

# Conversión de AFNs a AFDs

Matemáticas Computacionales  
(TC2020)

M.C. Xavier Sánchez Díaz  
mail@tec.mx



# Tabla de contenidos

1 AFNs versus AFDs

2 Conversión de AFN a AFD

# ¿Por qué convertir?

AFNs versus AFDs

Una máquina **no determinista** hace que el **diseño** sea más sencillo.

Una máquina **determinista** hace que la **implementación** sea más sencilla.

La realidad es que un lenguaje es aceptado por una **máquina determinista** **si y solo si** es aceptado por una **máquina no determinista**.

# ¿Por qué convertir?

AFNs versus AFDs

Una máquina **no determinista** hace que el **diseño** sea más sencillo.

Una máquina **determinista** hace que la **implementación** sea más sencilla.

La realidad es que un lenguaje es aceptado por una **máquina determinista** **si y solo si** es aceptado por una **máquina no determinista**.

# ¿Por qué convertir?

AFNs versus AFDs

Una máquina **no determinista** hace que el **diseño** sea más sencillo.

Una máquina **determinista** hace que la **implementación** sea más sencilla.

La realidad es que un lenguaje es aceptado por una **máquina determinista** **si y solo si** es aceptado por una **máquina no determinista**.

# Equivalencia

## AFNs versus AFDs

### Equivalencia de AFDs y AFNs

Sea  $N = (Q, \Sigma, \delta, q, F)$  un autómata no determinista. Existe un autómata determinista  $M$  tal que  $L(M) = L(N)$ .

El estado en el que se encuentra  $M$  después de haber leído la parte inicial de una palabra corresponde exactamente al **conjunto de todos los estados** que  $N$  puede alcanzar tras haber leído la misma parte de la palabra.

# Equivalencia

## AFNs versus AFDs

### Equivalencia de AFDs y AFNs

Sea  $N = (Q, \Sigma, \delta, q, F)$  un autómata no determinista. Existe un autómata determinista  $M$  tal que  $L(M) = L(N)$ .

El estado en el que se encuentra  $M$  después de haber leído la parte inicial de una palabra corresponde exactamente al **conjunto de todos los estados** que  $N$  puede alcanzar tras haber leído la misma parte de la palabra.

# Equivalencia

## AFNs versus AFDs

$$N = (Q, \Sigma, \delta, q, F)$$

$$M = (Q', \Sigma, \delta', q', F')$$

¿Cómo pasamos de un AFN a un AFD?



# Equivalencia

## AFNs versus AFDs

$$N = (Q, \Sigma, \delta, q, F)$$

$$M = (Q', \Sigma, \delta', q', F')$$

¿Cómo pasamos de un AFN a un AFD?

# Equivalencia

## AFNs versus AFDs

$$N = (Q, \Sigma, \delta, q, F)$$

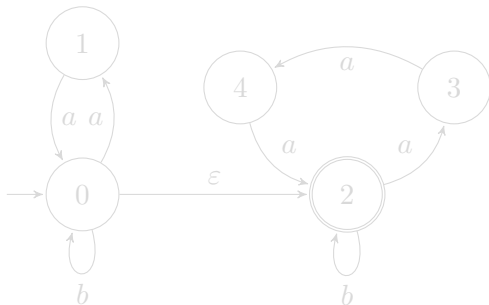
$$M = (Q', \Sigma, \delta', q', F')$$

¿Cómo pasamos de un AFN a un AFD?

# Estados de estados

## Conversión de AFN a AFD

El AFN **acepta** una palabra si existe al menos un estado final dentro del conjunto de estados donde la secuencia termina.



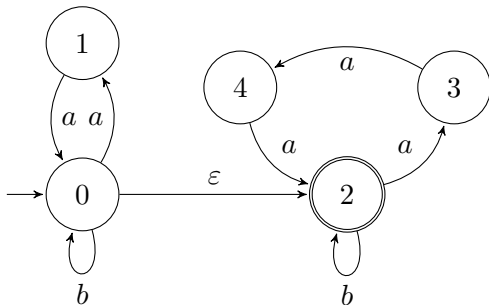
La palabra  $aa$  termina en  $\{0, 2, 4\}$ .

Podemos pensar entonces que  $\{0, 2, 4\}$  tendría que ser un **estado final** en el AFD equivalente.

# Estados de estados

## Conversión de AFN a AFD

El AFN **acepta** una palabra si existe al menos un estado final dentro del conjunto de estados donde la secuencia termina.



