Pruebe que

si $A \subseteq B \rightarrow A \cup B \subseteq B$

Paul Grimaldo

Demostración

$$A \subseteq B \longrightarrow A \cup B$$

Hipótesis (Suposición Verdadera)

Tesis lo que debo probar

Hacer mas simple la hipótesis

Suponyamos que $A \subseteq B$, esto es

$$x \in A \rightarrow x \in B$$

Probemos que $A \cup B \subseteq B$

$x \in A \cup B$ $\equiv$ Def. Union

$\equiv x \in A \vee x \in B$ $\equiv$ Hipótesis

$\equiv x \in A \vee x \in B$ $\equiv$ Idenpotencia

$x \in B$

$\therefore A \cup B \subseteq B$

Pruebe que $A \subseteq B \rightarrow A \cap B = A$

Supongamos que $A \subseteq B$

$x \in A \rightarrow x \in B$

Demostrar $A \cap B = A$ Def. Igualdad

$$\underbrace{A \cap B \subseteq A}_{\mathrm{I}} \wedge \underbrace{A \subseteq A \cap B}_{\mathrm{II}}$$

I) $A \cap B \subseteq A$

$x \in (A \cap B)$ Def. Inter.

$x \in A \wedge x \in B$ simpl..

$x \in A$ $\therefore A \cap B \subseteq A$

II $A \subseteq A \cap B$

$x \in A$ Por idempotencia

$x \in A \wedge x \in A$ Hipótesis

$x \in A \wedge x \in B$ def. inters.

$x \in (A \cap B)$

$\therefore \boxed{A \subseteq A \cap B}$ $\therefore A \subseteq B \rightarrow A \cap B = A$

Pruebe que

$$(A \cup B)^c = A^c \cap B^c$$

$(A \cup B)^c \subseteq A^c \cap B^c \wedge A^c \cap B^c \subseteq (A \cup B)^c$

$x \in (A \cup B)^c \subseteq A^c \cap B^c$ def. complemento

$x \in (A \cup B)^c$ def. c

$x \notin (A \cup B)$ Def $\notin$

$\sim (x \in A \vee x \in B)$ Demorgan

$x \notin A \wedge x \notin B$ def. Int.

$x \in A^c \cap B^c$

$\therefore (A \cup B)^c \subseteq A^c \cap B^c$

II $A^c \cap B^c \subseteq (A \cup B)^c$

$x \in (A^c \cap B^c)$ Def. int

$x \in A^c \wedge x \in B^c$ Def C

$x \notin A \wedge x \notin B$ Morgan

$\sim (x \in A \vee x \in B)$ def. Union

$\sim (x \in A \cup B)$ def. $\sim$

$x \notin A \cup B$

$x \in (A \cup B)^c$

$\therefore A^c \cap B^c \subseteq (A \cup B)^c$

$\therefore (A \cup B)^c = A^c \cap B^c$

$x \in (A \cup B)^c \equiv$

$x \notin (A \cup B)$ def. $\notin$

$\sim (x \in (A \cup B))$ def. $\cup$

$\sim (x \in A \vee x \in B)$ de Morgan

$\sim (x \in A) \wedge \sim (x \in B)$ def. $\sim$

$x \notin A \wedge x \notin B$ def. c

$x \in A^c \cap x \in B^c$ def. Int

$x \in (A^c \cap B)^c$

$\underline{\text{II}}$ $A^c \cap B^c \subseteq (A \cup B)^c$

$(x \in A^c \cap B^c)$ def. Int.

$\equiv x \in A^c \wedge x \in B^c$ def. C

$x \notin A \wedge x \notin B$ def. $\notin$

$\sim(x \in A) \wedge \sim(x \in B)$ de morgan

$\sim(x \in A \vee x \in B)$ def. $\cup$

$\sim(x \in A \cup B)$ def. $\sim$

$x \notin A \cup B$ def. $\notin$

$x \in (A \cup B)^c$

1.7 Concatenación de lenguajes

Def. Sean $A, B \subseteq \Sigma^*$ la concatenación anotado $A \cdot B$ se define $A \cdot B = \{ s \in \Sigma^* / s = x \cdot y, x \in A \wedge y \in B \}$

(i) Dados los lenguajes $(\Sigma = \{a, b, c\})$

$A = \{\lambda, ab, aab\}$, $B = \{cc, ca\}$

$A \cdot B = \{\lambda cc, \lambda ca, abcc, abca, aabcc, aabca\}$

$A \cdot B = \{cc, ca, abcc, abca\ aabcc, aabca\}$

(ii) $B \cdot A = \{cc\lambda, ccab, ccaab, ca\lambda, caab, caaab\}$

(i) Dados $P, Q \subseteq \Sigma^*$ calcule

$(P' \cdot Q) \cap P$ donde

$P = \{ab, cd\}$, $Q = \{bb, a\}$

$P' = \{ba, dc\}$

$P' \cdot Q = \{babb, baa, dcbb, dca\}$

$(P' \cdot Q) \cap P = \phi$

Para usar en demostraciones:

$x \in A \cdot B \equiv x = y \cdot z \,,\ y \in A \wedge z \in B$

(i) Dado $A, B, C \subseteq \Sigma^*$

pruebe que

$$A \cdot (B \cap C) \subseteq (A \cdot B) \cap (A \cdot C)$$

Sea $x \in A \cdot (B \cap C)$ Def. conc.

$x = z \cdot y \,,\ z \in A \wedge y \in (B \cap C)$ def. $\cap$

$x = z \cdot y \,,\ z \in A \wedge y \in B \wedge y \cap C$ Idempotencia ia

$x = z \cdot y \,,\ z \in A \wedge z \in A \wedge y \in B \wedge y A C$ pef.

$x = z y \,,\ z \in A \wedge y \in B \wedge z \in A \wedge y A C$ def. conc.

$x = z y \,,\ y \in (A \cdot B) \wedge x \in A \cdot B$ def $\cap$

$x = x \cdot y \,,\ x \in (A \cdot B \cap A C)$

$$\therefore A \cdot (B \cap C) = (A \cdot B) \cap (A \cdot C)$$

Si $\lambda \in Q$ entonces $P \subseteq P \cdot Q$

$(P, Q \subseteq \Sigma^*)$

supongamos que

$\lambda \in Q$.

Demostrar $P \subseteq P \cdot Q$

$x \in P$ def. λ

$x \cdot \lambda \in P$ Hipotesis

$x = x \cdot \lambda$, $x \in P \wedge \lambda \in Q$ def. conc.

$x = x \cdot \lambda$, $\underline{x \in P \cdot Q}$

$x \in P$, debo encontrar 2 cadenas talque $u, v \in \Sigma^*$ talque

$x = u \cdot v \Rightarrow x \in P \cdot Q$

$\downarrow P \quad \downarrow Q$

$\therefore \boxed{P \subseteq P \cdot Q}$ //

Si $B \subseteq C \rightarrow A \cdot B \subseteq A \cdot C$

Demostrar:

$x \in B \in C \Rightarrow x \in B \rightarrow x \in C$

$x \in A \cdot B$ Def. Conc.

$x = z \cdot y$, $z \in A \wedge y \in B$ Hipotesis

$x = z \cdot y$, $z \in A \wedge y \in C$ def. Conc.

$\underline{x \in A \cdot C}$

$\therefore B \subseteq C \rightarrow A \cdot B \subseteq A \cdot C$

1.8 La estrella de Kleene

Sobre la concatenación definimos la potencia de lenguajes.

Sea $L \subseteq \Sigma^*$ un lenguaje

$L^0 = \{\lambda\}$
$L^1 = L$
$L^2 = L \cdot L$
$L^3 = L \cdot L \cdot L$

$$\begin{cases} L^0 = \lambda \\ L^n = L^{n-1} \cdot L \quad , (n \geq 1) \end{cases}$$

$L = \{ab, c\}$

calcular $L^3 = L^2 \cdot L$

$L^3 = \{abab, abc, cab, cc\} \cdot L$

$L^3 = \{abab, abc, cab, cc\} \cdot \{ab, c\}$

$L^3 = \{ababab, ababc, abcab, abcc, cabab, cabc, ccab, ccc\}$

Definición

Se define como L^* como

$$L^* = L^0 \cup L^1 \cup L^2 \cdots \cup L^n \cup \ldots$$

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

Teorema 1.1

Sea $L \subseteq \Sigma^*$

(i) $\lambda \in L^*$

(ii) $L \subseteq L^*$

Teorema 1.2 (Teorema de la estrella de Kleene).

Lema 2.1 para todo $a \in \Sigma$

$$a' = a$$

/* este lema me dice que la inversa de una cadena de longitud 1 es la misma cadena */

Todos los teoremas y lemas pueden ser usados en los ejercicios.

<u>Demostración</u>

Demostrar $a' = a \quad \forall a \in \Sigma$

// (i) Partir de un miembro y llegar al otro
(ii) Desarrollar ambos miembros simultaneamente hasta encontrar una identidad.

$a' =$ Ele. neutro

$a' = (\lambda a)'$ Por def. de inversa

$(wa)' = a \cdot w'$

$a' = a \cdot \lambda'$ def. Inversa

$a' = a \cdot \lambda$ elemento neutro

$a' = a$

Definicios Alternativas

Muchas def. usan $w \cdot a$, pero es posible usar $a \cdot w$

Def. $\begin{cases} \lambda' = \lambda \\ (w \cdot a)' = a \cdot w' \end{cases}$, $a \in \Sigma$, $w \in \Sigma^*$

Def. Alterna $\begin{cases} \lambda' = \lambda \\ (a \cdot w)' = w' \cdot a \end{cases}$, $a \in \Sigma$, $w \in \Sigma^*$

2.3 Longitud de una cadena

si $\Sigma = \{a, b, c\}$

$w = aabc$, $|w| = 4$. //

$$\begin{cases} |\lambda| = 0 \\ |w \cdot a| = |w| + 1 \end{cases}, \quad a \in \Sigma, w \in \Sigma^*$$

Def. Alterna $\begin{cases} |\lambda| = 0 \\ |a \cdot w| = 1 + |w| \end{cases}$

lema 2.2 $\forall a, a \in \Sigma$

$|a| = 1$.

<u>Demostraciones</u>

$|a| = |\lambda a| = |\lambda| + 1 \quad 0 + 1 = 1$ //

2.4 Principio de la inducción

Sea $P[u]$ un esquema proposicional en Σ^*
$P[u]$ será valido si se prueba:

(i) $P[\lambda]$ es verdadero

(ii) $P[w]$ es verdadero, entonces $P[wa]$

__Nota:__ La segunda proposición con $P[a \cdot w]$

(ii) Si $P[w]$ entonces $P[a \cdot w]$ es verdadero.

Si $P[w]$ entonces $P[w \cdot a]$

$\downarrow$ Hipotesis inductiva

$\downarrow$ Tesis.

__Demostrar:__

$$|u'| = |u| \qquad \forall u \in \Sigma^*$$

Sea:

$$P[u] = \text{"} |u'| = |u| \text{"}$$

// Probar las 2 proposiciones

(i) $|\lambda'|$ Def. inversa

$= |\lambda|$

(ii) $P[w]$ es true $\quad P[w \cdot a]$

Supongamos que $P[w]$ es true

$P[w] = " |w'| = |w| "$

Probar que $P[w \cdot a]$ es verdadero

$$|(wa)'| = |wa|$$

$|(wa)'| =$ def. Inv.

$|aw'| =$ def.

$1+|w'| =$ def. long.

$|w'|+1 =$ ord.

$|w|+1 =$ Hip.

$|w \cdot a| =$ def. long.

