguientes proposiciones son verdaderas y cuales falsas?:

- $R \cup \emptyset = R$ ☞ Verdadero
- $R\epsilon = R$ ☞ Verdadero
- $R \cup \epsilon = R$ ☞ Falso

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Diferencia entre ϵ y $\emptyset$

Sea R una expresión regular. ¿Cuáles de las siguientes proposiciones son verdaderas y cuales falsas?:

- $R \cup \emptyset = R$ ☞ Verdadero
- $R\epsilon = R$ ☞ Verdadero
- $R \cup \epsilon = R$ ☞ Falso
- $R\emptyset = R$

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Diferencia entre ϵ y $\emptyset$

Sea R una expresión regular. ¿Cuáles de las siguientes proposiciones son verdaderas y cuales falsas?:

- $R \cup \emptyset = R$ ☞ Verdadero
- $R\epsilon = R$ ☞ Verdadero
- $R \cup \epsilon = R$ ☞ Falso
- $R\emptyset = R$ ☞ Falso

Aplicación a los compiladores

Las expresiones regulares sirven para definir la estructura aceptable de las expresiones usadas en nombres y valores de las variables y constantes de los lenguajes de programación.

Aplicación a los compiladores

Las expresiones regulares sirven para definir la estructura aceptable de las expresiones usadas en nombres y valores de las variables y constantes de los lenguajes de programación.

Por ejemplo, supongamos $\Sigma = \{+, -\} \cup D$, donde $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Las expresiones numéricas pueden restringirse mediante la siguiente expresión regular:

$$(+ \cup - \cup \epsilon)(D^{+} \cup D^{+}.D^{*} \cup D^{*}.D^{+})$$

Aplicación a los compiladores

Las expresiones regulares sirven para definir la estructura aceptable de las expresiones usadas en nombres y valores de las variables y constantes de los lenguajes de programación.

Por ejemplo, supongamos $\Sigma = \{+, -\} \cup D$, donde $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Las expresiones numéricas pueden restringirse mediante la siguiente expresión regular:

$$(+ \cup - \cup \epsilon)(D^{+} \cup D^{+}.D^{*} \cup D^{*}.D^{+})$$

Ejemplos

72

Aplicación a los compiladores

Las expresiones regulares sirven para definir la estructura aceptable de las expresiones usadas en nombres y valores de las variables y constantes de los lenguajes de programación.

Por ejemplo, supongamos $\Sigma = \{+, -\} \cup D$, donde $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Las expresiones numéricas pueden restringirse mediante la siguiente expresión regular:

$$(+ \cup - \cup \epsilon)(D^+ \cup D^+.D^* \cup D^*.D^+)$$

Ejemplos

72 , 3.14159

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Aplicación a los compiladores

Las expresiones regulares sirven para definir la estructura aceptable de las expresiones usadas en nombres y valores de las variables y constantes de los lenguajes de programación.

Por ejemplo, supongamos $\Sigma = \{+, -\} \cup D$, donde $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Las expresiones numéricas pueden restringirse mediante la siguiente expresión regular:

$$(+ \cup - \cup \epsilon)(D^{+} \cup D^{+}.D^{*} \cup D^{*}.D^{+})$$

Ejemplos

72 , 3.14159 , +7.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Aplicación a los compiladores

Las expresiones regulares sirven para definir la estructura aceptable de las expresiones usadas en nombres y valores de las variables y constantes de los lenguajes de programación.

Por ejemplo, supongamos $\Sigma = \{+, -\} \cup D$, donde $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Las expresiones numéricas pueden restringirse mediante la siguiente expresión regular:

$$(+ \cup - \cup \epsilon)(D^+ \cup D^+.D^* \cup D^*.D^+)$$

Ejemplos

72 , 3.14159 , +7. , -.01

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Regex de Python

```
import re

#Check if the string contains "ai"

txt = "The rain in Spain"
x = re.search(".*ai.*", txt)

if x:
  print("YES! We have a match!")
else:
  print("No match")
```

Metacaracteres

https://www.w3schools.com/python/python_regex.asp

Contenido

MACC
Matemáticas Aplicadas y
Ciencias de la Computación

El teorema de Kleene

Teorema

Un lenguaje A es regular si y solo si hay una expresión regular R tal que $L(R) = A$.

MACC
Matemáticas Aplicadas y Ciencias de la Computación

El teorema de Kleene

Teorema

Un lenguaje A es regular si y solo si hay una expresión regular R tal que $L(R) = A$.

Lema (esta sesión)

Si R es una expresión regular, entonces $L(R)$ es regular.

El teorema de Kleene

Teorema

Un lenguaje A es regular si y solo si hay una expresión regular R tal que $L(R) = A$.

