Teorema de Kleene I

Clase 07

IIC2223 / IIC2224

Prof. Cristian Riveros

Outline

NFA versus ϵ-NFA (clase anterior)

Desde Expresiones a Autómatas

Outline

NFA versus ϵ-NFA (clase anterior)

Desde Expresiones a Autómatas

Autómata finito no-determinista con ϵ-transiciones

Definición

Un autómata finito no-determinista con **ϵ-transiciones** (ϵ-NFA) es:

$$\mathcal{A} \quad = \quad (Q, \Sigma, \Delta, I, F)$$

- Q es un conjunto finito de estados.
- Σ es el alfabeto de input.
- $I \subseteq Q$ es un conjunto de estados iniciales.
- $F \subseteq Q$ es el conjunto de estados finales (o aceptación).

$$+$$

- $\Delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ es la relación de transición.

Ejecución de ϵ-NFA

Para ϵ-NFA veremos una **forma alternativa**
para definir las nociones de ejecución y aceptación.

Sea $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ un ϵ-NFA.

Definiciones

- Un par $(q, w) \in Q \times \Sigma^*$ es una **configuración** de $\mathcal{A}$.
- Una configuración (q, w) es **inicial** si $q \in I$.
- Una configuración (q, w) es **final** si $q \in F$ y $w = \epsilon$.

"Intuitivamente, una configuración (q, aw) representa que $\mathcal{A}$ se encuentra en el estado q procesando la palabra aw y leyendo a."

Ejecución de ϵ-NFA

Sea $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ un ϵ-NFA.

Definición

Se define la relación $\vdash_{\mathcal{A}}$ de **siguiente-paso** entre configuraciones de $\mathcal{A}$:

$$(p, u) \ \vdash_{\mathcal{A}} \ (q, v)$$

si, y solo si, existe $(p, c, q) \in \Delta$ con $c \in \Sigma \cup \{\epsilon\}$ tal que $u = c \cdot v$.

Notar que $\vdash_{\mathcal{A}} \subseteq \ (Q \times \Sigma^*) \times (Q \times \Sigma^*)$.

Se define $\vdash_{\mathcal{A}}^*$ como la clausura **refleja** y **transitiva** de $\vdash_{\mathcal{A}}$:

para toda configuración (q, w) :	$(q, w) \ \vdash_{\mathcal{A}}^* \ (q, w)$
si $(p, u) \vdash_{\mathcal{A}} (p', w)$ y $(p', w) \vdash_{\mathcal{A}}^* (q, v)$:	$(p, u) \ \vdash_{\mathcal{A}}^* \ (q, v)$

$(p, u) \ \vdash_{\mathcal{A}}^* \ (q, v)$ si uno puede ir de (p, u) a (q, v) en **0 o más pasos**.

Lenguaje aceptado por un ϵ-NFA

Sea $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ un ϵ-NFA y $w \in \Sigma^*$.

Definiciones

- $\mathcal{A}$ **acepta** w si existe una configuración **inicial** (q_0, w) y una configuración **final** (q_f, ϵ) tal que:

$$(q_0, w) \ \vdash^*_{\mathcal{A}} \ (q_f, \epsilon)$$

- El **lenguaje aceptado** por $\mathcal{A}$ se define como:

$$\mathcal{L}(\mathcal{A}) \ = \ \{w \in \Sigma^* \mid \mathcal{A} \text{ acepta } w\}$$

OJO: si $\mathcal{A}$ no tiene ϵ-transiciones, esta es una forma **alternativa** para definir ejecución para NFA y DFA.

Equivalencia entre NFA y ϵ-NFA

Teorema

Para todo autómata finito no-determinista con ϵ-transiciones $\mathcal{A}$, existe un autómata no-determinista $\mathcal{A}'$ tal que:

$$\mathcal{L}(\mathcal{A}) \;=\; \mathcal{L}(\mathcal{A}')$$

En otras palabras, NFA $\equiv$ ϵ-NFA.

¿cómo simulamos las ϵ-transiciones, esto es, sin leer el input?

Demostración

Para demostrar este teorema, mostraremos como construir un autómata no-determinista a partir de un ϵ-NFA removiendo las ϵ-transiciones.