$\therefore$ $P[wa]$ es verdadero$_{/\!/}$

Demostrar.

$$(u \cdot v)' = v' \cdot u' \quad , \quad u, v \in \Sigma^*$$

$P[u] =$ " $(u \cdot v)' = v' \cdot u'$ "

// Usar la cadena que esta mas a la derecha y usar $P[w \cdot a]$

// Si se usa la cadena mas a la izquierda probar $P[a \cdot w]$

$P[u] =$ " $(u \cdot v)' = v' u'$

Probar $P[a \cdot w]$

$P[v] =$ " $(u \cdot v)' = v' u'$

Probar $P[w \cdot a]$

(i) $P[\lambda] = (u \cdot \lambda)' =$ el. neutro

$u' =$ el. neutro

$\lambda u' =$ def. inv en $\lambda \Rightarrow \lambda' \cdot u'$

(ii) Supongamos que $P[w]$ es verdad.

Probar $P[w \cdot a]$

$(u \cdot w a)' =$ def. inv

$a \cdot (u \cdot w)' =$ hip

$a \cdot v' \cdot u' =$ prop. $a' = a$

$a' = a$

$\therefore$ $P[wa]$ es verdadera.

Pruebe que:

$$|u \cdot v| = |u| + |v|$$

sea $P[v] =$ " $|u \cdot v| = |u| + |v|$ "

Probemos (i) $P[\lambda]$

$P[\lambda] =$ " $|u \cdot \lambda| = |u| + |\lambda|$

$|u\lambda| =$ neutro.

$|u| + 0 =$ add.

$|u| + |\lambda|$ def $| \ |$ $P[wa] = |uwa| = |u| + |wa|$

(ii) si $P[w] \to P[wa]$ es verdadero.

Supongamos que $P[w] =$ " $|u \cdot w| = |u| + |w|$ " es verdadero

$|u(wa)| =$ aso. $|u| + |w| + 1$

$|(uw) \cdot a| =$ def long.

$|u \cdot w| + 1 =$ Hip

Recordar estas propiedades

$$|a'| = |a|$$
$$|a| = 1$$
$$|u'| = |u|$$
$$(u \cdot v)' = v' + u'$$
$$|u \cdot v) = |u| + |v|$$

① De el sgt paso a las expresiones

a) $|u \cdot a \cdot w| = | \; |u| + |a| + |w|$?

Intentando aplicar la def.

$= |(ua) \cdot w)$ asociando

$|(ua) \cdot w \quad =$

Tiene que estar a la derecha o a la iz.

b) $|u' \cdot a \cdot v)$

Sol.

$$|u' \cdot a \cdot v| = |(u' \cdot a) \cdot v| \qquad \text{def. inverso}$$
$$|(u \cdot a)' \cdot v)$$

// Hasta aquí Parcial

Demostración por RAA

Consiste en negar la tesis y buscar un absurdo o una contradicción.

$$x \in \emptyset$$

La hipótesis nunca se niega.

- Use el método por contradicción cuando vea que el conjunto $\emptyset$ está en la hipótesis y/o tesis.

Sea. $A \subseteq \Sigma^*$

pruebe que $\quad A \cdot \emptyset = \emptyset$ def =

Asumamos que $A \cdot \emptyset \neq \emptyset$

$$x \in (A \cdot \emptyset) \neq \emptyset \qquad \text{def. concatenación}$$

$$x = z \cdot y, \quad z \in A \wedge y \in \emptyset = \emptyset \qquad \text{Def. Absurdo } y \in \emptyset$$

$$\therefore A \cdot \emptyset = \emptyset$$

Pruebe que

si $A \subseteq \phi \rightarrow A = \phi$

Supongamos que

$$A \subseteq \phi \Rightarrow x \in A \rightarrow x \in \phi$$

Asumamos que $A \neq \phi$

$x \in A \quad = \quad$ Hip

$x \in \phi \quad\quad$ contradicción

$\therefore \quad A = \phi$ // Debe ser entonces que $A = \phi$ //

pruebe que

$$A \cap \phi = \phi$$

Demostración

Asumamos que $A \cap \phi \neq \phi$

//osea.

$x \in A \cap \phi$ def. $\cap$

$x \in A \wedge x \in \phi \rightarrow | \leftarrow$

Debe ser entonces que

$$\underline{A \cap \phi = \phi}$$

//

// Supongamos que $A \cap \phi \neq \phi \Rightarrow$ def. <u>int</u>

$\exists x / x \in A \wedge x \in \phi$ Simp. $\wedge$

① $x \in \phi$ def. ϕ

② $x \notin \phi$

de ① $\wedge$ ②

$x \in \phi \wedge x \notin \phi$ $\therefore A \cap \phi = \phi$

$x \in (A \cap B) \equiv x \in A \wedge x \in B$ def $\cap$

$x \in (A \cup B) \equiv x \in A \vee x \in B$ def. $\cup$

$x \in (A - B) \equiv x \in A \wedge x \notin B$ def $-$

$x \in A^c \equiv x \notin A$ def. C

$x \in (A \cdot B) \equiv x = y \cdot z \,;\, y \in A \wedge z \in B$ def. <u>concate</u>

$x \in P' \equiv x \notin P$; def inversa L

$x \in L^* \equiv x = x_1 \cdot x_2 \cdots x_n$; $x_i \in L$

Unidad 3 Módulos

Un módulo es una terna $\langle \Sigma, S, f \rangle$ donde

Σ = Alfabeto

S = Conjunto de estados

$f = S \times \Sigma \to S$

$S = \{ d, i, p \}$

$\Sigma = \{ r, v, a \}$

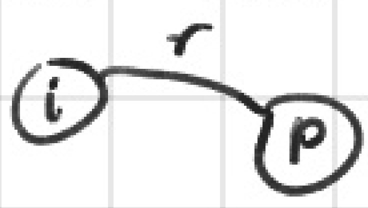
$f(i, r) = p$

Representación de un módulo.

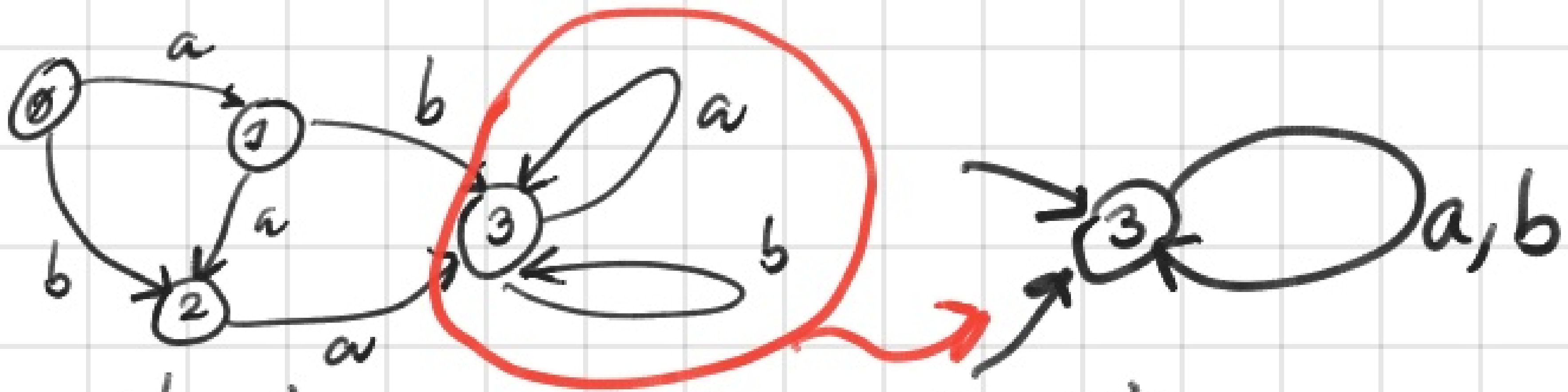
La representación más utilizada es el llamado grafo de estados, donde:

- Los vértices son los estados
- Las aristas representan una transición.

$f(i, r) = p$



(.) Dado el sgt grafo de estados



Escriba los componentes matemáticos

- **Solución**

$< \Sigma, S, f >$

$\Sigma = \{a, b\}$, $S = \{0, 1, 2, 3\}$

$f(0, a) = 1, f(0, b) = 2$

$f(1, a) = 2$, $f(1, b) = 3$

$f(2, a) = 3$, $f(2, b) = 2$

$f(3, a) = 3$, $f(3, b) = 3$

3.3 función de estado terminado $(\hat{f})$

Supongamos que (en el anterior modelo) estamos en el estado 0 y pulsamos: babb

¿En qué estado nos quedamos?

__Respuesta:__

$$1 \xrightarrow{b} 3 \xrightarrow{a} 3 \xrightarrow{b} 3 \xrightarrow{a} 3$$

" la máquina se queda en el estado 3 "

$$\hat{f}(1, \text{"babb"}) = \boxed{3} \,//$$

Con el módulo de la cinta, __calcule__:

$\hat{f}(p, \text{"rvva"})$ $\qquad S = \{d, i, p\}$

$\qquad\qquad\qquad\qquad \Sigma = \{r, v, a\}$

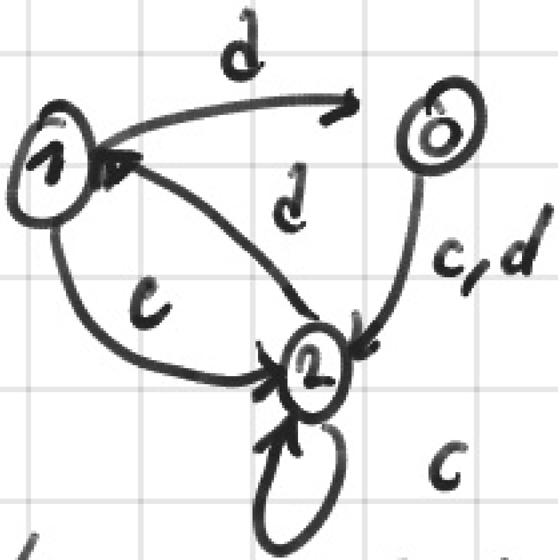
Definición de $\hat{f}$

Sea $M = \langle \Sigma, S, f \rangle$ un módulo definimos $\hat{f}$:

$$\underline{\hat{f} : S \times \Sigma^* \to S} \qquad \begin{cases} \hat{f}(s, \lambda) = s \\ \hat{f}(s, a \cdot \omega) = \hat{f}(f(s, a), \omega) \end{cases}$$

$$a \in \Sigma, \omega \in \Sigma^*$$

Dado el módulo



$\Sigma = \{c, d\}$

$S = \{0, 1, 2\}$

calcule por definición

$\hat{f}(1, \text{"}cdd\text{"}) =$

$\hat{f}(f(1, c), dd) = \hat{f}(2, dd) =$

$\hat{f}(f(2, d), d) = \hat{f}(1, d) =$

$\hat{f}(f(1, d), \lambda) =$

$\hat{f}(\emptyset, \lambda) = \emptyset$ //

Lema 3.1

sea $M = \langle \Sigma, \delta, f \rangle$

$$\hat{f}(s,a) = f(s,a)$$

Demostración:

$$\hat{f}(s,a)$$

Teorema 3.2

Sea $M = \langle \Sigma, s, f \rangle$ un módulo

$$\hat{f}(s, u \cdot v) = \hat{f}(\hat{f}(s,u), v) \ \forall u,v \in \Sigma^*$$

Demostración:

definir $P[u]$ o $P[v]$

Tomando $P[u]$.

Sea $P[u] =$ " $\hat{f}(s, u v) = \hat{f}(\hat{f}(s,u), v)$ "

(i) probar $P[\lambda]$

$P[\lambda] =$ " $\hat{f}(s, \lambda v) = \hat{f}(\hat{f}(s,\lambda), v)$ "

$\hat{f}(s, \lambda v)$ $\equiv$ el. neutro

$\hat{f}(s, v)$ $\equiv$ def. $\hat{f}$ hacia atrás.

