No-determinismo

Clase 03

IIC 2223

Prof. Cristian Riveros

¿qué significa "no-determinismo"?

"Indeterminism is the concept that events (certain events, or events of certain types) are not caused deterministically *(cf. causality) by prior events. It is the opposite of determinism and related to chance. It is highly relevant to the philosophical problem of* free will*."*

Wikipedia.

¿por qué nuestros autómatas son deterministas?

Definición

Un autómata finito **determinista** (DFA) es una estructura:

$$\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$$

- Q es un conjunto finito de estados.
- Σ es el alfabeto de input.
- $\delta : Q \times \Sigma \to Q$ es la **función de transición**.
- $q_0 \in Q$ es el estado inicial.
- $F \subseteq Q$ es el conjunto de estados finales (o aceptación).

¿qué sería un autómata finito **no-determinista**?

Outline

Definición de NFA

Relevancia del concepto

Comparación con DFA

Outline

Autómata finito no-determinista

Definición

Un autómata finito **no-determinista** (NFA) es una estructura:

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

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

$$+$$

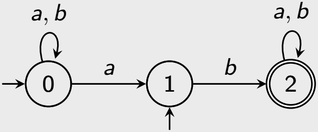
- $\Delta \subseteq Q \times \Sigma \times Q$ es la **relación de transición**.
- $I \subseteq Q$ es un **conjunto de estados iniciales**.

Autómata finito no-determinista

Ejemplo

- $Q = \{0, 1, 2\}$
- $\Sigma = \{a, b\}$
- $\Delta \subseteq Q \times \Sigma \times Q$ se define como:

$$
\begin{array}{rcl}
(0, a, 0) & \in & \Delta \\
(0, a, 1) & \in & \Delta \\
(0, b, 0) & \in & \Delta \\
(1, b, 2) & \in & \Delta \\
(2, a, 2) & \in & \Delta \\
(2, b, 2) & \in & \Delta
\end{array}
$$



- $I = \{0, 1\}$
- $F = \{2\}$

¿cómo ejecuto un autómata no-determinista?

Sea:

- Un autómata finito no-determinista $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$.
- El input $w = a_1 a_2 \ldots a_n \in \Sigma^*$.

Una **ejecución** (o run) ρ de $\mathcal{A}$ sobre w es una secuencia:

$$\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} p_n$$

- $p_0 \in I$
- para todo $i \in \{0, \ldots, n-1\}$, $(p_i, a_{i+1}, p_{i+1}) \in \Delta$.

Una ejecución ρ de $\mathcal{A}$ sobre w es de **aceptación** si:

$$p_n \in F.$$

Desde ahora hablaremos de **las ejecuciones** de $\mathcal{A}$ sobre w

Lenguaje aceptado por un autómata no-determinista

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

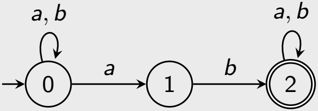
Definiciones

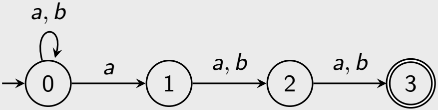
- $\mathcal{A}$ **acepta** w si **existe** una ejecución de $\mathcal{A}$ sobre w que es de aceptación.
- $\mathcal{A}$ **rechaza** w si **todas** las ejec. de $\mathcal{A}$ sobre w **NO** son de aceptación.
- El **lenguaje aceptado** por $\mathcal{A}$ se define como:

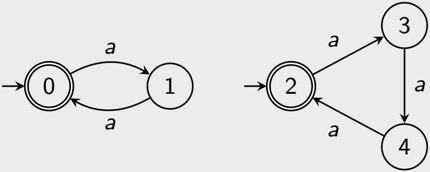
$$\mathcal{L}(\mathcal{A}) \;=\; \{w \in \Sigma^* \mid \mathcal{A} \text{ acepta } w\}$$

Lenguaje aceptado por un autómata no-determinista

¿qué lenguaje acepta cada autómata no-determinista?

1. 

2. 

3. 

Interpretación de la relación de transición

Un autómata finito **no-determinista** (NFA) es una estructura:

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

1. $\Delta \subseteq Q \times \Sigma \times Q$ es la **relación de transición**.

