

Autómatas Finitos No deterministas

Matemáticas Computacionales
(TC2020)

M.C. Xavier Sánchez Díaz

sax@itesm.mx



Tabla de contenidos

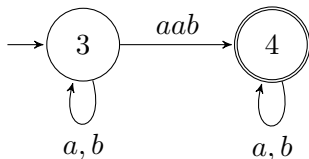
1 AFNs: las diferencias

2 Diseño de AFNs

Palabras como transiciones

AFNs: las diferencias

Las **transiciones** se etiquetan con **palabras**, no con símbolos del alfabeto.



¿Cuál es su definición formal?

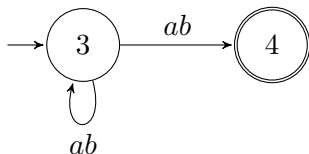
$$M = (\{3, 4\}, \{a, b\}, \\ \{(3, a, 3), (3, b, 3), (3, aab, 4), (4, a, 4), (4, b, 4)\}, \\ 3, \{4\})$$

Transiciones repetidas

AFNs: las diferencias

Pueden existir **varias transiciones con la misma palabra** a partir de un mismo estado.

Un estado puede no tener transiciones.

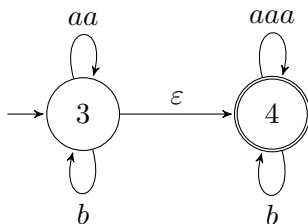


$$M = (\{3, 4\}, \{a, b\}, \{(3, ab, 3), (3, ab, 4)\}, 3, \{4\})$$

Transiciones vacías

AFNs: las diferencias

Se puede pasar de un estado a otro **sin consumir caracteres** de la entrada.



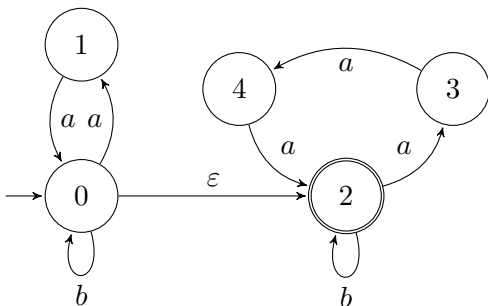
$$M = (\{3, 4\}, \{a, b\}, \\ \{(3, aa, 3), (3, b, 3), (3, \varepsilon, 4), (4, aaa, 4), (4, b, 4)\}, \\ 3, \{4\})$$

Aceptación de palabras

AFNs: las diferencias

Una palabra es **aceptada** por un AFN si existe una secuencia de transiciones a partir del estado inicial que la acepte:

- Que la consuma completamente.
- Que termine en un estado final.



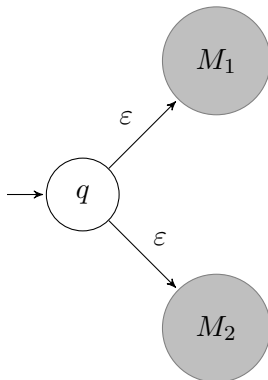
¿Acepta *aaaaa*? ¿Qué tal *abba*?

Diseño por unión

Diseño de AFNs

Definición

Sean M_1 y M_2 autómatas finitos no deterministas. Si M_1 y M_2 aceptan L_1 y L_2 respectivamente, la **combinación** acepta $L_1 \cup L_2$.

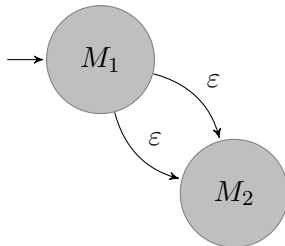


Diseño por concatenación

Diseño de AFNs

Definición

Sean M_1 y M_2 autómatas finitos no deterministas. Si M_1 y M_2 aceptan L_1 y L_2 respectivamente, la **concatenación** acepta L_1L_2 .



Diseño por intersección

Diseño de AFNs

Definición

Sean M_1 y M_2 autómatas finitos no deterministas. Si M_1 y M_2 aceptan L_1 y L_2 respectivamente, la **intersección**:

$$L_1 \cap L_2 \equiv (L_1^c \cup L_2^c)^c$$

¿La intersección?