$\hat{f}(\hat{f}(s,\lambda), v)$ //

$\therefore P[\lambda]$ es true

si $P[w]$ es verdad entonces $P[aw]$

Supongamos que $P[w]$ es verdad.

$$P[w] = \hat{f}(s, w v) = \hat{f}(\hat{f}(s,w), v)$$

Probar $P[aw]$

$$P[aw] = \hat{f}(s, aw \cdot v) = \hat{f}(\hat{f}(s,aw), v)$$

$\hat{f}(s, aw \cdot v)$ def $\hat{f}$

$\hat{f}(f(s,a), w \cdot v)$ sea $s_1 = f(s,a)$

$\hat{f}(s_1, w \cdot v)$ H.I

$\hat{f}(\hat{f}(s_1, w), v)$

$\hat{f}(\hat{f}(f(s,a), w), v)$

segundo miembro

$\hat{f}(\hat{f}(s,aw), v)$ def. $\hat{f}$

$\hat{f}(\hat{f}(f(s,a),w),v)$ //

$\therefore$ $P[aw]$ es verdadero //

(c) probar que

si $\hat{f}(s, a \cdot a) = s \to \hat{f}(s, u a') = s$

Supongamos que $\underline{\hat{f}(s, a \cdot a)} = s$ es true $\forall a \in \Sigma$

Demostrar

$\hat{f}(s, u a') = s$

sea $P[u] = \hat{f}(s, u a') = s$

(i) probar $P[\lambda] = \hat{f}(s, \lambda \cdot \lambda') = s$

$\hat{f}(s, \lambda \lambda') =$ def '

$\hat{f}(s, \lambda \cdot \lambda)$ = el. neutro

$\hat{f}(s, \lambda) = s$ //

Supongamos que $P[w]$ es verdad.

$P[w] =$ " $\hat{f}(s, w w') = s$ "

Demostrar. $P[aw] = \hat{f}(s, aw \cdot (aw)') = s$

$\hat{f}(s, (a w) \cdot (aw)')$ def $\hat{f}$

$\hat{f}(f(s,a), w \cdot (aw)')$ sea $s_1 = f(s,a)$

$\hat{f}(s_1, w (aw)')$ def '

$\hat{f}(s_1, w \cdot w' a)$ por teorema

$\hat{f}(\hat{f}(s_1, w w'), a)$ HI

$\hat{f}(s_1, a)$ pero $s_1 = f(s,a)$

$= \hat{f}(f(s,a), a)$ def $\hat{f}$

$= \hat{f}(s, a \cdot a)$ HI.

$\underline{s}$ //

3.4 Alcanzabilidad

Definición

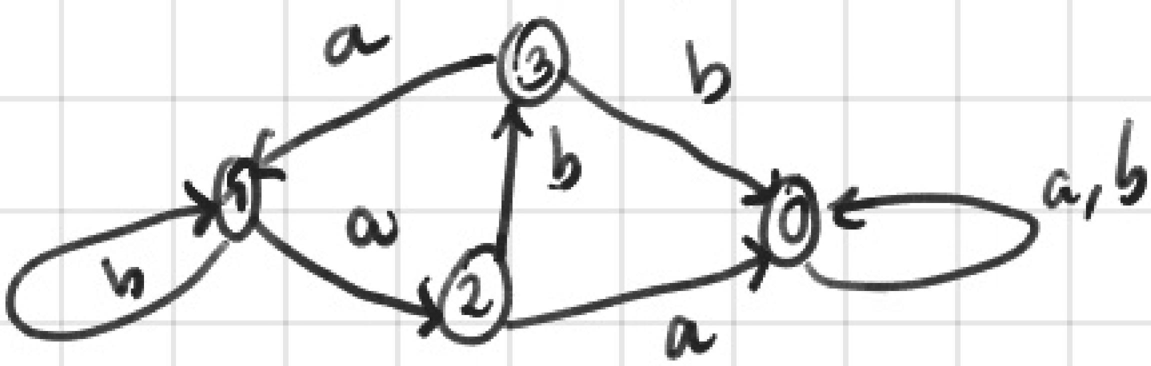
Sea $M = \langle \Sigma, S, f \rangle$ un módulo, decimos que $t \in S$ es alcanzable a partir de $s \in S$, anotado

$$s \xrightarrow{*} t$$

ssi existe $u \in \Sigma^*$

$$\hat{f}(s, u) = t$$

Considerando el módulo



¿ $3 \xrightarrow{*} 2$?

$$f(3, aa) = 2$$

(·) Demuestre que $\forall s \in S$

$$s \xrightarrow{*} s$$

// esto quiere decir que $\xrightarrow{*}$ es reflexiva

Sol.

La def. dice que:

$$q \xrightarrow{*} r = \exists \omega \in \Sigma^* / \hat{f}(q,\omega) = r$$

Existe por def. de $\hat{f}$

$\hat{f}(s,\lambda) = s$

se prueba que $s \xrightarrow{*} s$

Pruebe que:

si $s \xrightarrow{*} t$ y $t \xrightarrow{*} r \Rightarrow s \xrightarrow{*} r$

Supongamos: simpli.

$s \xrightarrow{*} t \wedge t \xrightarrow{*} r$ $\quad s \xrightarrow{*} t, t \xrightarrow{*} r$

$s \xrightarrow{*} t$ esto es $\exists \omega \in \Sigma^* / \hat{f}(s,\omega) = t$

$t \xrightarrow{*} r = \exists \omega \in \Sigma^* / \hat{f}(t,\omega) = r$

probemos que:

$s \xrightarrow{*} r$

Tomando en cuenta la cadena $\omega \cdot u$

$\hat{f}(s, \omega \cdot u)$ Teorema

$= \hat{f}(\hat{f}(s,\omega), u)$ HI

=

$= \hat{f}(t, u)$ HI

$\underbrace{= r}_{u}$

$\therefore\ s \xrightarrow{*} r$

Expresiones Regulares

Son expresiones que definen un lenguaje

4.1 Operadores de la ER

usan lo siguiente:

$*$ = 0 o más
$+$ = 1 o más
$|$ = Unión (se lee ó)

$a^+ = a^* \cdot a = a \cdot a^*$

El operador (|) formalmente se escribe como unión ($\cup$)

$$| = \cup$$

Si r es una ER
L(r) es el lenguaje generado.
(·) Especifique un lenguaje.

$L = \{\lambda, a, aa, aaa, ..\}$
usando una ER r
$L(r) = \{a^*\}$
(·) Especificar cadenas que se formen por
empieza con a y termina con 1 o mas b's

$L = \{ab^+\}$ // $a \cdot b^+$

(·) Empieza con [1 o más b y terminan
en a] o [emiezan con
1 o mas a y terminan en b]

$b^+a \cup a^+b$

Sea $\Sigma = \{a,b\}$ escribir una ER r
que especifique cualquier combinación de:

a) De longitud 0 o mayor
b) De longitud 1 o mayor

$$(a|b)^*$$

a) $(a|b)^*$

b) $(a|b)^+$

(·) sea $\Sigma = \{a,e,b,c,d\}$

Escriba una ER tal que reconozca todas cadenas
que terminan en vocal.

$$(a|e|b|c|d)^* \cdot (a|e)$$

(·) Dado $\Sigma = \{0, 1, 2, 3, 4\}$

Escribir una ER r que especifique los números naturales pares

$$(0|1|2|3|4)^* \cdot (0|2|4)$$

4.2 Lenguajes Regulares

Un lenguaje es regular ssi es generado por una $\underline{ER\ r}$

$$A \subseteq \Sigma^*$$

$$L(r) = A$$

(·) el lenguaje (conjunto de cadenas) especificado por una ER R

$L(R)$ es regular

(·) Si queremos mostrar que un lenguaje es regular debemos encontrar una expresión regular que lo genere

$L(r) = r$

1* En el anterior AFD, la cadena "bab", **NO** es reconocida por el AFD porque

$$\hat{f}(1, bab) = 0 \wedge 0 \notin F$$

↳ NO es un estado final.

Tambien λ no es reconocida por el AFD porque

$$\hat{f}(1, \lambda) = 1 \wedge 1 \notin F$$

¿Cuantas cadenas son reconocidas?
Infinitas.

Si todas las cadenas que reconoce el automata la introducimos a un conjunto a este conjunto se le llamara "lenguaje generado por el autómata"

Si A es el AFD entonces L(A) es el lenguaje generado por A.

Definimos

$M = \langle \Sigma, S, f, s_0, F \rangle$

un AFD definimos $L(M)$

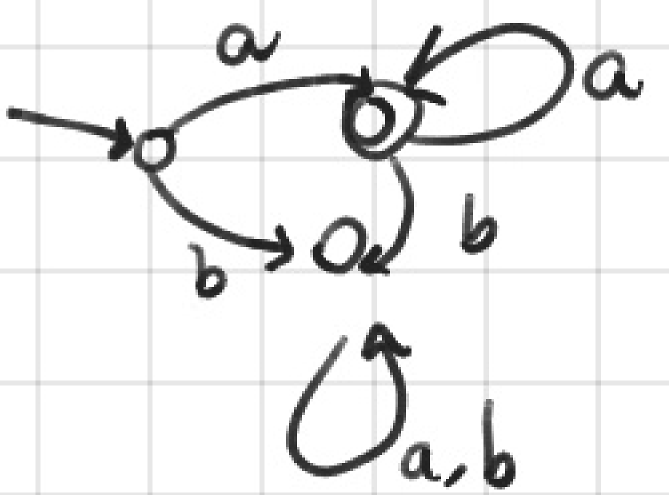
$$L(M) = \{ u \in \Sigma^* / \hat{f}(s_0, u) \in F \}$$

llamado lenguaje generado por M.

5.2 Ejercicios

(1.) Dibuje un AFD que reconozca una o más a'es

$L(A) = \{a, aa, aaa, \ldots\}$

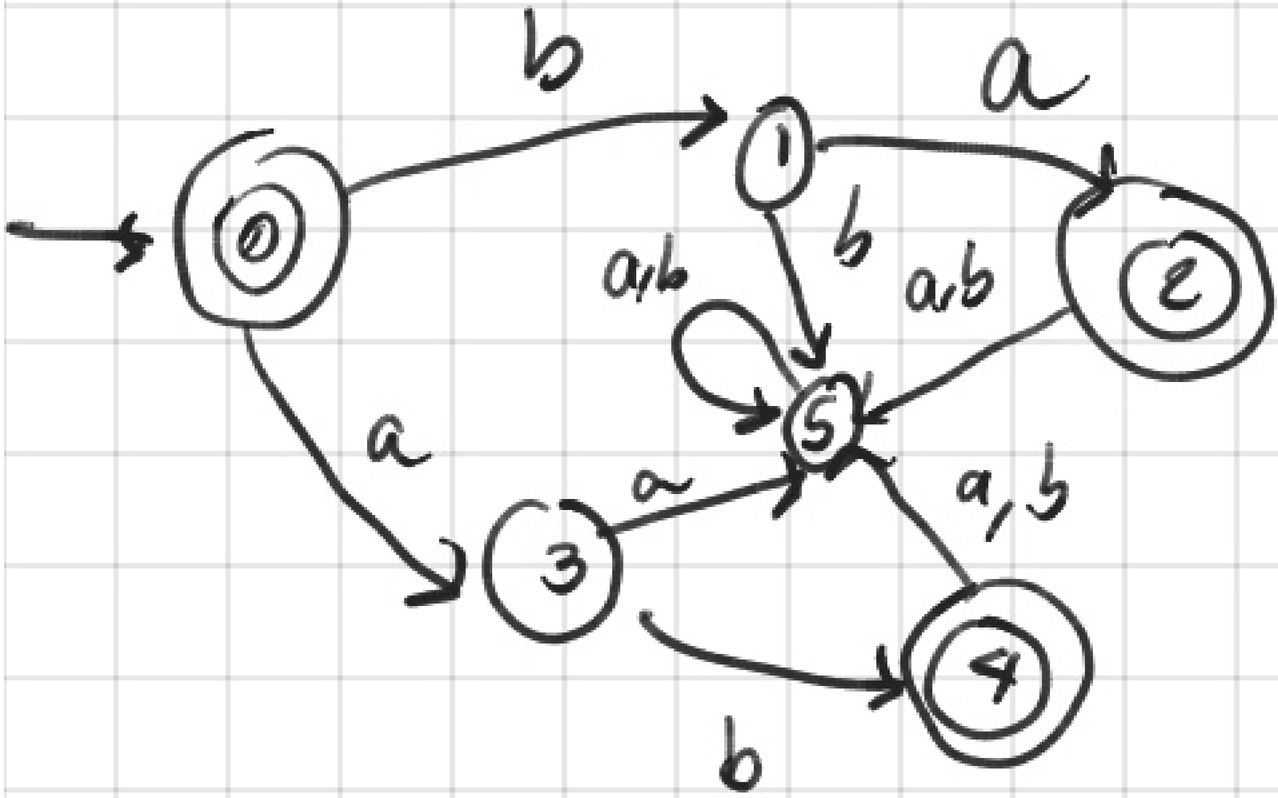


si desea reconocer λ haga que el estado inicial sea también final.