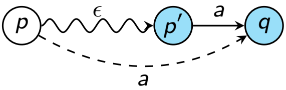
Desde un ϵ-NFA a un NFA

Construcción

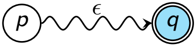
Dado un ϵ-NFA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ se define el NFA:

$$\mathcal{A}^{\not\epsilon} \quad = \quad (Q, \Sigma, \Delta^{\not\epsilon}, I, F^{\not\epsilon})$$

- para todo $p, q \in Q$, $(p, a, q) \in \Delta^{\not\epsilon}$ si, y solo si, existe $p' \in Q$ tal que:
  - $(p, \epsilon) \vdash_{\mathcal{A}}^{*} (p', \epsilon)$ y
  - $(p', a, q) \in \Delta$.



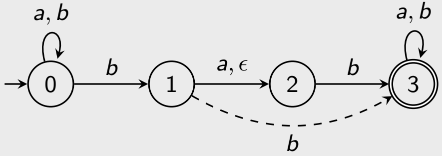
- $F^{\not\epsilon} = \{ p \in Q \mid \exists q \in F. (p, \epsilon) \vdash_{\mathcal{A}}^{*} (q, \epsilon) \}$



Por definición, si $(p, a, q) \in \Delta$ entonces $(p, a, q) \in \Delta^{\not\epsilon}$ para todo $a \in \Sigma$.

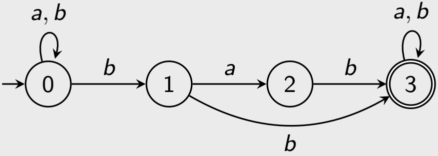
Desde un ϵ-NFA a un NFA

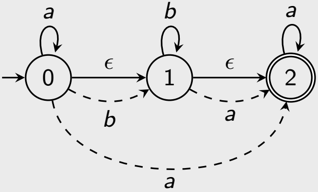
¿cuál es el autómata $\mathcal{A}^{\not\epsilon}$ para cada caso?

1. 

Desde un ϵ-NFA a un NFA

¿cuál es el autómata $\mathcal{A}^{\not\epsilon}$ para cada caso?

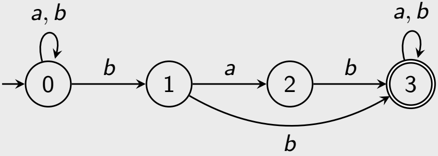
1. 

2. 

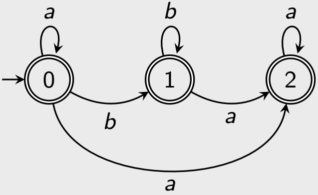
Desde un ϵ-NFA a un NFA

¿cuál es el autómata $\mathcal{A}^{\not\epsilon}$ para cada caso?

1.



2.



Desde un ϵ-NFA a un NFA

Teorema

Dado un ϵ-NFA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ se tiene que:

$$\mathcal{L}(\mathcal{A}) \;=\; \mathcal{L}(\mathcal{A}^{\not\epsilon})$$

Demostración

Demostrar que para todo $p \in Q$ y $w \in \Sigma^*$:

$$\exists q \in F.\ (p, w) \vdash^*_{\mathcal{A}} (q, \epsilon) \quad \text{si, y solo si,} \quad \exists q' \in F^{\not\epsilon}.\ (p, w) \vdash^*_{\mathcal{A}^{\not\epsilon}} (q', \epsilon)$$

De aquí **concluimos** que $\mathcal{A}$ acepta w si, y solo si, $\mathcal{A}^{\not\epsilon}$ acepta w.

Demostración $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^{\not\epsilon})$

Demostraremos que para todo $p \in Q$ y $w \in \Sigma^*$:

$$\exists q \in F.\ (p, w) \vdash^*_{\mathcal{A}} (q, \epsilon) \quad \text{si, y solo si,} \quad \exists q' \in F^{\not\epsilon}.\ (p, w) \vdash^*_{\mathcal{A}^{\not\epsilon}} (q', \epsilon)$$

Por **inducción** sobre el largo de w.

