Transductores

Clase 15

IIC2223 / IIC2224

Prof. Cristian Riveros

Outline

Evaluación de NFA (clase anterior)

Transductores

Propiedades de transductores

Outline

Evaluación de NFA (clase anterior)

Transductores

Propiedades de transductores

¿cómo evaluamos un autómata no-determinista?

`PROBLEMA:`	Evaluación de NFA
`INPUT:`	un autómata no-determinista $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ y
	un documento w
`OUTPUT:`	`TRUE` si, y solo si, $w \in \mathcal{L}(R)$

Tamaño del input

- $|w| :=$ largo de documento
- $|\mathcal{A}| := |Q| + |\Delta|$

Algoritmo **lineal en data-complexity** y
ojalá **polinomial en combined-complexity**.

Algoritmos de evaluación de autómatas no-deterministas

Veremos varias soluciones ...

1. Backtracking ✓
2. DFA
3. NFA determinización
4. NFA on-the-fly

Solución 2: DFA

Para un DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ y una palabra $w = a_1 \ldots a_n$.

Function `eval-DFA` *($\mathcal{A}$, w)*

$q := q_0$
for $i = 1$ **to** n **do**
  $q := \delta(q, a_i)$
return check $(q \in F)$

Análisis de tiempo

- ¿cómo hacemos $q := \delta(q, a_i)$ de manera **eficiente**?
- ¿cuál es el tiempo de `eval-DFA` en el **peor caso**? $\mathcal{O}(|\mathcal{A}| + |w|)$

¿para qué nos sirve **evaluar** un DFA?

Solución 3: NFA determinización

Para un NFA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ y una palabra $w = a_1 \dots a_n$.

Function `eval-NFA` *($\mathcal{A}$, w)*

$\mathcal{A}^{\text{det}} :=$ `NFAtoDFA` $(\mathcal{A})$

return `eval-DFA` $(\mathcal{A}^{\text{det}}, w)$

Análisis de tiempo

- ¿cuál es el tiempo de `eval-NFA` en el **peor caso**? $\mathcal{O}(2^{|Q|} + |w|)$

¿cuál es el problema con esta solución?

¿es necesario construir la determinización completa?

Recordatorio

Para un autómata no-determinista $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$,
se define el autómata determinista (**determinización** de $\mathcal{A}$):

$$\mathcal{A}^{\text{det}} \quad = \quad (2^Q, \Sigma, \delta^{\text{det}}, q_0^{\text{det}}, F^{\text{det}})$$

- $2^Q = \{S \mid S \subseteq Q\}$ es el conjunto potencia de Q.
- $q_0^{\text{det}} = I$.
- $\delta^{\text{det}} : 2^Q \times \Sigma \to 2^Q$ tal que:

$$\delta^{\text{det}}(S, a) = \{q \in Q \mid \exists p \in S.\ (p, a, q) \in \Delta\}$$

- $F^{\text{det}} = \{S \in 2^Q \mid S \cap F \neq \varnothing\}$.

Solución 4: NFA *on-the-fly*

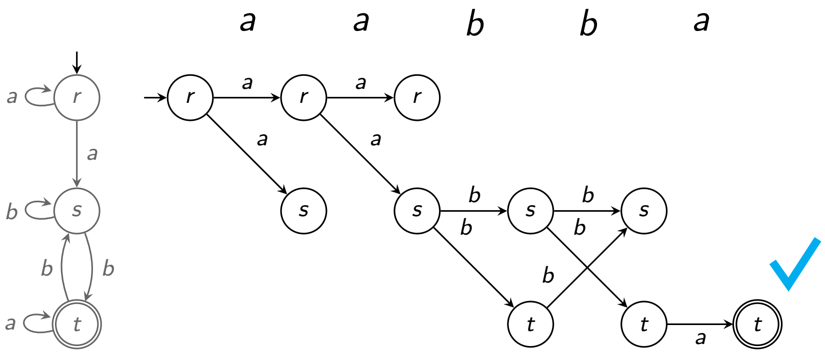
$\delta^{\text{det}} : 2^Q \times \Sigma \to 2^Q$ tal que:

$$\delta^{\text{det}}(S, a) = \{q \in Q \mid \exists p \in S.\ (p, a, q) \in \Delta\}$$

Estrategia *on-the-fly*

1. Mantenemos un conjunto S de estados actuales.
2. Por cada nueva letra a, calculamos el conjunto $\delta^{\text{det}}(S, a)$.