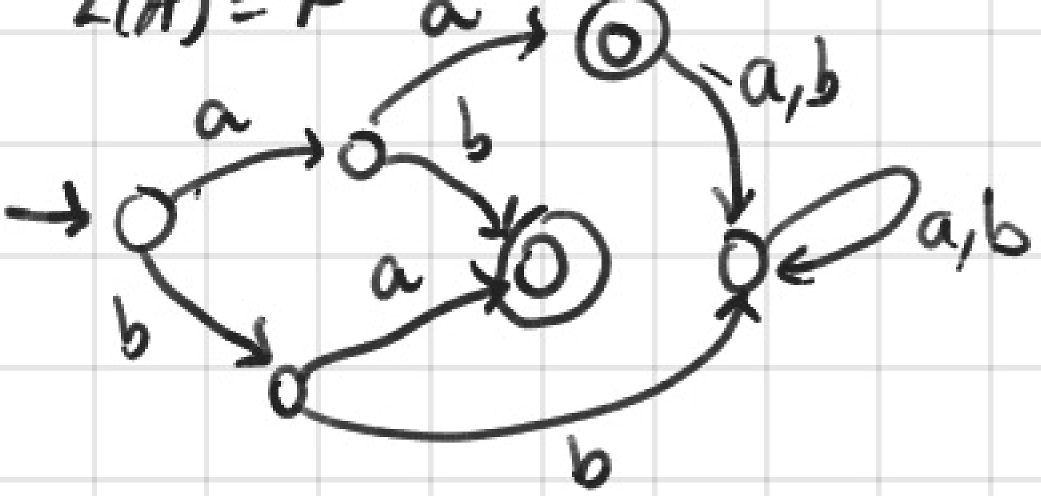
(1) Encuentre un AFD M tal que

$$L(M) = \{ \lambda, ba, ab \}$$

con $\Sigma = \{a, b\}$



sea $P \subseteq \Sigma^*$ con $\Sigma = \{a, b\}$
donde $P = \{ab, aa, ba\}$
encuentre un AFD, tal que
$L(A) = P$

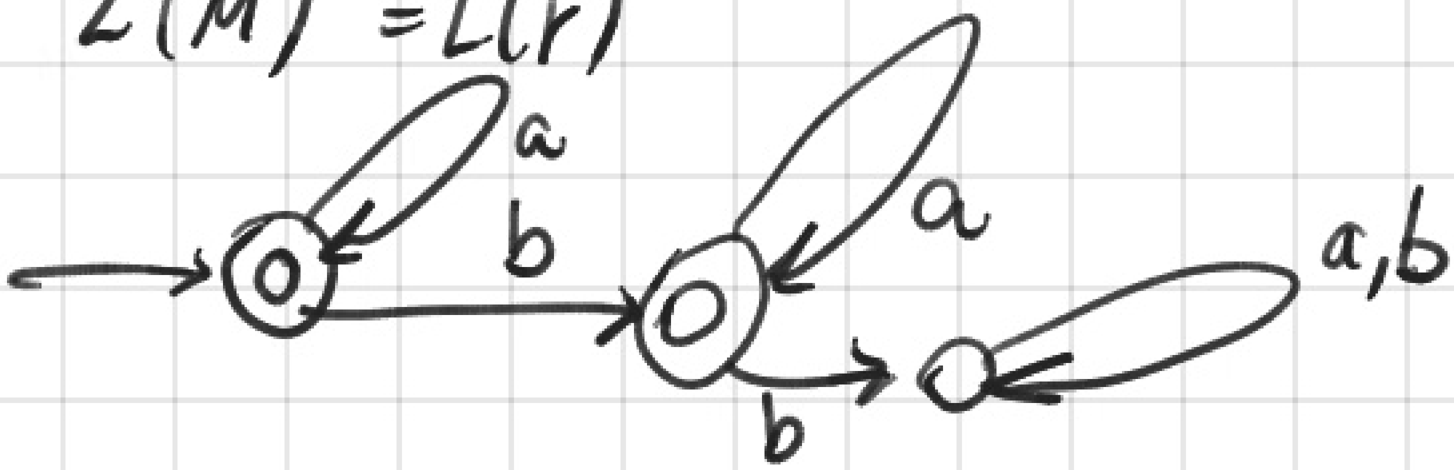


(.) dado la ER r

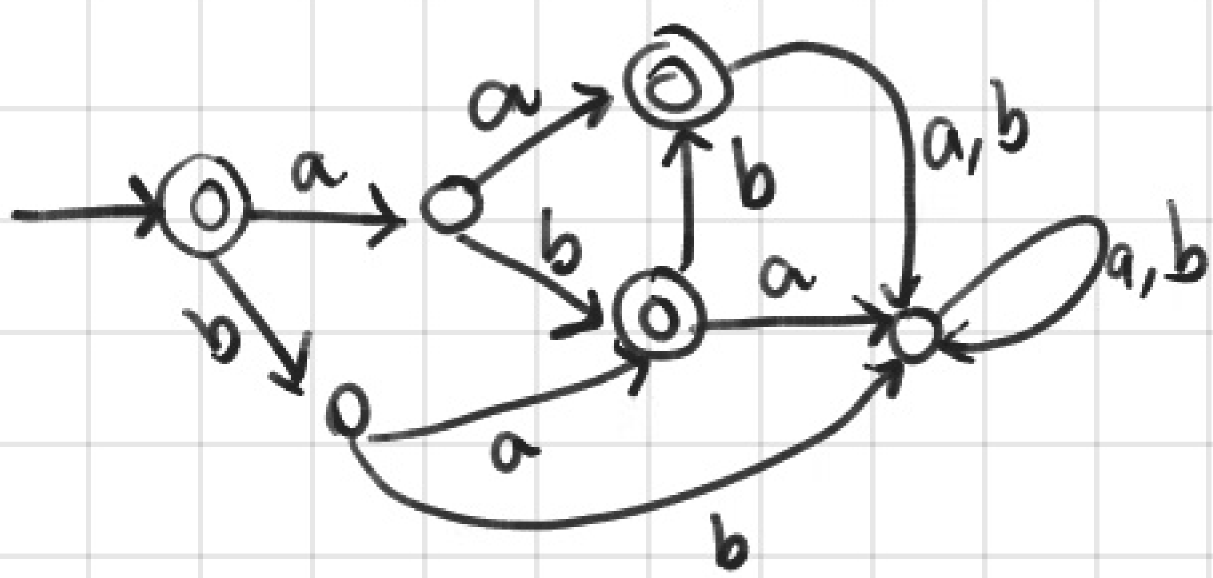
$r = a^* b a^*$

Encuentre un AFD M, tal que

$L(M) = L(r)$

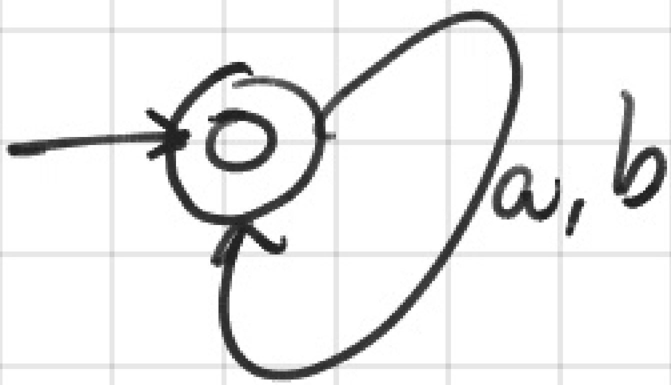


(.) dado $L = \{bab, ba, \lambda, ab, aa\}$
encuentre un AFD A tal que
$L(A) = L$ $\qquad \Sigma = \{a, b\}$



(.) Dibujar un AFD M
tal que

$$L(M) = \Sigma^*$$



5.3 Teorema Principal.
Sea A un AFD en lenguaje por A
es regular ($L(A)$ es un lenguaje regular)

- Es posible construir automatas que reconozcan el lenguaje generado por una ER

Ejercicio

sean $A = \langle \Sigma, S, f, s_0, F_A \rangle$ y $B = \langle \Sigma, S, f, s_0, F_B \rangle$
2 automatas que solo difieren en sus estados finales.

Pruebe que
si $F_A \subseteq F_B \longrightarrow L(A) = L(B)$

Solución

Supongamos que $F_A \subseteq F_B$

$$z \in F_A \Rightarrow z \in F_B$$

Probemos que
$L(A) = L(B)$

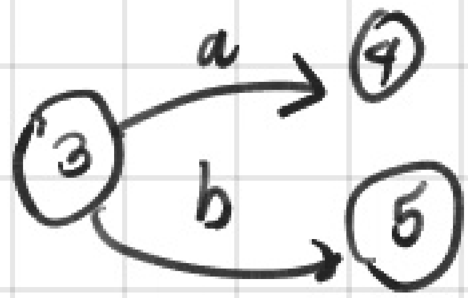
HI $\quad x \in L(A) = \hat{f}(s_0, x) = t \wedge t \in L(A)$

Sea $x \in L(A) \equiv \hat{f}(s_0, x) = t \wedge t \in F_A$ $\qquad$ por HI

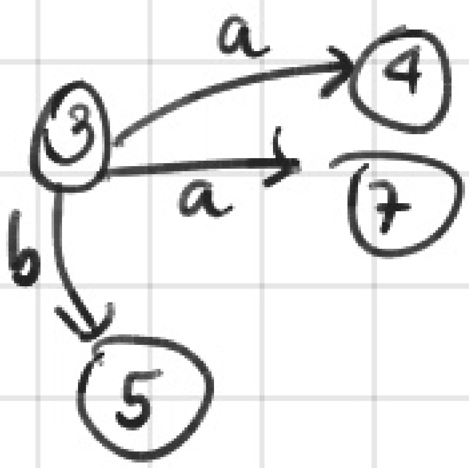
$\hat{f}(s_0, x) = t \wedge t \in F_B$

$\boxed{x \in L(B)}$ //

Unidad 6. Automatas finitos No deterministas



AFD

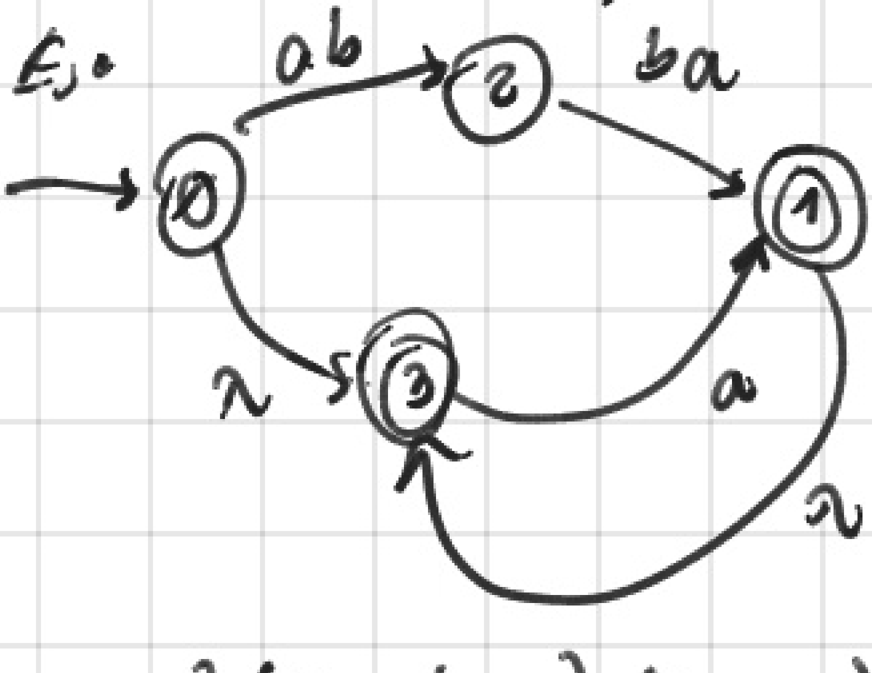


Un AFN es una quintupla $\langle \Sigma, S, \Delta, s_0, F \rangle$ donde

$\Delta \subseteq S \times \Sigma^* \times S$ es un conjunto de ternas

$(1, \text{"bab"}, 2)$

Ej.