   "$(q, a, p) \in \Delta$ entonces existe una transición desde q *a* p *al leer* a*."*

2. $I \subseteq Q$ es un **conjunto de estados iniciales**.

   "$p \in I$ entonces p *es un posible estado inicial del autómata."*

Interpretación de la relación de transición

Un autómata finito **no-determinista** (NFA) es una estructura:

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

Otras posibles definición:

1'. $\Delta : Q \times \Sigma \to 2^Q$ es una **función de transición**.

“$q \in \Delta(p, a)$ entonces

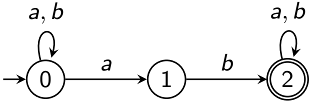
q *es un posible estado que puedo llegar desde* p *al leer* a.”

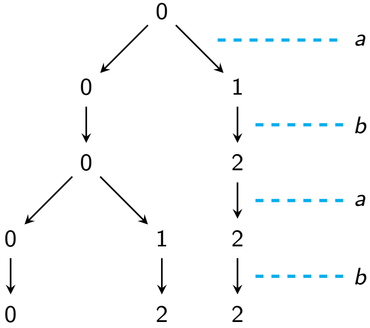
Esta última definición es la que posiblemente verán en libros

Interpretación del no-determinismo

El **no-determinismo** puede ser visto como:

1. Paralelización infinita.
   - Cada ejecución es un thread distinto.





2. "Guessing and Verifying" (adivinar y verificar).

Interpretación del no-determinismo

El no-determinismo NO debe ser visto como:

- explicitamente como el **indeterminismo** o “libre albedrío”.
  - Para un input, un NFA siempre produce el mismo resultado.

- comportamiento **aleatorio** del autómata.

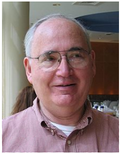
Outline

¿qué tan importante es el no-determinismo en CS?

Propuesto en el paper:

"Finite Automata and Their Decision Problem" (1959)

Michael O. Rabin
- Israel
- 89 años



Dana Scott
- EE UU
- 88 años

¿qué tan importante es el no-determinismo en CS?

Ambos ganadores del **Turing Award** (Novel en Computación) en 1979:

> *"For their joint paper 'Finite Automata and Their Decision Problem' which introduced the idea of* nondeterministic machines*, which has proved to be an enormously valuable concept. Their (Scott and Rabin) classic paper has been a continuous source of inspiration for* subsequent work *in this field."*

ACM

¿cuál ha sido este legado o "subsequent work" en el área?

¿cuál es la pregunta abierta más importante en CS?

$$\mathsf{P} \stackrel{?}{=} \mathsf{NP}$$

"If the solution to a problem can be quickly verified by a computer, can the computer also solve that problem quickly?"

Wikipedia.

¿cuál es la pregunta abierta más importante en CS?

Uno de los 7 "Millennium Prize Problems":

1. Yang–Mills and Mass Gap
2. Riemann Hypothesis
3. **P vs NP Problem**
4. Navier–Stokes Equation
5. Hodge Conjecture
6. ~~Poincaré Conjecture~~
7. Birch and Swinnerton-Dyer Conjecture

propuestos por *The Clay Mathematics Institute*.

¿cuál es la pregunta abierta mas importante en CS?

$$\mathsf{P} \stackrel{?}{=} \mathsf{NP}$$

US$ 1 millón a quien lo resuelva.

"Aside from being an important problem in computational theory, a proof either way would have profound implications *for mathematics, cryptography, algorithm research, artificial intelligence, game theory, multimedia processing, philosophy, economics and many other fields."*

Wikipedia.

Outline

¿qué tan poderoso es el no-determinismo en autómatas?

$$\text{DFA} \subseteq \text{NFA}$$

Dado un autómata finito determinista $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ podemos construir un autómata finito no-determinista:

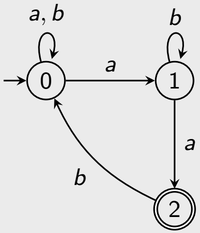
$$\mathcal{A}' \quad = \quad (Q, \Sigma, \Delta, I, F)$$

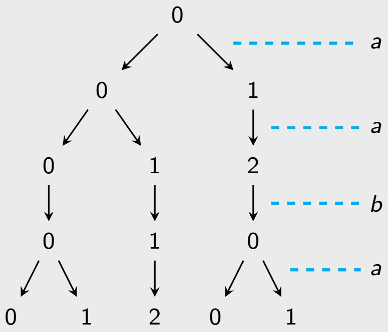
- $(p, a, q) \in \Delta$ si, y solo si, $\delta(p, a) = q$.
- $I = \{q_0\}$.
- $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.