Solución 4: NFA *on-the-fly* (ejemplo)



Solución 4: NFA *on-the-fly*

Para un NFA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ y una palabra $w = a_1 \dots a_n$.

Function `eval-NFAonthefly` $(\mathcal{A}, w)$

$S := I$

for $i = 1$ **to** n **do**

$\quad S_{\text{old}} := S$

$\quad S := \varnothing$

$\quad$ **foreach** $p \in S_{old}$ **do**

$\quad\quad S \;:=\; S \cup \{q \mid (p, a_i, q) \in \Delta\}$

return check $(S \cap F \neq \varnothing)$

Análisis de tiempo

- ¿cuál es el tiempo de `eval-NFAonthefly` en el **peor caso**? $\mathcal{O}(|\mathcal{A}| \cdot |w|)$

¿es posible mejorar este algoritmo?

... es posible demostrar que no.

Resumen de técnicas de evaluación simple

Para un autómata $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ y una palabra $w = a_1 \ldots a_n$:

	Tiempo
Backtracking	$\mathcal{O}(\lvert\mathcal{A}\rvert^{\lvert w\rvert})$
DFA	$\mathcal{O}(\lvert\mathcal{A}\rvert + \lvert w\rvert)$
NFA	$\mathcal{O}(2^{\lvert Q\rvert} + \lvert w\rvert)$
NFA on-the-fly	$\mathcal{O}(\lvert\mathcal{A}\rvert \cdot \lvert w\rvert)$

¿cuál funciona mejor **en la práctica**?

Bonus: NFA *on-the-fly* en la práctica

Para un NFA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ y una palabra $w = a_1 \dots a_n$.

Function `eval-NFAonthefly` *($\mathcal{A}$, w)*

- **let** $H :=$ HashTable en $2^Q \times \Sigma$
- $S := I$
- **for** $i = 1$ **to** n **do**
  - **if** $H(S, a)$ is not null **then**
    - $S := H(S, a_i)$
  - **else**
    - $S_{\text{old}} := S$
    - $S := \varnothing$
    - **foreach** $p \in S_{old}$ **do**
      - $S \ := \ S \cup \{q \mid (p, a, q) \in \Delta\}$
    - $H(S_{\text{old}}, a_i) := S$
- **return check** $(S \cap F \neq \varnothing)$

¿cúal es la ventaja de este algoritmo en **la práctica**?

Outline

Evaluación de NFA (clase anterior)

Transductores

Propiedades de transductores

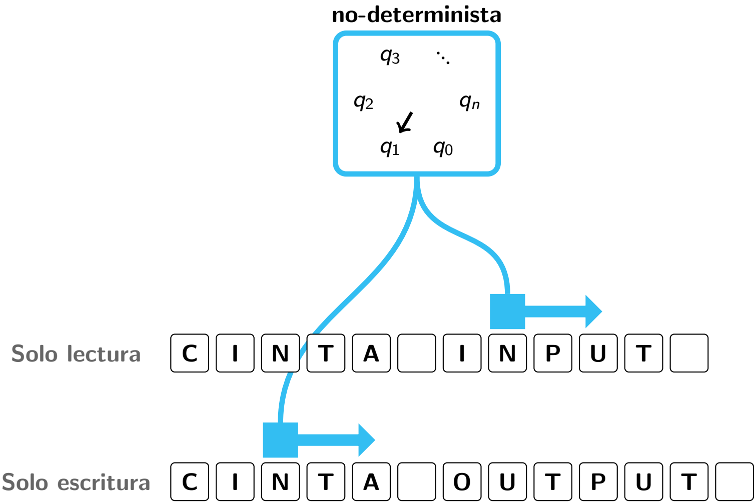
¿cuánto se parece un autómata a un algoritmo?