$\Sigma = \{a, b\}$

$S = \{0, 1, 2, 3\}$

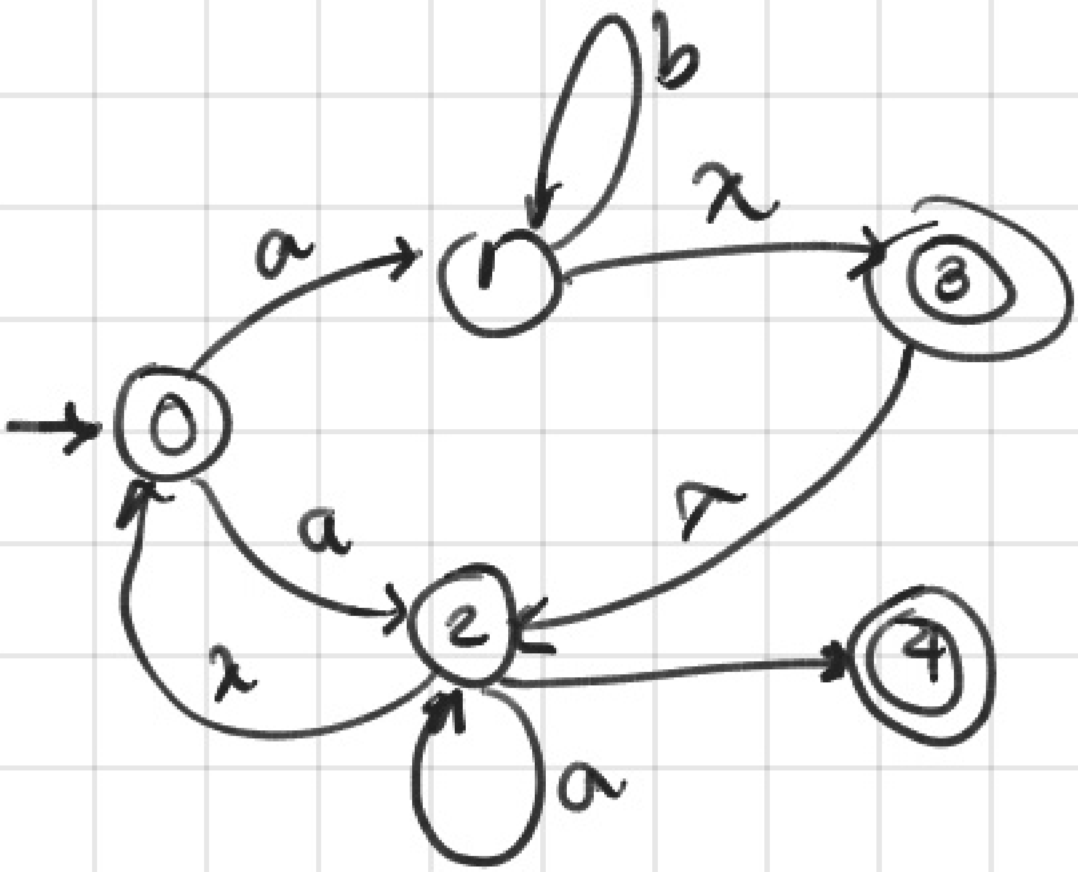
$s_0 = 0$

$F = \{1, 3\}$

$\Delta = \{(0, ab, 2), (0, \lambda, 3), (2, ba, 1), (3, a, 1)$
$(1, \lambda, 3)\}$

Automatas simples

Todo AFN puede reducirse a un AFN simple, por tanto, todos los resultados se expresan en AFN simples.
Es simple si cada una de sus aristas usa un char o λ.

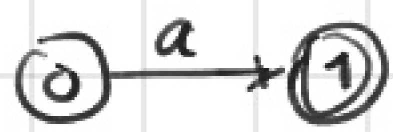


Paso de una ER a un AFN

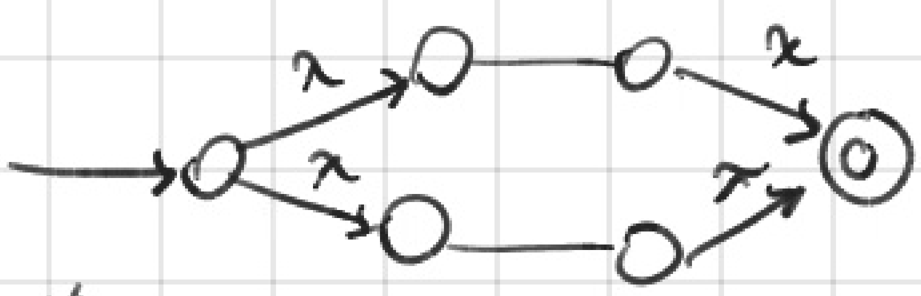
Sea r una ER. Se desea construir un AFN A / $L(r) = L(A)$

Algoritmo

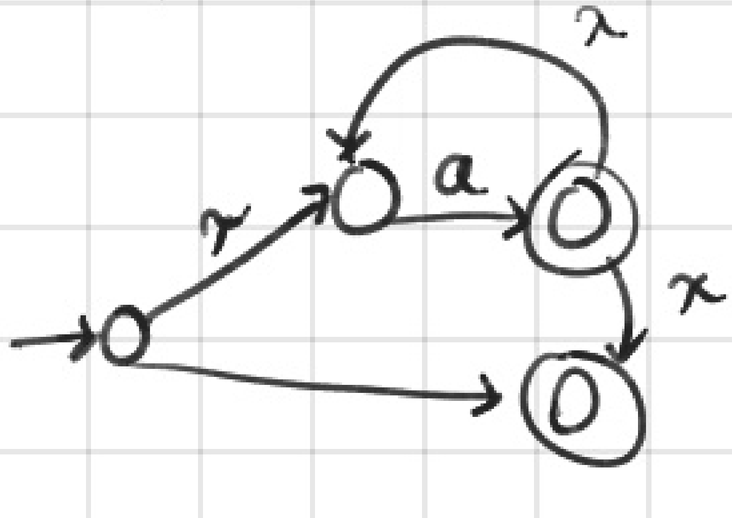
$\forall a \in \Sigma$ construir su AFN



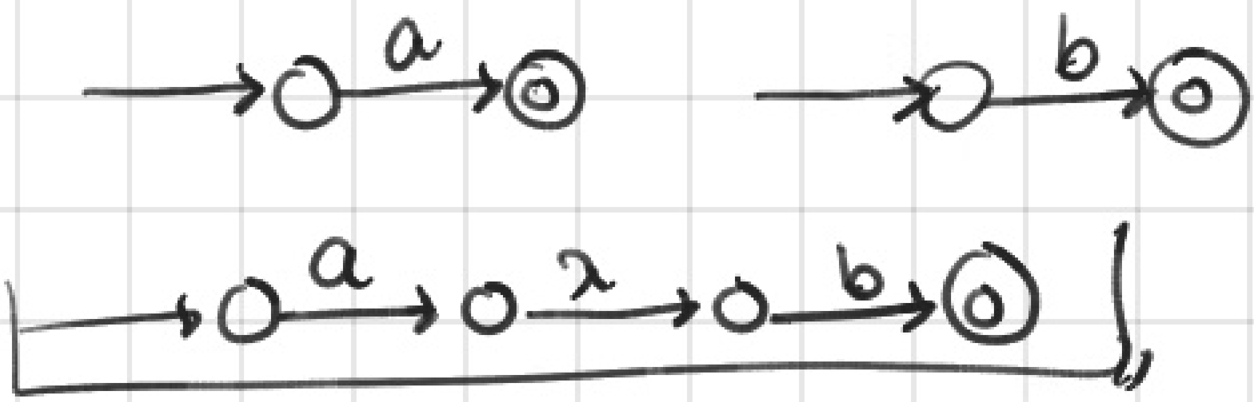
- Dados 2 ER r y s con sus AFN construir un AFN tal que reconoce $r|s$