¿ DFA $\subsetneq$ NFA ?

¿qué tan poderoso es el no-determinismo en autómatas?

Ejemplo





¿puede un autómata determinista almacenar **todas** las ejecuciones?

¿qué tan poderoso es el no-determinismo en autómatas?

Teorema

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

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

En otras palabras, DFA $\equiv$ NFA.

Ambos modelos computan lo mismo

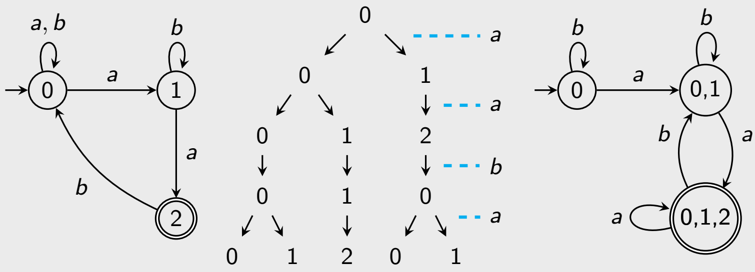
Demostración

Para demostrar este resultado,
construiremos la "determinación" del autómata no-determinista $\mathcal{A}$.

Idea de determinización

"Almacenar en el autómata determinista todos los estados actuales de las ejecuciones en curso (sin repetidos)."

Ejemplo



Determinización (subset construction)

Formalización

Para un autómata no-determinista $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$, definimos 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:

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

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

Determinización (subset construction)

Proposición

Dado un autómata no-determinista $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ se tiene que:

$$\mathcal{L}(\mathcal{A}) \;\; = \;\; \mathcal{L}(\mathcal{A}^{\text{det}})$$

¿cómo demostramos que ambos autómatas definen el mismo lenguaje?

Determinización (subset construction)

Demostración: $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}^{\text{det}})$

Sea $w = a_1 a_2 \ldots a_n \in \mathcal{L}(\mathcal{A})$.

Existe una ejecución ρ de $\mathcal{A}$ sobre w:

$$\rho : p_0 \overset{a_1}{\to} p_1 \overset{a_2}{\to} \ldots \overset{a_n}{\to} p_n$$

- $p_0 \in I$.
- $(p_i, a_{i+1}, p_{i+1}) \in \Delta \quad \forall\, i \in \{0, \ldots, n-1\}$.
- $p_n \in F$.

Como $\mathcal{A}^{\text{det}}$ es determinista, entonces existe una ejec. ρ' de $\mathcal{A}^{\text{det}}$ sobre w:

$$\rho' : S_0 \overset{a_1}{\to} S_1 \overset{a_2}{\to} \ldots \overset{a_n}{\to} S_n$$

- $S_0 = I$.
- $\delta^{\text{det}}(S_i, a_{i+1}) = S_{i+1} \quad \forall\, i \in \{0, 1, \ldots, n-1\}$.

¿qué debemos demostrar?

Determinización (subset construction)

Demostración: $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}^{\text{det}})$

PD: $p_i \in S_i$ para todo $i \in \{0, 1, \ldots, n-1\}$.

Por **inducción** sobre i.

Caso base: $p_0 \in S_0$ ✓ (¿por qué?)

Inducción: Suponemos que $p_i \in S_i$ y demostramos para $i+1$.

Como sabemos que:

- $\delta^{\text{det}}(S_i, a_{i+1}) = S_{i+1} = \{q \in Q \mid \exists p \in S_i.\ (p, a, q) \in \Delta\}$ y
- $(p_i, a_{i+1}, p_{i+1}) \in \Delta$.