¿cuáles son las diferencias?

1. Memoria. ✕
2. “Movimiento” de la máquina. ✓
3. Output. **?**

En esta clase, veremos como extender autómatas con 3.

Transductores



Definición de transductor

Definición

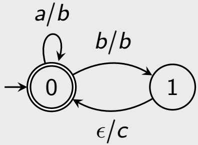
Un transductor (*en inglés*, transducer) es una tupla:

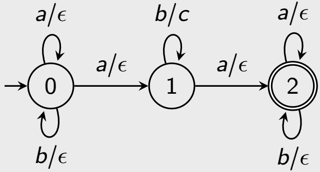
$$\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$$

- Q es un conjunto finito de estados.
- Σ es el alfabeto de input.
- Ω es el alfabeto de **output**.
- $\Delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Omega \cup \{\epsilon\}) \times Q$ es la **relación de transición**.
- $I \subseteq Q$ es un conjunto de estados iniciales.
- $F \subseteq Q$ es el conjunto de estados finales.

Definición de transductor

Algunos ejemplos

1. 

2. 

Configuración de un transductor

Sea $\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$ un transductor.

Definiciones

- Un par $(q, u, v) \in Q \times \Sigma^* \times \Omega^*$ es una **configuración** de $\mathcal{T}$.
- Una configuración (q, u, ϵ) es **inicial** si $q \in I$.
- Una configuración (q, ϵ, v) es **final** si $q \in F$.

"Intuitivamente, una configuración (q, au, vb) representa que $\mathcal{T}$ se encuentra en el estado q procesando la palabra au y leyendo a, y hasta ahora grabó la palabra vb y el último símbolo impreso es b."

Ejecución de un transductor

Sea $\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$ un transductor.

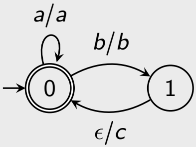
Definición

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

$$(p, u_1, v_1) \ \vdash_{\mathcal{T}} \ (q, u_2, v_2)$$

si, y solo si, existe $(p, a, b, q) \in \Delta$ tal que $u_1 = a \cdot u_2$ y $v_2 = v_1 \cdot b$.

Ejemplo



$$(0, aba, \epsilon) \vdash_{\mathcal{T}} (0, ba, a)$$

$$(0, ba, a) \vdash_{\mathcal{T}} (1, a, ab) \vdash_{\mathcal{T}} (0, a, abc) \vdash_{\mathcal{T}} (0, \epsilon, abca)$$

Ejecución de un transductor

Sea $\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$ un transductor.

Definición

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

$$(p, u_1, v_1) \ \vdash_{\mathcal{T}} \ (q, u_2, v_2)$$

si, y solo si, existe $(p, a, b, q) \in \Delta$ tal que $u_1 = a \cdot u_2$ y $v_2 = v_1 \cdot b$.

$$\vdash_{\mathcal{T}} \ \subseteq \ (Q \times \Sigma^* \times \Omega^*) \times (Q \times \Sigma^* \times \Omega^*).$$

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

para toda configuración (q, u, v) : $\quad (q, u, v) \ \vdash^*_{\mathcal{T}} \ (q, u, v)$

si $(q_1, u_1, v_1) \vdash^*_{\mathcal{T}} (q_2, u_2, v_2)$ y
$(q_2, u_2, v_2) \vdash_{\mathcal{T}} (q_3, u_3, v_3)$: $\quad (q_1, u_1, v_1) \ \vdash^*_{\mathcal{T}} \ (q_3, u_3, v_3)$

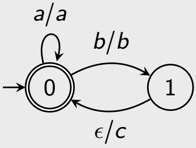
Ejecución de un transductor

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

$$\text{para toda configuración } (q, u, v): \qquad (q, u, v) \ \vdash^*_{\mathcal{T}} \ (q, u, v)$$

$$\begin{array}{l} \text{si } (q_1, u_1, v_1) \vdash^*_{\mathcal{T}} (q_2, u_2, v_2) \text{ y} \\ \qquad\qquad (q_2, u_2, v_2) \vdash_{\mathcal{T}} (q_3, u_3, v_3): \end{array} \qquad (q_1, u_1, v_1) \ \vdash^*_{\mathcal{T}} \ (q_3, u_3, v_3)$$