- r^*



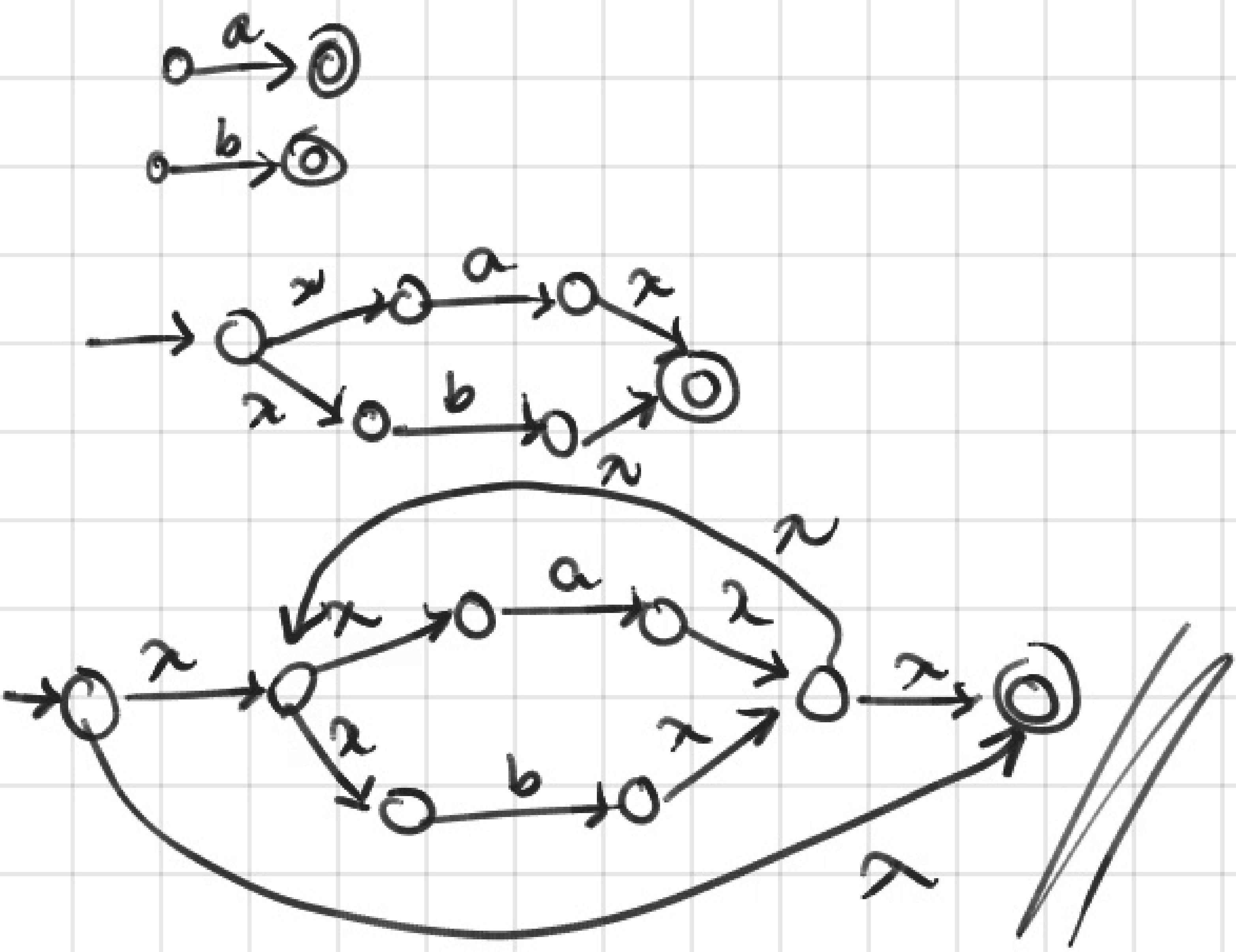
$r \cdot s$



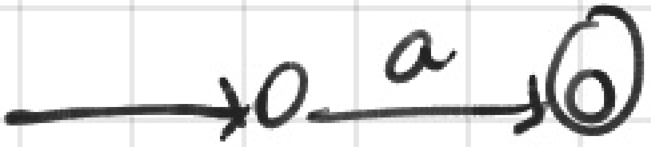
Pasar la sgt

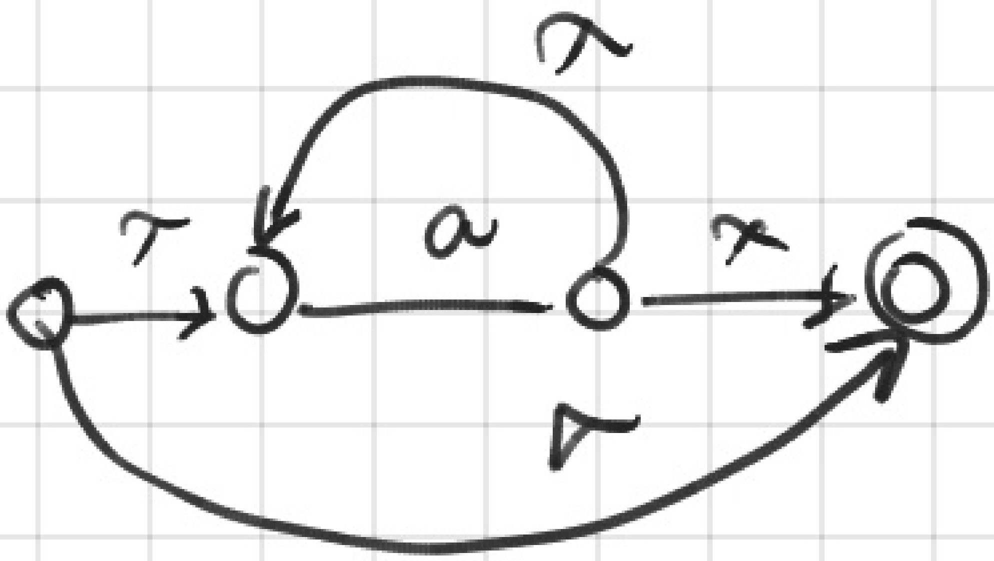
$r = (a/b)^*$

a un AFN



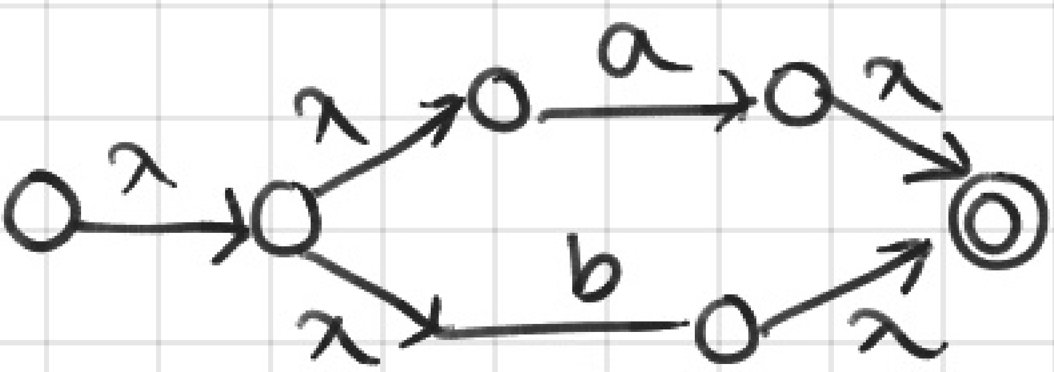
$r = a^*(b|a)b$



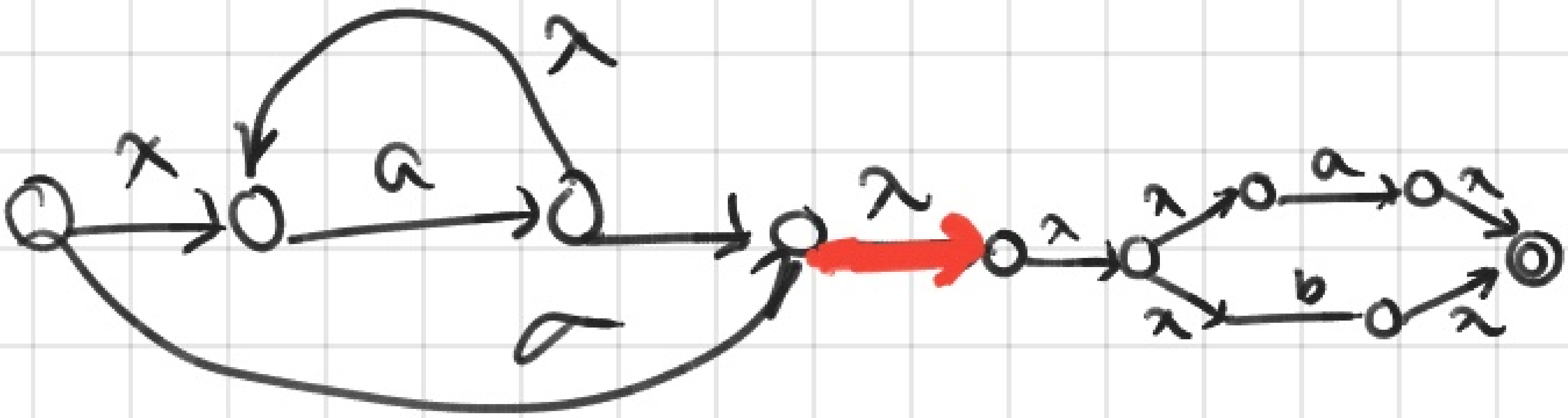


a^*

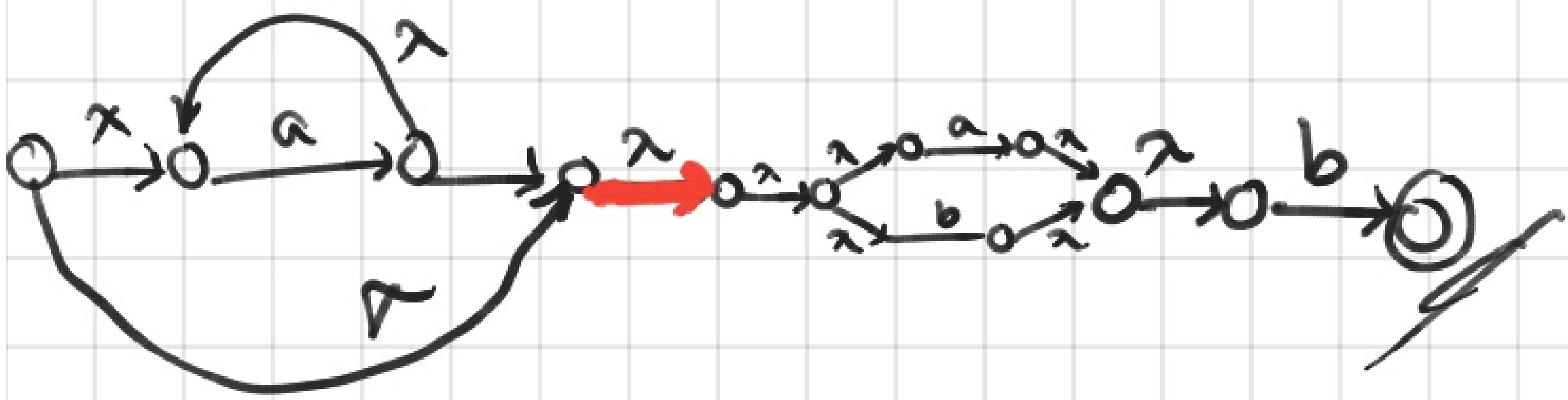
$a|b$



$a^* \cdot (a|b)$

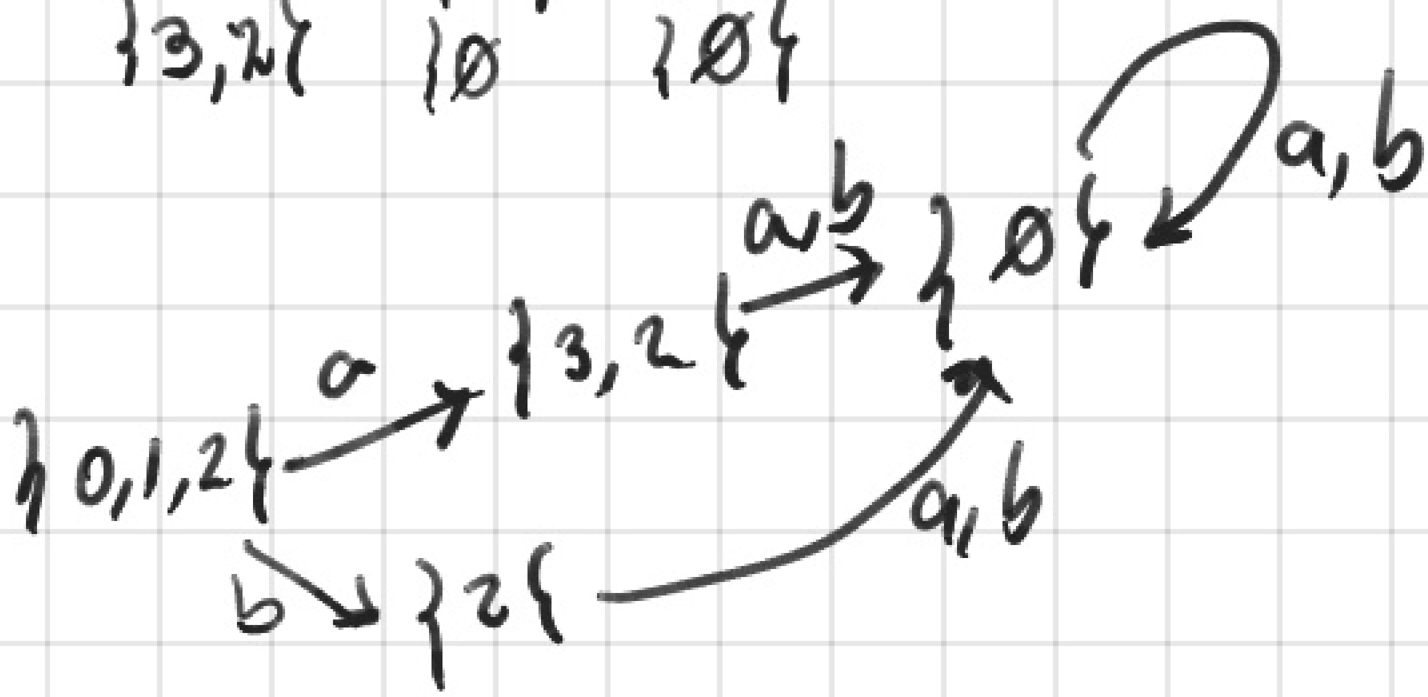


$r = a^{*} \cdot (a / b) \cdot b$



Paso de AFN a AFD

	ε	a	b
0	{0,1,2}	{3}	{2}
1	{1,2}	{2}	{∅}