Caso base: Para $w = \epsilon$:

($\Rightarrow$) Sea $q \in F$ tal que $(p, \epsilon) \vdash^*_{\mathcal{A}} (q, \epsilon)$.

Por definición de $F^{\not\epsilon}$, se tiene que $p \in F^{\not\epsilon}$.

Por lo tanto, $(p, \epsilon) \vdash^*_{\mathcal{A}^{\not\epsilon}} (p, \epsilon)$. ✓

($\Leftarrow$) Sea $q' \in F^{\not\epsilon}$ tal que $(p, \epsilon) \vdash^*_{\mathcal{A}^{\not\epsilon}} (q', \epsilon)$.

Como $\mathcal{A}^{\not\epsilon}$ no tiene ϵ-transiciones, entonces $p = q'$ y $p \in F^{\not\epsilon}$.

Por definición de $F^{\not\epsilon}$, existe $q \in F$ tal que $(p, \epsilon) \vdash^*_{\mathcal{A}} (q, \epsilon)$. ✓

Demostración $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^{\notin})$

Demostraremos que para todo $p \in Q$ y $w \in \Sigma^*$:

$$\exists q \in F.\ (p, w) \vdash_{\mathcal{A}}^{*} (q, \epsilon) \quad \text{si, y solo si,} \quad \exists q' \in F^{\notin}.\ (p, w) \vdash_{\mathcal{A}^{\notin}}^{*} (q', \epsilon)$$

Caso inductivo: Sea $w = a \cdot u$ y $p \in Q$:

($\Leftarrow$) Sea $q' \in F^{\notin}$ tal que $(p, au) \vdash_{\mathcal{A}^{\notin}}^{*} (q', \epsilon)$.

Por definición de $\vdash_{\mathcal{A}^{\notin}}^{*}$ existen $p' \in Q$ tal que:

$$(p, au) \ \vdash_{\mathcal{A}^{\notin}} \ (p', u) \ \vdash_{\mathcal{A}^{\notin}}^{*} \ (q', \epsilon)$$

Demostración $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^{\not\epsilon})$

Demostraremos que para todo $p \in Q$ y $w \in \Sigma^*$:

$$\exists q \in F.\ (p, w) \vdash_{\mathcal{A}}^{*} (q, \epsilon) \quad \text{si, y solo si,} \quad \exists q' \in F^{\not\epsilon}.\ (p, w) \vdash_{\mathcal{A}^{\not\epsilon}}^{*} (q', \epsilon)$$

Caso inductivo: Sea $w = a \cdot u$ y $p \in Q$:

($\Leftarrow$) Sea $q' \in F^{\not\epsilon}$ tal que $(p, au) \vdash_{\mathcal{A}^{\not\epsilon}}^{*} (q', \epsilon)$.

Por definición de $\vdash_{\mathcal{A}^{\not\epsilon}}^{*}$ existen $p' \in Q$ tal que:

$$(p, au) \overset{(1)}{\vdash_{\mathcal{A}^{\not\epsilon}}} (p', u) \overset{(2)}{\vdash_{\mathcal{A}^{\not\epsilon}}^{*}} (q', \epsilon)$$

Por (1) sabemos que $(p, \epsilon) \vdash_{\mathcal{A}}^{*} (p'', \epsilon)$ y $(p'', a, p') \in \Delta$. (3)

Por (3), $(p, au) \vdash_{\mathcal{A}}^{*} (p', u)$. (4)

Como $|u| < |au|$ y (2), **por HI** existe $q \in F$: $(p', u) \vdash_{\mathcal{A}}^{*} (q, \epsilon)$. (5)

Juntando (4) y (5), tenemos que $(p, au) \vdash_{\mathcal{A}}^{*} (q, \epsilon)$. ✓

Demostración $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^{\notin})$

Demostraremos que para todo $p \in Q$ y $w \in \Sigma^*$:

$$\exists q \in F.\ (p, w) \vdash_{\mathcal{A}}^{*} (q, \epsilon) \quad \text{si, y solo si,} \quad \exists q' \in F^{\notin}.\ (p, w) \vdash_{\mathcal{A}^{\notin}}^{*} (q', \epsilon)$$

Caso inductivo: Sea $w = a \cdot u$ y $p \in Q$:

($\Rightarrow$) Sea $q \in F$ tal que $(p, au) \vdash_{\mathcal{A}}^{*} (q, \epsilon)$.