Ejemplo



$$\frac{\begin{array}{c}(0, aba, \epsilon) \vdash_{\mathcal{T}} (0, ba, a) \\ (0, ba, a) \vdash_{\mathcal{T}} (1, a, ab) \vdash_{\mathcal{T}} (0, a, abc) \vdash_{\mathcal{T}} (0, \epsilon, abca)\end{array}}{(0, aba, \epsilon) \vdash^*_{\mathcal{T}} (0, ba, a) \ \text{ y } \ (0, aba, \epsilon) \vdash^*_{\mathcal{T}} (0, \epsilon, abca)}$$

Función definida por un transductor

Sea $\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$ un transductor y $u, v \in \Sigma^*$.

Definiciones

- $\mathcal{T}$ **entrega** v con **input** u si existe una configuración **inicial** (q_0, u, ϵ) y una configuración **final** (q_f, ϵ, v) tal que:

$$(q_0, u, \epsilon) \vdash_{\mathcal{T}}^{*} (q_f, \epsilon, v)$$

- Se define la función $[\![\mathcal{T}]\!] : \Sigma^* \to 2^{\Omega^*}$:

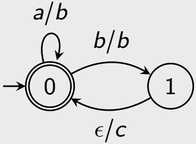
$$[\![\mathcal{T}]\!](u) = \{v \in \Omega^* \mid \mathcal{T} \text{ entrega } v \text{ con input } u\}$$

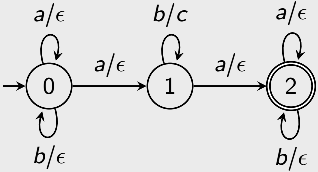
- Se dice que $f : \Sigma^* \to 2^{\Omega^*}$ es una **función racional** si existe un transductor $\mathcal{T}$ tal que $f = [\![\mathcal{T}]\!]$.

Un transductor define una función de palabras a conjunto de palabras.

Función definida por un transductor

¿qué función define cada transductor?

1. 

2. 

Funciones versus relaciones

Dos interpretaciones para un transductor

1. $\mathcal{T}$ define la **función** $[\![\mathcal{T}]\!] : \Sigma^* \to 2^{\Omega^*}$:

$$[\![\mathcal{T}]\!](u) \;\;=\;\; \{v \in \Omega^* \mid \mathcal{T} \text{ entrega } v \text{ con input } u\}$$

2. $\mathcal{T}$ define la **relación** $[\![\mathcal{T}]\!] \subseteq \Sigma^* \times \Omega^*$:

$$(u, v) \in [\![\mathcal{T}]\!] \quad \text{si, y solo si,} \quad \mathcal{T} \text{ entrega } v \text{ con input } u$$

Desde ahora, hablaremos de función o relación **indistintamente** y hablaremos de las **relaciones racionales** (definidas por un transductor).

Outline

Evaluación de NFA (clase anterior)

Transductores

Propiedades de transductores

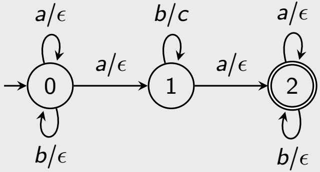
Lenguaje de input y lenguaje de output

Definiciones

Para una relación $R \subseteq \Sigma^* \times \Omega^*$ se define:

- $\pi_1(R) \ = \ \{ \ u \in \Sigma^* \mid \exists v \in \Omega^*. \ (u, v) \in R \ \}$.
- $\pi_2(R) \ = \ \{ \ v \in \Omega^* \mid \exists u \in \Sigma^*. \ (u, v) \in R \ \}$.