2	{2}	{∅}	{∅}
3	{3,2}	{∅	{∅}

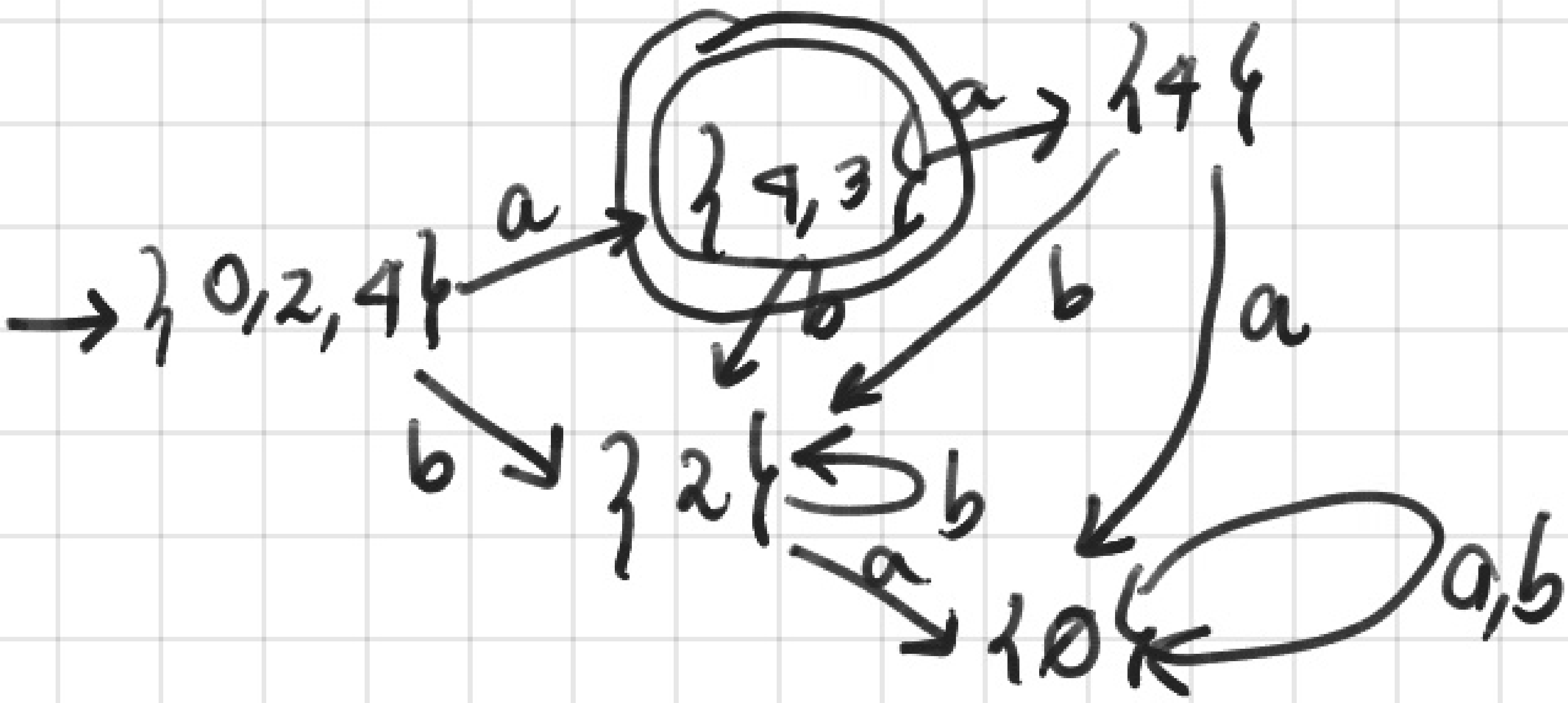


si un elemento $t \in E(s)$ tiene una transicion con un estado x

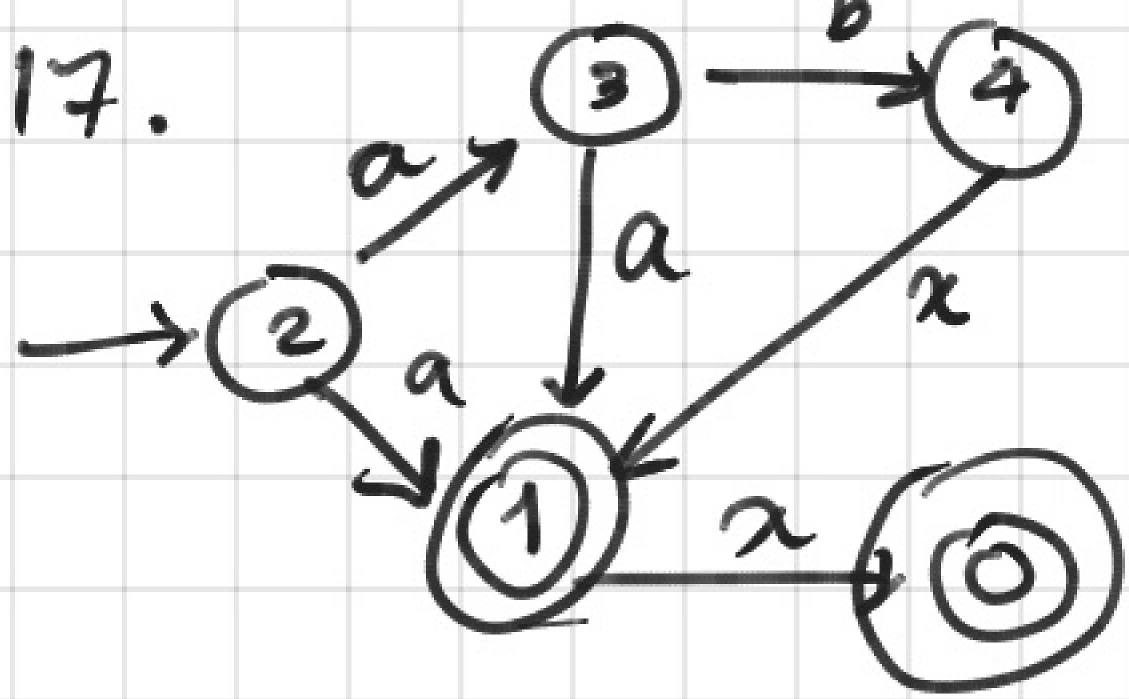
$$t \xrightarrow{a} x \cong t \xrightarrow{a} E(x)$$

(c) Dado el AFN N encuentre un AFD A / L(N) = L(A)

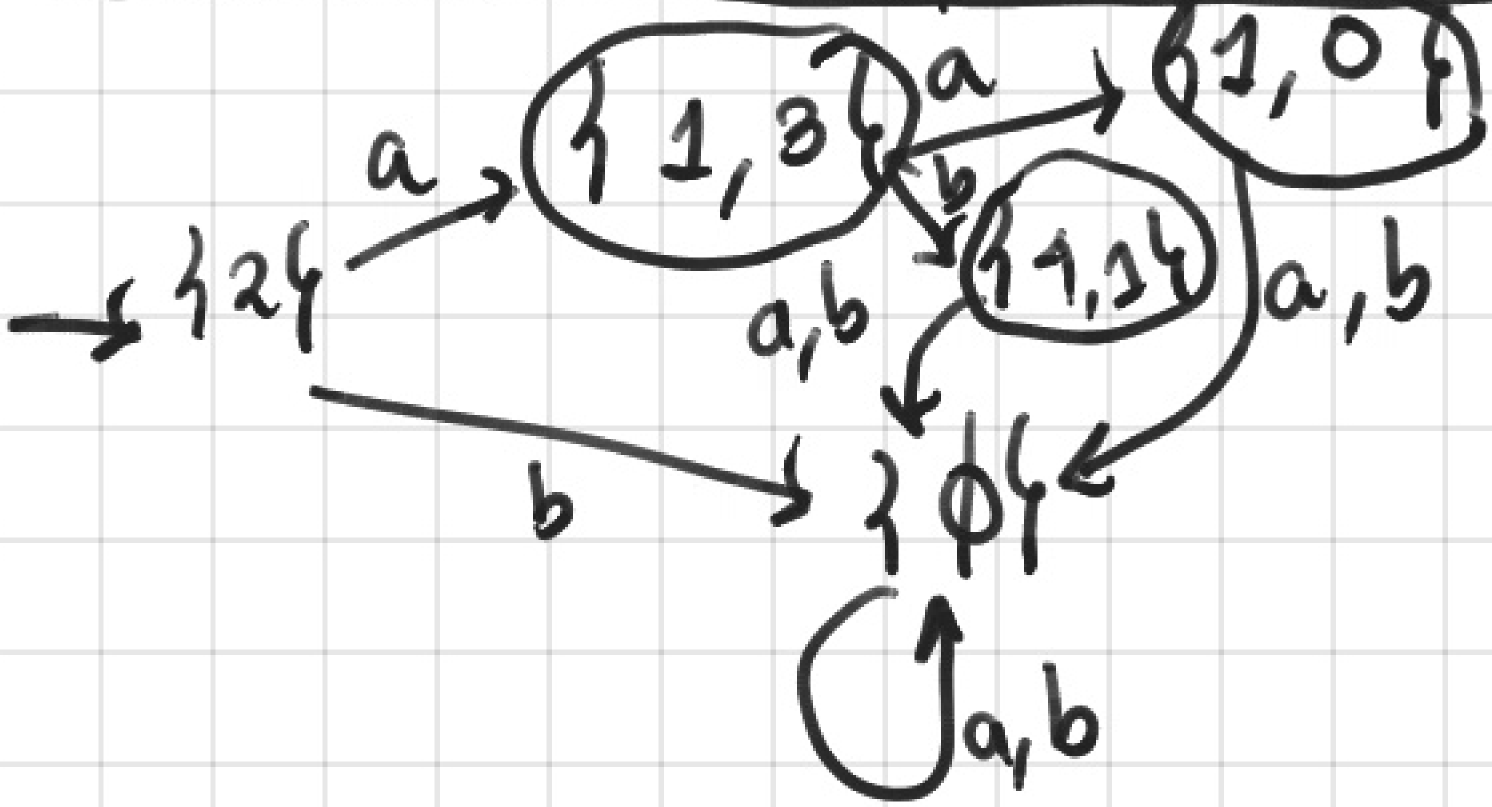
	x	a	b
0	$\{0,2,4\}$	$\{4,3\}$	$\{2\}$
2	—	$\{\emptyset\}$	$\{2\}$
3		$\{4\}$	$\{2\}$
4	—	$\{\emptyset\}$	$\{2\}$



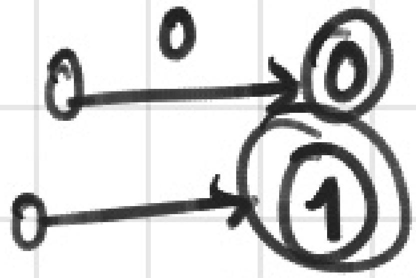
17.