Por definición de $\vdash_{\mathcal{A}}^{*}$ existen $p', p'' \in Q$ tal que:

$$(p, au) \ \vdash_{\mathcal{A}}^{*} \ (p', au) \ \vdash_{\mathcal{A}} \ (p'', u) \ \vdash_{\mathcal{A}}^{*} \ (q, \epsilon)$$

Demostración $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^{\not\epsilon})$

Demostraremos que para todo $p \in Q$ y $w \in \Sigma^*$:

$$\exists q \in F.\ (p, w) \vdash^*_{\mathcal{A}} (q, \epsilon) \quad \text{si, y solo si,} \quad \exists q' \in F^{\not\epsilon}.\ (p, w) \vdash^*_{\mathcal{A}^{\not\epsilon}} (q', \epsilon)$$

Caso inductivo: Sea $w = a \cdot u$ y $p \in Q$:

($\Rightarrow$) Sea $q \in F$ tal que $(p, au) \vdash^*_{\mathcal{A}} (q, \epsilon)$.

Por definición de $\vdash^*_{\mathcal{A}}$ existen $p', p'' \in Q$ tal que:

$$(p, au) \overset{(1)}{\vdash^*_{\mathcal{A}}} (p', au) \overset{(2)}{\vdash_{\mathcal{A}}} (p'', u) \overset{(3)}{\vdash^*_{\mathcal{A}}} (q, \epsilon)$$

Por (1) tenemos que $(p, \epsilon) \vdash^*_{\mathcal{A}} (p', \epsilon)$. (4)

Por (2) tenemos que $(p', a, p'') \in \Delta$. (5)

Por (4) y (5), sabemos que $(p, a, p'') \in \Delta^{\not\epsilon}$ y $(p, a \cdot u) \vdash_{\mathcal{A}^{\not\epsilon}} (p'', u)$. (6)

Como $|u| < |au|$ y (3), **por HI** existe $q' \in F^{\not\epsilon}$: $(p'', u) \vdash^*_{\mathcal{A}^{\not\epsilon}} (q', \epsilon)$. (7)

Juntando (6) y (7), tenemos que $(p, au) \vdash^*_{\mathcal{A}^{\not\epsilon}} (q', \epsilon)$. ✓

■

Outline

NFA versus ϵ-NFA (clase anterior)

Desde Expresiones a Autómatas

Toda ExpReg se puede transformar en un autómata

Lenguajes Regulares

DFA $\equiv$ NFA $\equiv$ ϵ-NFA

$\cup\!\mid$

Regex

Para cada $R \in$ Regex construiremos un ϵ-NFA $\mathcal{A}_R$ **inductivamente**

Toda ExpReg se puede transformar en un autómata

Construcción inductiva

Para cada $R \in$ Regex, construimos un ϵ-NFA $\mathcal{A}_R$:

$$\mathcal{A}_R \;=\; (Q, \Sigma, \Delta, \{q_0\}, \{q_f\})$$

tal que $\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$.

Expresiones regulares (recordatorio)

R es una **expresión regular** sobre Σ si R es igual a:

1. a — para alguna letra $a \in \Sigma$.
2. ϵ
3. $\varnothing$
4. $(R_1 + R_2)$ — donde R_1 y R_2 son expresiones regulares.
5. $(R_1 \cdot R_2)$ — donde R_1 y R_2 son expresiones regulares.
6. (R_1^*) — donde R_1 es una expresión regular.

Toda ExpReg se puede transformar en un autómata

Construcción inductiva

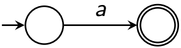
Para cada $R \in$ Regex, construimos un ϵ-NFA $\mathcal{A}_R$:

$$\mathcal{A}_R \;=\; (Q, \Sigma, \Delta, \{q_0\}, \{q_f\})$$

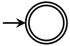
tal que $\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$.