Entonces $p_{i+1} \in S_{i+1}$ (¿por qué?). ✓

Como $p_n \in S_n \overset{?}{\Rightarrow} S_n \cap F \neq \varnothing \overset{?}{\Rightarrow} S_n \in F^{\text{det}}$.

Por lo tanto, $w \in \mathcal{L}(\mathcal{A}^{\text{det}})$.

Determinización (subset construction)

Demostración: $\mathcal{L}(\mathcal{A}^{\text{det}}) \subseteq \mathcal{L}(\mathcal{A})$

Sea $w = a_1 a_2 \dots a_n \in \mathcal{L}(\mathcal{A}^{\text{det}})$.

Existe una ejecución ρ de $\mathcal{A}^{\text{det}}$ sobre w:

$$\rho : S_0 \xrightarrow{a_1} S_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} S_n$$

- $S_0 = I$.
- $\delta^{\text{det}}(S_i, a_{i+1}) = S_{i+1} \quad \forall\, i \in \{0, 1, \dots, n-1\}$.
- $S_n \in F^{\text{det}}$. $\qquad (S_n \cap F \neq \emptyset)$

¿cómo demostramos una **ejecución de aceptación** de $\mathcal{A}$ sobre w?

Determinización (subset construction)

Demostración: $\mathcal{L}(\mathcal{A}^{\text{det}}) \subseteq \mathcal{L}(\mathcal{A})$

PD: Para todo $i \leq n$ y para todo $p \in S_i$, existe una ejecución:

$$\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_i} p_i = p$$

1. $p_0 \in I$.
2. $(p_j, a_{j+1}, p_{j+1}) \in \Delta \quad \forall\, j \in \{0, \dots, i-1\}$.

Por **inducción** sobre i.

Caso base: Si $p \in S_0 = I$, entonces la ejec. $\rho : p$ cumple 1. y 2. ✓

Determinización (subset construction)

Demostración: $\mathcal{L}(\mathcal{A}^{\text{det}}) \subseteq \mathcal{L}(\mathcal{A})$

PD: Para todo $i \leq n$ y para todo $p \in S_i$, existe una ejecución:

$$\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_i} p_i = p$$

1. $p_0 \in I$.
2. $(p_j, a_{j+1}, p_{j+1}) \in \Delta \quad \forall\, j \in \{0, \dots, i-1\}$.

Inducción: Supongamos que se cumple para todo $p \in S_i$. Sea $q \in S_{i+1}$.

Como $\delta^{\text{det}}(S_i, a_{i+1}) = S_{i+1} = \{q \in Q \mid \exists p \in S_i.\ (p, a, q) \in \Delta\}$ y $q \in S_{i+1}$
entonces existe $p \in S_i$ tal que $(p, a_{i+1}, q) \in \Delta$.

Por **HI**, existe $\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_i} p_i = p$ que satisface 1. y 2.

Por lo tanto, $\rho' : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_i} p_i \xrightarrow{a_{i+1}} q$ también satisface 1. y 2. ✓

Determinización (subset construction)

Demostración: $\mathcal{L}(\mathcal{A}^{\text{det}}) \subseteq \mathcal{L}(\mathcal{A})$

Por lo tanto: Para todo $i \leq n$ y para todo $p \in S_i$, existe una ejecución:

$$\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \ldots \xrightarrow{a_i} p_i = p$$

1. $p_0 \in I$.
2. $(p_j, a_{j+1}, p_{j+1}) \in \Delta \quad \forall\, j \in \{0, \ldots, i-1\}$.

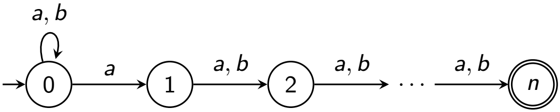
Como $S_n \cap F \neq \varnothing$, (¿por qué?)
para $p \in S_n \cap F$ existe una ejecución de acept. de $\mathcal{A}$ sobre w.

Por lo tanto, $w \in \mathcal{L}(\mathcal{A})$. ■

¿cuál es la ventaja de los autómatas no-deterministas?

Ventajas

1. Su representación es más sencilla para algunos lenguajes.
2. Son **exponencialmente** más compactos.



- ¿cuántos estados tiene la determinización?
- ¿es posible hacerlo con menos estados?

Demostración: ejercicio.