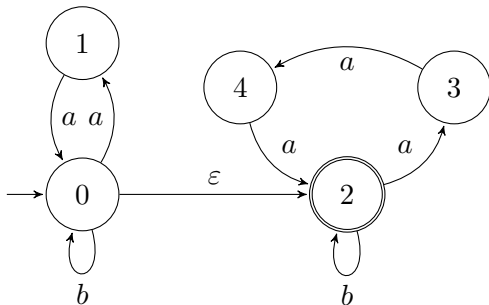
La palabra *aa* termina en  $\{0, 2, 4\}$ .

Podemos pensar entonces que  $\{0, 2, 4\}$  tendría que ser un **estado final** en el AFD equivalente.

# Estados de estados

## Conversión de AFN a AFD

El AFN **acepta** una palabra si existe al menos un estado final dentro del conjunto de estados donde la secuencia termina.



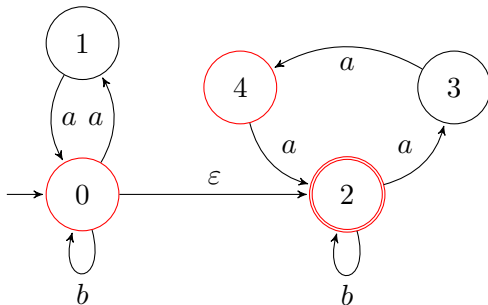
La palabra  $aa$  termina en  $\{0, 2, 4\}$ .

Podemos pensar entonces que  $\{0, 2, 4\}$  tendría que ser un **estado final** en el AFD equivalente.

# Estados de estados

## Conversión de AFN a AFD

El AFN **acepta** una palabra si existe al menos un estado final dentro del conjunto de estados donde la secuencia termina.

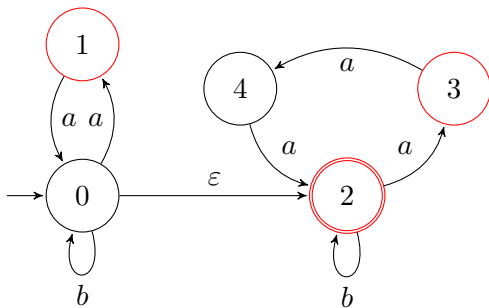


La palabra  $aa$  termina en  $\{0, 2, 4\}$ .

Podemos pensar entonces que  $\{0, 2, 4\}$  tendría que ser un **estado final** en el AFD equivalente.

# Estados de estados

## Conversión de AFN a AFD

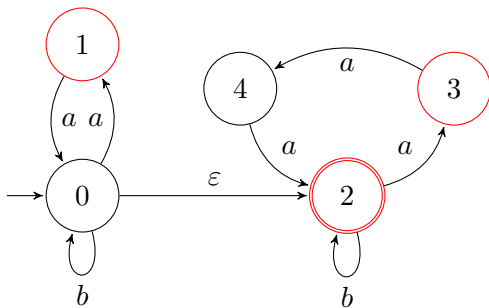


La palabra *abbaa* termina en  $\{1, 2, 3\}$ .

Podemos pensar entonces que  $\{1, 2, 3\}$  también tendría que ser un **estado final** en el AFD equivalente.

# Estados de estados

## Conversión de AFN a AFD



La palabra *abbaa* termina en  $\{1, 2, 3\}$ .

Podemos pensar entonces que  $\{1, 2, 3\}$  también tendría que ser un **estado final** en el AFD equivalente.



# Conjunto de estados

## Conversión de AFN a AFD

Significa que el AFD equivalente,  $M$ , tiene que tener **estados finales** de conjuntos de estados donde se acepte la palabra en el AFN.

Por tanto,  $Q' = \wp(Q)$ : el conjunto de estados de  $M$  es igual al conjunto potencia del conjunto de estados de  $N$ .

# Conjunto de estados

## Conversión de AFN a AFD

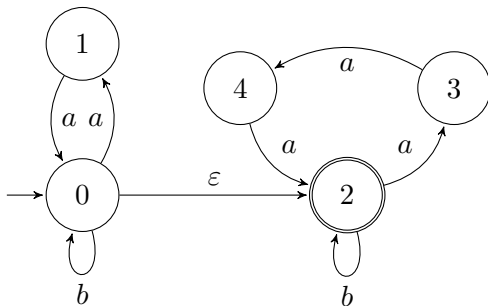
Significa que el AFD equivalente,  $M$ , tiene que tener **estados finales** de conjuntos de estados donde se acepte la palabra en el AFN.

Por tanto,  $Q' = \wp(Q)$ : el conjunto de estados de  $M$  es igual al conjunto potencia del conjunto de estados de  $N$ .