Casos bases

1. si $R = a$, para alguna letra $a \in \Sigma$:

$\mathcal{A}_R :$ 

2. si $R = \epsilon$:

$\mathcal{A}_R :$ 

3. si $R = \varnothing$:

$\mathcal{A}_R :$ 

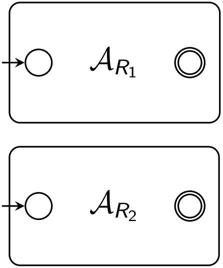
Toda ExpReg se puede transformar en un autómata

Construcción inductiva

Para cada $R \in$ Regex, construimos un ϵ-NFA $\mathcal{A}_R$ tal que $\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$:

4. si $R = (R_1 + R_2)$ donde R_1 y R_2 son expresiones regulares.

¿cómo construimos un autómata $\mathcal{A}_{R_1+R_2}$ para la expresión $R_1 + R_2$?



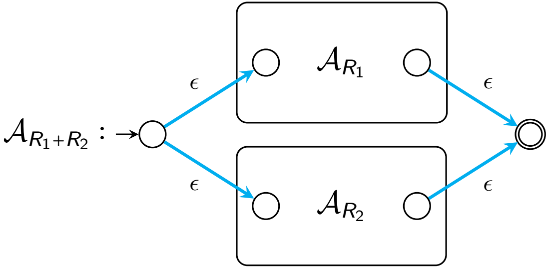
Toda ExpReg se puede transformar en un autómata

Construcción inductiva

Para cada $R \in$ Regex, construimos un ϵ-NFA $\mathcal{A}_R$ tal que $\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$:

4. si $R = (R_1 + R_2)$ donde R_1 y R_2 son expresiones regulares.

¿cómo construimos un autómata $\mathcal{A}_{R_1+R_2}$ para la expresión $R_1 + R_2$?



Toda ExpReg se puede transformar en un autómata

Construcción inductiva: $R = (R_1 + R_2)$

Por **inducción**, sea $\mathcal{A}_{R_1}$ y $\mathcal{A}_{R_2}$ los ϵ-NFA para R_1 y R_2 resp., tal que:

- $\mathcal{A}_{R_1} = (Q_1, \Sigma, \Delta_1, \{q_0^1\}, \{q_f^1\})$
- $\mathcal{A}_{R_2} = (Q_2, \Sigma, \Delta_2, \{q_0^2\}, \{q_f^2\})$

Definimos el ϵ-NFA $\mathcal{A}_{R_1+R_2} = (Q, \Sigma, \Delta, \{q_0\}, \{q_f\})$ tal que:

- $Q = Q_1 \uplus Q_2 \uplus \{q_0, q_f\}$
- $\Delta = \Delta_1 \uplus \Delta_2 \uplus \{(q_0, \epsilon, q_0^1), (q_0, \epsilon, q_0^2), (q_f^1, \epsilon, q_f), (q_f^2, \epsilon, q_f)\}$

Proposición

Si $R = (R_1 + R_2)$, entonces $\mathcal{L}(R_1 + R_2) = \mathcal{L}(\mathcal{A}_{R_1+R_2})$.

Demostración: ejercicio.

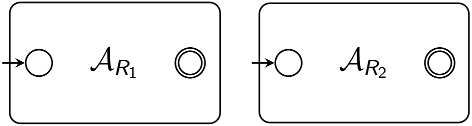
Toda ExpReg se puede transformar en un autómata

Construcción inductiva

Para cada $R \in \text{Regex}$, construimos un ϵ-NFA $\mathcal{A}_R$ tal que $\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$:

5. Si $R = (R_1 \cdot R_2)$ donde R_1 y R_2 son expresiones regulares.

¿cómo construimos un autómata $\mathcal{A}_{R_1 \cdot R_2}$ para la expresión $R_1 \cdot R_2$?



Toda ExpReg se puede transformar en un autómata

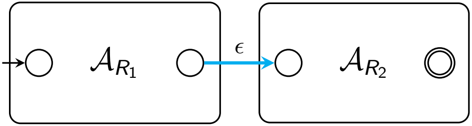
Construcción inductiva

Para cada $R \in$ Regex, construimos un ϵ-NFA $\mathcal{A}_R$ tal que $\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$:

5. Si $R = (R_1 \cdot R_2)$ donde R_1 y R_2 son expresiones regulares.

¿cómo construimos un autómata $\mathcal{A}_{R_1 \cdot R_2}$ para la expresión $R_1 \cdot R_2$?

