Autómatas con transiciones sin lectura

Clase 06

IIC2223 / IIC2224

Prof. Cristian Riveros

Outline

Regex (clase anterior)

ϵ-NFA

NFA versus ϵ-NFA

Outline

Regex (clase anterior)

ϵ-NFA

NFA versus ϵ-NFA

Expresiones regulares

Definición (Sintaxis)

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.

Denotaremos como Regex el conjunto de
todas las expresiones regulares sobre Σ

Cada expresión regular define un lenguaje

Definición (Semántica)

Para una expresión regular R cualquiera,
se define el lenguaje $\mathcal{L}(R) \subseteq \Sigma^*$ **inductivamente** como:

1. $\mathcal{L}(a) = \{a\}$ para toda letra $a \in \Sigma$.
2. $\mathcal{L}(\epsilon) = \{\epsilon\}$.
3. $\mathcal{L}(\varnothing) = \varnothing$.
4. $\mathcal{L}(R_1 + R_2) = \mathcal{L}(R_1) \cup \mathcal{L}(R_2)$ donde R_1 y R_2 son expresiones regulares.
5. $\mathcal{L}(R_1 \cdot R_2) = \mathcal{L}(R_1) \cdot \mathcal{L}(R_2)$ donde R_1 y R_2 son expresiones regulares.
6. $\mathcal{L}(R_1^*) = \bigcup_{k=0}^{\infty} \mathcal{L}(R_1)^k$ donde R_1 es una expresión regular.

Abreviaciones útiles para expresiones regulares

Definición

Usamos las siguientes **abreviaciones** de expresiones regulares para $R \in \text{Regex}$ y $\Sigma = \{a_1, \ldots, a_n\}$:

$$
\begin{aligned}
R^+ &\equiv R \cdot R^* \\
R^k &\equiv R \cdot \overset{k}{\cdots} \cdot R \\
R^? &\equiv R + \epsilon \\
\Sigma &\equiv a_1 + \ldots + a_n
\end{aligned}
$$

Más ejemplos de expresiones regulares

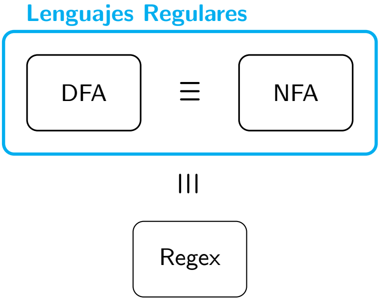
Defina una Regex para los siguientes lenguajes

1. Todas las palabras sobre $\{a, b\}$
   cuya ante-penúltima letra es una a-letra.

2. Todas las palabras sobre $\{a, b\}$
   con una cantidad par de a-letras.

3. Todas las palabras sobre $\{a, b\}$
   con a lo mas un par de a-letras consecutivas.

Mapa actual de nuestros modelos de computación



Para demostrar Regex $\subseteq$ NFA, primero necesitamos un nuevo modelo

Outline

Regex (clase anterior)

ϵ-NFA

NFA versus ϵ-NFA

Autómata finito no-determinista con ϵ-transiciones

¿qué tiene de nuevo este autómata?

1. tiene transiciones no deterministas y
2. tiene transiciones leyendo la palabra ϵ:

$$p \xrightarrow{\epsilon} q$$

¿Importancia?

- modelo **muy útil** para construir nuevos autómatas y
- **NO** agrega más poder de computación a los NFA.

Autómata finito no-determinista con ϵ-transiciones

Definición

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

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

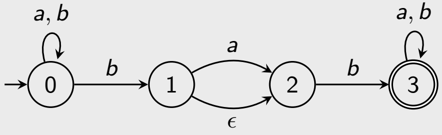
- Q es un conjunto finito de estados.
- Σ es el alfabeto de input.
- $I \subseteq Q$ es un conjunto de estados iniciales.
- $F \subseteq Q$ es el conjunto de estados finales (o aceptación).

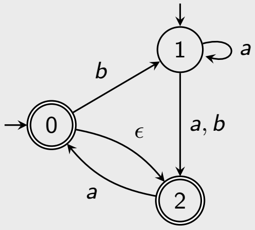
$$+$$

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

Autómata finito no-determinista con ϵ-transiciones

Algunos ejemplos

1. 

2. 

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.

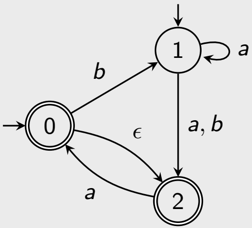
Definición

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 $u = c \cdot v$.