Lema (esta sesión)

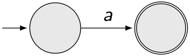
Si R es una expresión regular, entonces $L(R)$ es regular.

Lema (la próxima sesión)

Si A es un lenguaje regular, entonces existe R una expresión regular tal que $L(R) = A$.

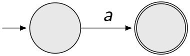
Demostración

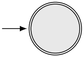
Vamos por casos:

- $R = a$ para algún $a \in \Sigma$. Construimos: 

Demostración

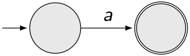
Vamos por casos:

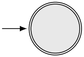
- $R = a$ para algún $a \in \Sigma$. Construimos: 

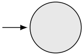
- $R = \epsilon$. Construimos: 

Demostración

Vamos por casos:

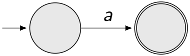
- $R = a$ para algún $a \in \Sigma$. Construimos: 

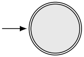
- $R = \epsilon$. Construimos: 

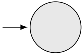
- $R = \emptyset$. Construimos: 

Demostración

Vamos por casos:

- $R = a$ para algún $a \in \Sigma$. Construimos: 

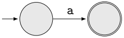
- $R = \epsilon$. Construimos: 

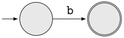
- $R = \emptyset$. Construimos: 

- Si $R = R_1 \cup R_2$, $R = R_1 \circ R_2$, o $R = R_1^*$ podemos utilizar las construcciones que hemos visto en la clase pasada. $\square$

Ejemplos

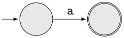
Vamos s construir un $\texttt{NFA}$ que reconozca el lenguaje de $(\texttt{ab} \cup \texttt{a})^*$:

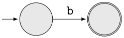
a: 

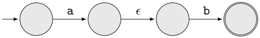
b: 

Ejemplos

Vamos s construir un `NFA` que reconozca el lenguaje de $(\mathtt{ab} \cup \mathtt{a})^*$:

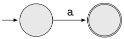
a: 

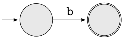
b: 

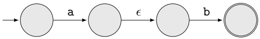
ab: 

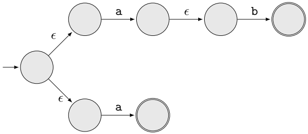
Ejemplos

Vamos s construir un $\texttt{NFA}$ que reconozca el lenguaje de $(\texttt{ab} \cup \texttt{a})^*$:

$\texttt{a}$: 

$\texttt{b}$: 

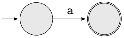
$\texttt{ab}$: 

$\texttt{ab} \cup \texttt{a}$: 

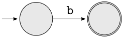
Ejemplos

Vamos s construir un `NFA` que reconozca el lenguaje de $(\mathtt{ab} \cup \mathtt{a})^*$:

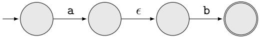
$\mathtt{a}$:



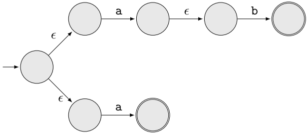
$\mathtt{b}$:



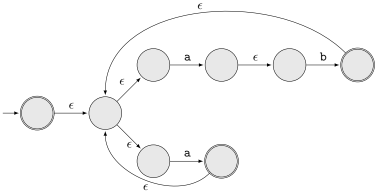
$\mathtt{ab}$:



$\mathtt{ab} \cup \mathtt{a}$:



$(\mathtt{ab} \cup \mathtt{a})^*$:

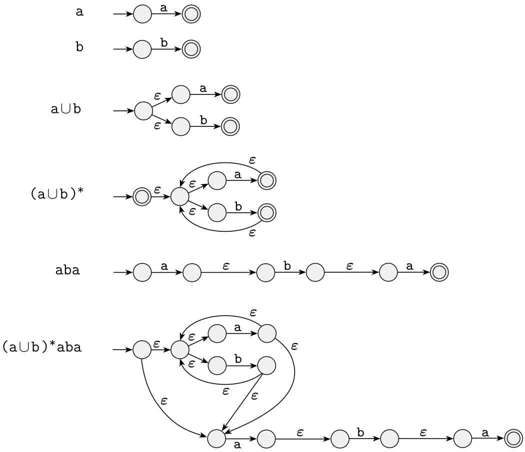


Ejemplos

El siguiente es un $\texttt{NFA}$ que reconoce $(\texttt{a} \cup \texttt{b})^*\texttt{aba}$:

Ejemplos

El siguiente es un NFA que reconoce $(a \cup b)^*aba$:



Take away

En esta sesión usted aprendió:

- Expresiones regulares y sus correspondientes lenguajes.
- Cómo crear un NFA que reconozca el lenguaje correspondiente a una expresión regular.