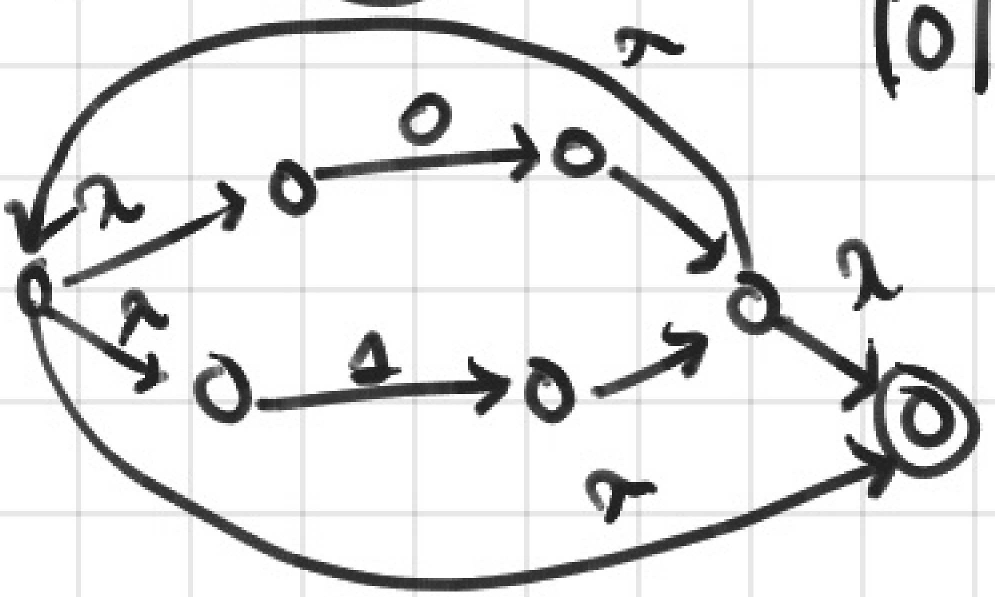
	x	a	b
2	{2}	{1,3}	{∅}
1		{∅}	{∅}
3		{1,0}	{4,1}
4		{∅}	{∅}
0		{∅}	{∅}

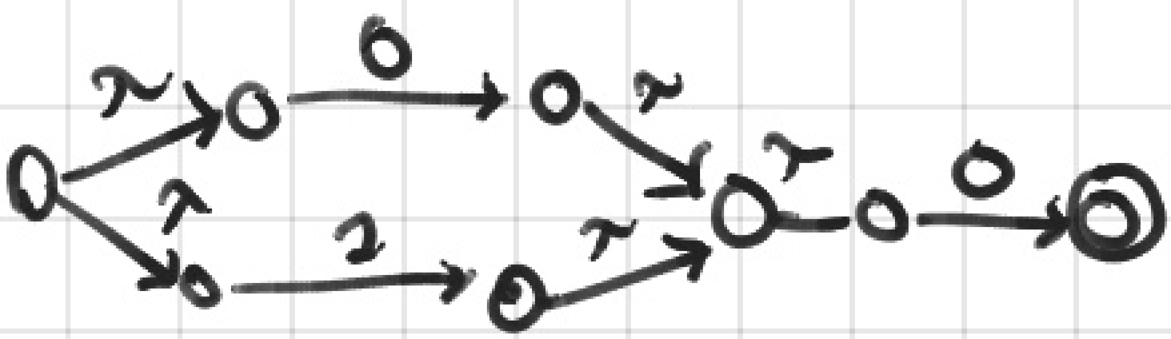


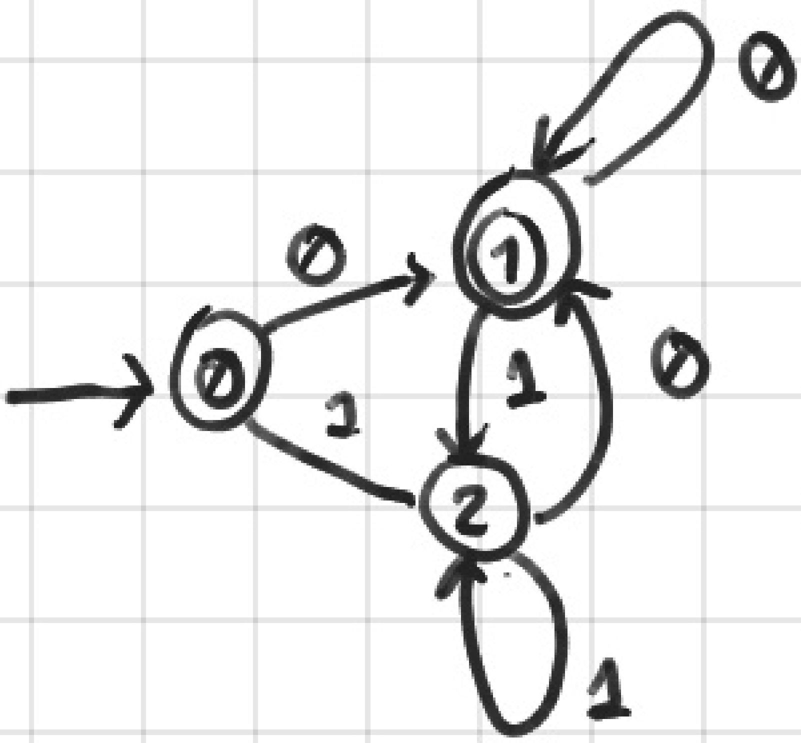
(9) $\Sigma = \{0,1\}$

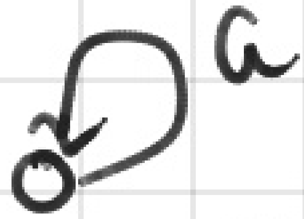


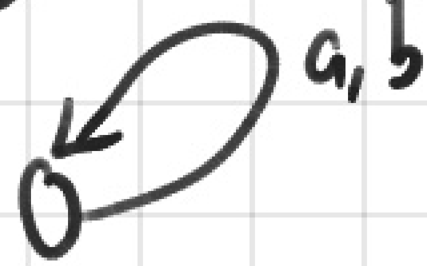
$(0|1)^* \cdot 0$

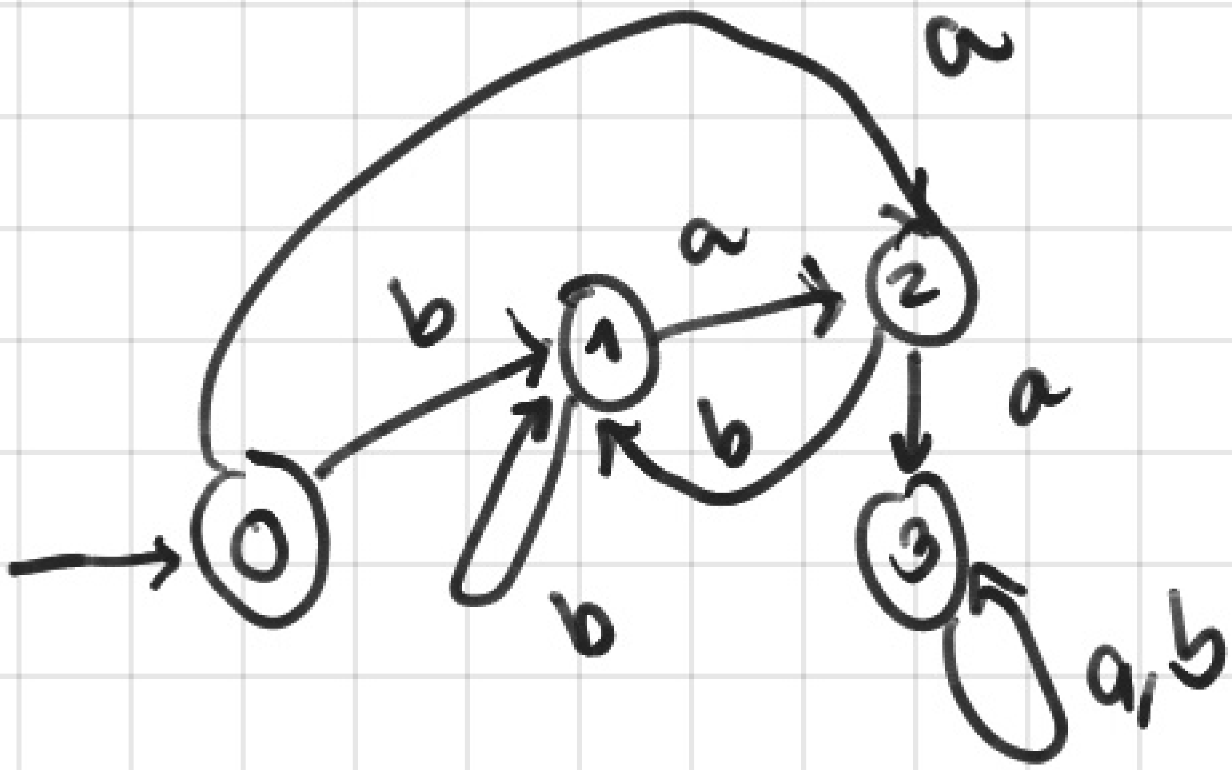




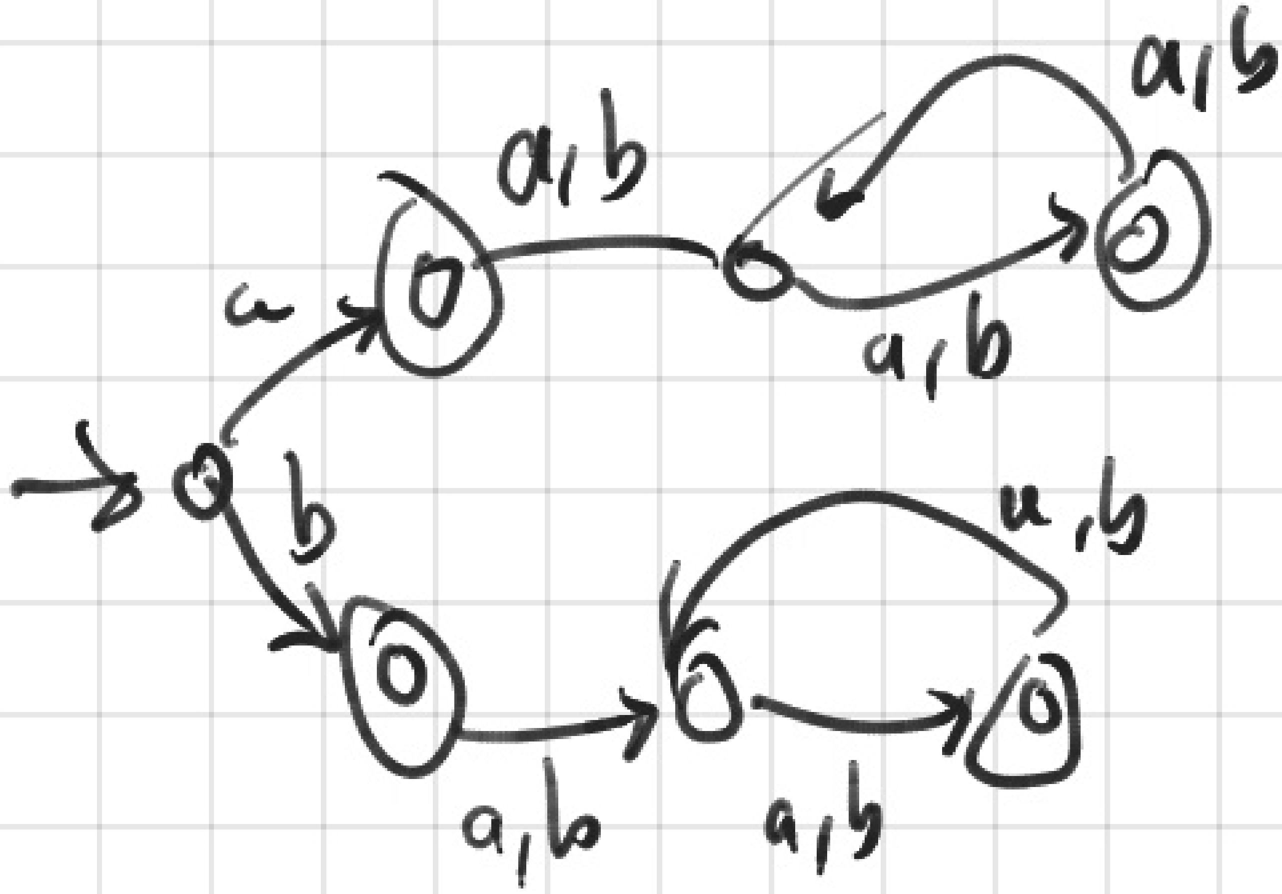


$a^* =$ 

$(a|b)^* =$ 



a
a
b
b
a
b
a,b



abbaa bbbaaba