# Cerradura de vacío ( $\varepsilon$ )

## Conversión de AFN a AFD

Dado a que existe una transición  $\varepsilon$  entre 0 y 2 en  $N$ , claramente el estado inicial en  $M$  debe considerar  $\{0, 2\}$ .

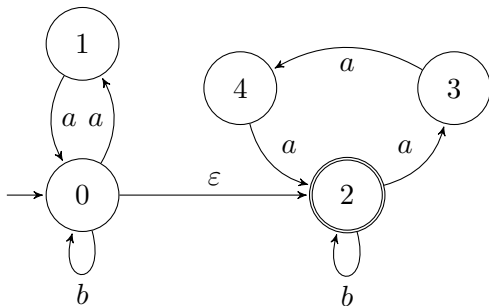


Para eso, hay que enfocarnos en la cerradura- $\varepsilon$ ,  $C_\varepsilon(r)$ , donde  $r$  es cualquier estado del AFN  $N$ .

# Cerradura de vacío ( $\varepsilon$ )

## Conversión de AFN a AFD

Dado a que existe una transición  $\varepsilon$  entre 0 y 2 en  $N$ , claramente el estado inicial en  $M$  debe considerar  $\{0, 2\}$ .



Para eso, hay que enfocarnos en la cerradura- $\varepsilon$ ,  $C_\varepsilon(r)$ , donde  $r$  es cualquier estado del AFN  $N$ .

# Cerradura de vacío ( $\varepsilon$ )

Conversión de AFN a AFD

## Definición

Para cada estado  $r$  del AFN  $N$ , la **cerradura de vacío**, representada con  $C_\varepsilon(r)$ , se define como el conjunto de todos los estados de  $N$  que pueden alcanzarse desde  $r$ , haciendo cero o más transiciones  $\varepsilon$ .

## Equivalencia en AFDs

Para cada estado  $R$  del AFD  $M$  (es decir,  $R \subseteq Q$ ):

$$C_\varepsilon(R) = \bigcup_{r \in R} C_\varepsilon(r)$$

# Cerradura de vacío ( $\varepsilon$ )

Conversión de AFN a AFD

## Definición

Para cada estado  $r$  del AFN  $N$ , la **cerradura de vacío**, representada con  $C_\varepsilon(r)$ , se define como el conjunto de todos los estados de  $N$  que pueden alcanzarse desde  $r$ , haciendo cero o más transiciones  $\varepsilon$ .

## Equivalencia en AFDs

Para cada estado  $R$  del AFD  $M$  (es decir,  $R \subseteq Q$ ):

$$C_\varepsilon(R) = \bigcup_{r \in R} C_\varepsilon(r)$$

# El estado inicial

## Conversión de AFN a AFD

El **estado inicial**  $q'$  en un AFD  $M$ , antes de leer cualquier símbolo 'real' de  $\Sigma$ , hace cero o más transiciones  $\varepsilon$ .

Al momento de que  $N$  lee el primer símbolo de  $\Sigma$ , puede estar en cualquier estado de  $C_\varepsilon(q)$ . Por tanto:

$$q' = C_\varepsilon(q) = C_\varepsilon(\{q\})$$

# El estado inicial

## Conversión de AFN a AFD

El **estado inicial**  $q'$  en un AFD  $M$ , antes de leer cualquier símbolo 'real' de  $\Sigma$ , hace cero o más transiciones  $\varepsilon$ .

Al momento de que  $N$  lee el primer símbolo de  $\Sigma$ , puede estar en cualquier estado de  $C_\varepsilon(q)$ . Por tanto:

$$q' = C_\varepsilon(q) = C_\varepsilon(\{q\})$$



# El conjunto de estados finales

## Conversión de AFN a AFD

El conjunto de **estados finales**  $F'$  del AFD  $M$ , es igual al conjunto de **todos** los elementos  $R$  de  $Q'$  que tienen la propiedad de que  $R$  contiene al menos un estado final del AFN  $N$ , es decir:

$$F' = \{R \in Q' : R \cap F \neq \emptyset\}$$

# El conjunto de estados finales

## Conversión de AFN a AFD

El conjunto de **estados finales**  $F'$  del AFD  $M$ , es igual al conjunto de **todos** los elementos  $R$  de  $Q'$  que tienen la propiedad de que  $R$  contiene al menos un estado final del AFN  $N$ , es decir:

$$F' = \{R \in Q' : R \cap F \neq \emptyset\}$$

# La función de transición

## Conversión de AFN a AFD

Pensemos que el AFD  $M$  está en el estado  $R$  y que recibe el símbolo  $a$ . En este momento, el AFN  $N$  habría estado en **cualquier** estado  $r$  que esté en  $R$ .