¿cuál es el lenguaje definido por $\pi_1(\llbracket \mathcal{T} \rrbracket)$ y $\pi_2(\llbracket \mathcal{T} \rrbracket)$?



Lenguaje de input y lenguaje de output

Definiciones

Para una relación $R \subseteq \Sigma^* \times \Omega^*$ se define:

- $\pi_1(R) \;=\; \{\; u \in \Sigma^* \mid \exists v \in \Omega^*.\ (u, v) \in R \;\}.$
- $\pi_2(R) \;=\; \{\; v \in \Omega^* \mid \exists u \in \Sigma^*.\ (u, v) \in R \;\}.$

Teorema

Si $\mathcal{T}$ es un transductor,
entonces $\pi_1(\llbracket \mathcal{T} \rrbracket)$ y $\pi_2(\llbracket \mathcal{T} \rrbracket)$ son lenguajes regulares sobre Σ y Ω, resp.

Demostración: $\pi_1(\llbracket \mathcal{T} \rrbracket)$

Para $\mathcal{T} = (Q, \Sigma, \Omega, \Delta, I, F)$, defina $\mathcal{A}_1 = (Q, \Sigma, \Delta_1, I, F)$ tal que:

$$(p, a, q) \in \Delta_1 \quad \text{si, y solo si,} \quad \exists b \in \Omega \cup \{\epsilon\}.\ (p, a, b, q) \in \Delta$$

y demuestre que $\mathcal{L}(\mathcal{A}_1) = \pi_1(\llbracket \mathcal{T} \rrbracket)$. ■

Operaciones de relaciones

Teorema

Sea $\mathcal{T}_1$ y $\mathcal{T}_2$ dos transductores con Σ y Ω alfabetos de input y output.

Las siguientes son **relaciones racionales**.

1. $[\![\mathcal{T}_1]\!] \cup [\![\mathcal{T}_2]\!] = \{(u, v) \in \Sigma^* \times \Omega^* \mid (u, v) \in [\![\mathcal{T}_1]\!] \vee (u, v) \in [\![\mathcal{T}_2]\!]\}$.
2. $[\![\mathcal{T}_1]\!] \cdot [\![\mathcal{T}_2]\!] = \{(u_1 u_2, v_1 v_2) \in \Sigma^* \times \Omega^* \mid (u_1, v_1) \in [\![\mathcal{T}_1]\!] \wedge (u_2, v_2) \in [\![\mathcal{T}_2]\!]\}$.
3. $[\![\mathcal{T}_1]\!]^* = \bigcup_{k=0}^{\infty} [\![\mathcal{T}_1]\!]^k$.

Demostración.

Operaciones de relaciones

Teorema

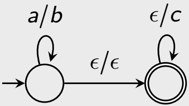
Existen transductores $\mathcal{T}_1$ y $\mathcal{T}_2$ sobre Σ y Ω tal que:

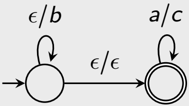
$$[\![\mathcal{T}_1]\!] \cap [\![\mathcal{T}_2]\!] \;=\; \{(u,v) \in \Sigma^* \times \Omega^* \mid (u,v) \in [\![\mathcal{T}_1]\!] \wedge (u,v) \in [\![\mathcal{T}_2]\!]\}$$

NO es una relación racional.

Demostración

Considere los siguientes transductores:





$[\![\mathcal{T}_1]\!] = \{(a^n, b^n c^m) \mid n \geq 0, m \geq 0\}$ $\quad$ $[\![\mathcal{T}_2]\!] = \{(a^n, b^m c^n) \mid n \geq 0, m \geq 0\}$

pero $[\![\mathcal{T}_1]\!] \cap [\![\mathcal{T}_2]\!] = \{(a^n, b^n c^n) \mid n \geq 0\}$,
y por lo tanto $[\![\mathcal{T}_1]\!] \cap [\![\mathcal{T}_2]\!]$ no es racional (¿por qué?) $\blacksquare$

Cierre de clase

En esta clase vimos:

1. Modelo de transductores.
2. Algunas propiedades de transductores.

Próxima clase: Análisis léxico.