Ejemplo



$$(1, baa) \vdash_{\mathcal{A}} (2, aa)$$

$$(2, aa) \vdash_{\mathcal{A}} (0, a) \vdash_{\mathcal{A}} (2, a) \vdash_{\mathcal{A}} (0, \epsilon)$$

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) : $\quad (q, w) \ \vdash_{\mathcal{A}}^* \ (q, w)$

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

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

Ejecución de ϵ-NFA

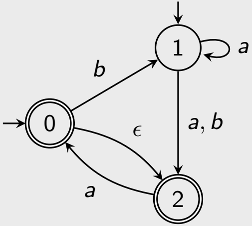
Se define $\vdash^*_{\mathcal{A}}$ como la clausura **refleja** y **transitiva** de $\vdash_{\mathcal{A}}$:

$$\text{para toda configuración } (q, w): \qquad (q, w) \vdash^*_{\mathcal{A}} (q, w)$$

$$\text{si } (p, u) \vdash_{\mathcal{A}} (p', w) \text{ y } (p', w) \vdash^*_{\mathcal{A}} (q, v): \qquad (p, u) \vdash^*_{\mathcal{A}} (q, v)$$

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

Ejemplo



$(1, baa) \vdash_{\mathcal{A}} (2, aa)$

$(2, aa) \vdash_{\mathcal{A}} (0, a) \vdash_{\mathcal{A}} (2, a) \vdash_{\mathcal{A}} (0, \epsilon)$

$(2, aa) \vdash^*_{\mathcal{A}} (0, \epsilon)$ y $(1, baa) \vdash^*_{\mathcal{A}} (0, \epsilon)$

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:

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

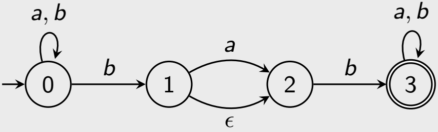
- El **lenguaje aceptado** por $\mathcal{A}$ se define como:

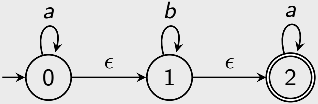
$$\mathcal{L}(\mathcal{A}) \ = \ \{w \in \Sigma^* \mid \mathcal{A} \text{ acepta } w\}$$

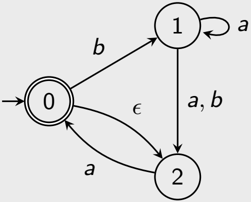
OJO: si $\mathcal{A}$ no tiene ϵ-transiciones, esta es una forma **alternativa** para definir ejecución para NFA y DFA.

Lenguaje aceptado por un ϵ-NFA

¿que lenguaje acepta cada ϵ-NFA?

1. 

2. 

3. 

Outline

Regex (clase anterior)

ϵ-NFA

NFA versus ϵ-NFA

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.

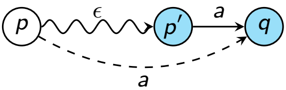
Desde un ϵ-NFA a un NFA

Construcción

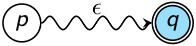
Dado un ϵ-NFA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ se define el NFA:

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

- 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
  - $(p', a, q) \in \Delta$.



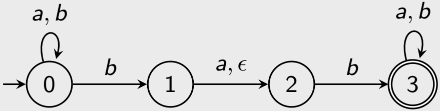
- $F^{\not\epsilon} = \{\, p \in Q \mid \exists q \in F.\, (p, \epsilon) \vdash^{*}_{\mathcal{A}} (q, \epsilon) \,\}$

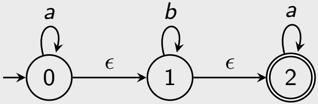


Por definición, si $(p, a, q) \in \Delta$ entonces $(p, a, q) \in \Delta^{\not\epsilon}$ para todo $a \in \Sigma$.

Desde un ϵ-NFA a un NFA

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

1. 

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

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

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

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 $F^{\notin}$, se tiene que $p \in F^{\notin}$.

Por lo tanto, $(p, \epsilon) \vdash_{\mathcal{A}^{\notin}}^{*} (p, \epsilon)$. ✓

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

Como $\mathcal{A}^{\notin}$ no tiene ϵ-transiciones, entonces $p = q'$ y $p \in F^{\notin}$.

Por definición de $F^{\notin}$, 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 $(p, au) \vdash^*_{\mathcal{A}} (p', u)$. (?) (3)

Como $|u| < |au|$ y (2), **por HI** existe $q \in F$: $(p', u) \vdash^*_{\mathcal{A}} (q, \epsilon)$. (4)

Juntando (3) y (4), 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 $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}^{\notin})$

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:

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

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 $(p, a, p'') \in \Delta^{\notin}$ y $(p, a \cdot u) \vdash_{\mathcal{A}^{\notin}} (p'', u)$. (6)

Como $|u| < |au|$ y (3), **por HI** existe $q' \in F^{\notin}$: $(p'', u) \vdash^*_{\mathcal{A}^{\notin}} (q', \epsilon)$. (7)

Juntando (6) y (7), tenemos que $(p, au) \vdash^*_{\mathcal{A}^{\notin}} (q', \epsilon)$. ✓

■

Cierre de la clase

Lenguajes Regulares

DFA $\equiv$ NFA $\equiv$ ϵ-NFA

$\equiv$

Regex

Próxima clase: demostraremos que
todos definen el mismo conjunto de lenguajes.