Al leer el símbolo  $a$ , la máquina  $N$  puede cambiar hacia **cualquier** estado de  $\delta(r, a)$ , y luego hacer cero o más transiciones- $\epsilon$ . Por lo que, para cada  $R \in Q'$ , y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_{\epsilon}(\delta(r, a))$$

# La función de transición

## Conversión de AFN a AFD

Pensemos que el AFD  $M$  está en el estado  $R$  y que recibe el símbolo  $a$ . En este momento, el AFN  $N$  habría estado en **cualquier** estado  $r$  que esté en  $R$ .

Al leer el símbolo  $a$ , la máquina  $N$  puede cambiar hacia **cualquier** estado de  $\delta(r, a)$ , y luego hacer cero o más transiciones- $\epsilon$ . Por lo que, para cada  $R \in Q'$ , y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_{\epsilon}(\delta(r, a))$$

# La función de transición

## Conversión de AFN a AFD

Pensemos que el AFD  $M$  está en el estado  $R$  y que recibe el símbolo  $a$ . En este momento, el AFN  $N$  habría estado en **cualquier** estado  $r$  que esté en  $R$ .

Al leer el símbolo  $a$ , la máquina  $N$  puede cambiar hacia **cualquier** estado de  $\delta(r, a)$ , y luego hacer cero o más transiciones- $\epsilon$ . Por lo que, para cada  $R \in Q'$ , y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_{\epsilon}(\delta(r, a))$$

# Resumen

## Conversión de AFN a AFD

Un autómata **finito no determinista**  $N = (Q, \Sigma, \delta, q, F)$  es convertido al autómata **finito determinista**  $M = (Q', \Sigma, \delta', q', F')$ , donde:

- $Q' = \wp(Q)$ ,
- $q' = C_\varepsilon(\{q\})$
- $F' = \{R \in Q' : R \cap F \neq \emptyset\}$
- $\delta' : Q' \times \Sigma \rightarrow Q'$ , donde para cada  $R \in Q'$  y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_\varepsilon(\delta(r, a))$$

¿Qué hace falta?

Hay que eliminar transiciones de varios símbolos, partiendo en una secuencia de estados.

# Resumen

## Conversión de AFN a AFD

Un autómata **finito no determinista**  $N = (Q, \Sigma, \delta, q, F)$  es convertido al autómata **finito determinista**  $M = (Q', \Sigma, \delta', q', F')$ , donde:

- $Q' = \wp(Q)$ ,
- $q' = C_\varepsilon(\{q\})$
- $F' = \{R \in Q' : R \cap F \neq \emptyset\}$
- $\delta' : Q' \times \Sigma \rightarrow Q'$ , donde para cada  $R \in Q'$  y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_\varepsilon(\delta(r, a))$$

¿Qué hace falta?

Hay que eliminar transiciones de varios símbolos, partiendo en una secuencia de estados.

# Resumen

## Conversión de AFN a AFD

Un autómata **finito no determinista**  $N = (Q, \Sigma, \delta, q, F)$  es convertido al autómata **finito determinista**  $M = (Q', \Sigma, \delta', q', F')$ , donde:

- $Q' = \wp(Q)$ ,
- $q' = C_\varepsilon(\{q\})$
- $F' = \{R \in Q' : R \cap F \neq \emptyset\}$
- $\delta' : Q' \times \Sigma \rightarrow Q'$ , donde para cada  $R \in Q'$  y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_\varepsilon(\delta(r, a))$$

¿Qué hace falta?

Hay que eliminar transiciones de varios símbolos, partiendo en una secuencia de estados.



# Resumen

## Conversión de AFN a AFD

Un autómata **finito no determinista**  $N = (Q, \Sigma, \delta, q, F)$  es convertido al autómata **finito determinista**  $M = (Q', \Sigma, \delta', q', F')$ , donde:

- $Q' = \wp(Q)$ ,
- $q' = C_\varepsilon(\{q\})$
- $F' = \{R \in Q' : R \cap F \neq \emptyset\}$
- $\delta' : Q' \times \Sigma \rightarrow Q'$ , donde para cada  $R \in Q'$  y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_\varepsilon(\delta(r, a))$$

¿Qué hace falta?

Hay que eliminar transiciones de varios símbolos, partiendo en una secuencia de estados.

# Resumen

## Conversión de AFN a AFD

Un autómata **finito no determinista**  $N = (Q, \Sigma, \delta, q, F)$  es convertido al autómata **finito determinista**  $M = (Q', \Sigma, \delta', q', F')$ , donde:

- $Q' = \wp(Q)$ ,
- $q' = C_\varepsilon(\{q\})$
- $F' = \{R \in Q' : R \cap F \neq \emptyset\}$
- $\delta' : Q' \times \Sigma \rightarrow Q'$ , donde para cada  $R \in Q'$  y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_\varepsilon(\delta(r, a))$$

¿Qué hace falta?