$\mathcal{A}_{R_1 \cdot R_2}$:



Toda ExpReg se puede transformar en un autómata

Construcción inductiva: $R = (R_1 \cdot R_2)$

Por inducción, sea $\mathcal{A}_{R_1}$ y $\mathcal{A}_{R_2}$ los ϵ-NFA para R_1 y R_2 resp., tal que:

- $\mathcal{A}_{R_1} = (Q_1, \Sigma, \Delta_1, \{q_0^1\}, \{q_f^1\})$
- $\mathcal{A}_{R_2} = (Q_2, \Sigma, \Delta_2, \{q_0^2\}, \{q_f^2\})$

Definimos el ϵ-NFA $\mathcal{A}_{R_1 \cdot R_2} = (Q, \Sigma, \Delta, \{q_0^1\}, \{q_f^2\})$ tal que:

- $Q = Q_1 \uplus Q_2$
- $\Delta = \Delta_1 \uplus \Delta_2 \uplus \{(q_f^1, \epsilon, q_0^2)\}$

Proposición

Si $R = (R_1 \cdot R_2)$, entonces $\mathcal{L}(R_1 \cdot R_2) = \mathcal{L}(\mathcal{A}_{R_1 \cdot R_2})$.

Demostración: ejercicio.

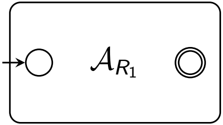
Toda ExpReg se puede transformar en un autómata

Construcción inductiva

Para cada $R \in$ Regex, construimos un ϵ-NFA $\mathcal{A}_R$ tal que $\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$:

6. si $R = (R_1^*)$ donde R_1 es una expresión regular.

¿cómo construimos un autómata $\mathcal{A}_{(R_1)^*}$ para la expresión $(R_1)^*$?



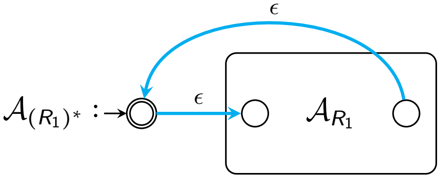
Toda ExpReg se puede transformar en un autómata

Construcción inductiva

Para cada $R \in$ Regex, construimos un ϵ-NFA $\mathcal{A}_R$ tal que $\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$:

6. si $R = (R_1^*)$ donde R_1 es una expresión regular.

¿cómo construimos un autómata $\mathcal{A}_{(R_1)^*}$ para la expresión $(R_1)^*$?



Toda ExpReg se puede transformar en un autómata

Construcción inductiva: $R = (R_1)^*$

Por inducción, sea $\mathcal{A}_{R_1}$ el ϵ-NFA para R_1, tal que:

- $\mathcal{A}_{R_1} = (Q_1, \Sigma, \Delta_1, \{q_0^1\}, \{q_f^1\})$.

Definimos el ϵ-NFA $\mathcal{A}_{(R_1)^*} = (Q, \Sigma, \Delta, \{q_0\}, \{q_0\})$ tal que:

- $Q = Q_1 \uplus \{q_0\}$
- $\Delta = \Delta_1 \uplus \{(q_0, \epsilon, q_0^1), (q_f^1, \epsilon, q_0)\}$

Proposición

Si $R = (R_1)^*$, entonces $\mathcal{L}((R_1)^*) = \mathcal{L}(\mathcal{A}_{(R_1)^*})$.

Demostración: ejercicio.

Regex $\subseteq$ ϵ-NFA

Teorema

Para todo $R \in$ Regex, existe un ϵ-NFA $\mathcal{A}_R$ tal que:

$$\mathcal{L}(R) = \mathcal{L}(\mathcal{A}_R)$$

En otras palabras, Regex $\subseteq$ ϵ-NFA.

¿de qué tamaño es $\mathcal{A}_R$ con respecto a R?

Cierre de clase

En esta clase vimos:

1. Como remover ϵ-transiciones de un ϵ-NFA.
2. Como convertir una expresión regular desde un ϵ-NFA.

Próxima clase: Como construir una expresión regular desde un NFA.