Hay que eliminar transiciones de varios símbolos, partiendo en una secuencia de estados.

# Resumen

## Conversión de AFN a AFD

Un autómata **finito no determinista**  $N = (Q, \Sigma, \delta, q, F)$  es convertido al autómata **finito determinista**  $M = (Q', \Sigma, \delta', q', F')$ , donde:

- $Q' = \wp(Q)$ ,
- $q' = C_\varepsilon(\{q\})$
- $F' = \{R \in Q' : R \cap F \neq \emptyset\}$
- $\delta' : Q' \times \Sigma \rightarrow Q'$ , donde para cada  $R \in Q'$  y para cada  $a \in \Sigma$ ,

$$\delta'(R, a) = \bigcup_{r \in R} C_\varepsilon(\delta(r, a))$$

¿Qué hace falta?

Hay que eliminar transiciones de varios símbolos, partiendo en una secuencia de estados.

# Resumen

## Conversión de AFN a AFD

Un autómata **finito no determinista**  $N = (Q, \Sigma, \delta, q, F)$  es convertido al autómata **finito determinista**  $M = (Q', \Sigma, \delta', q', F')$ , donde:

- $Q' = \wp(Q)$ ,
- $q' = C_\varepsilon(\{q\})$
- $F' = \{R \in Q' : R \cap F \neq \emptyset\}$
- $\delta' : Q' \times \Sigma \rightarrow Q'$ , donde para cada  $R \in Q'$  y para cada  $a \in \Sigma$ ,

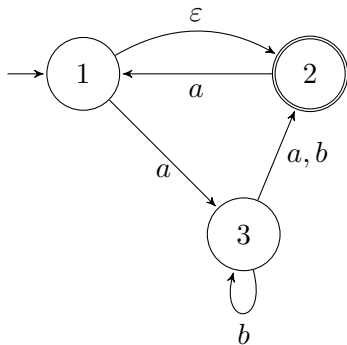
$$\delta'(R, a) = \bigcup_{r \in R} C_\varepsilon(\delta(r, a))$$

¿Qué hace falta?

Hay que eliminar transiciones de varios símbolos, partiendo en una secuencia de estados.

# Ejemplo

Conversión de AFN a AFD



# Ejemplo: complemento

## Conversión de AFN a AFD

Si tenemos que hacer un AF que acepte las palabras en  $\{a, b\}$  que **no contienen** ni  $abb$  ni  $aab$ ...

Se puede hacer primero un AFN que acepte las que contienen  $abb$  o bien  $aab$ .

Luego se convierte a AFD.

Y luego se encuentra la máquina complemento.

# Ejemplo: complemento

## Conversión de AFN a AFD

Si tenemos que hacer un AF que acepte las palabras en  $\{a, b\}$  que **no contienen** ni  $abb$  ni  $aab$ ...

Se puede hacer primero un AFN que acepte las que contienen  $abb$  o bien  $aab$ .

Luego se convierte a AFD.

Y luego se encuentra la máquina complemento.

# Ejemplo: complemento

## Conversión de AFN a AFD

Si tenemos que hacer un AF que acepte las palabras en  $\{a, b\}$  que **no contienen** ni  $abb$  ni  $aab$ ...

Se puede hacer primero un AFN que acepte las que contienen  $abb$  o bien  $aab$ .

Luego se convierte a AFD.

Y luego se encuentra la máquina complemento.



# Ejemplo: complemento

## Conversión de AFN a AFD

Si tenemos que hacer un AF que acepte las palabras en  $\{a, b\}$  que **no contienen** ni  $abb$  ni  $aab$ ...

Se puede hacer primero un AFN que acepte las que contienen  $abb$  o bien  $aab$ .

Luego se convierte a AFD.

Y luego se encuentra la máquina complemento.

# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.

# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.

# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.

# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.

# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.

# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.

# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.



# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.

# Ejemplo: intersección

## Conversión de AFN a AFD

Buscamos ahora un AFN que acepte palabras en  $\{a, b\}$  con número impar de  $bs$  y que contiene  $aab...$

¿Cuál es la solución?  $A \cap B = (A^c \cup B^c)^c$

- Hacer los AFN de los componentes.
- Convertirlos a AFDs.
- Complementarlos.
- Hacer la unión.
- Convertir el nuevo AFN a un AFD.
- Encontrar la máquina complemento.