

AUTÓMATAS Y LENGUAJES

LENGUAJES REGULARES - AÑO 2024
(Segunda Parte)
LIC. EN CS. DE LA COMPUTACIÓN - 4to. AÑO

Objetivos de la clase

- ★ *Estudiar otros dispositivos descriptores de los Lenguajes Regulares:*
 - *Expresiones Regulares (ER)*
 - *Gramáticas Regulares (GR)*
 - *Equivalencias entre ER y AFND- ϵ , GR y AFND, AFD y GR.*

Expresiones Regulares (ER)

Idea Intuitiva

- Consisten en una serie de caracteres que se pueden utilizar para acotar una búsqueda a los patrones deseados. Con ellas es posible realizar tareas como extraer una lista de e-mails de un informe o conocer cuántas páginas de un sitio web incluyen una o varias palabras determinadas en su URL.
- Por ejemplo, si introduces en Google “related:” y a continuación la dirección de una página web, el buscador te devolverá páginas web similares a aquella que indiques.

Expresiones Regulares

Idea Intuitiva

- Las **Expresiones Regulares** son otro tipo de notación para denotar lenguajes regulares, por lo tanto definen exactamente los mismos lenguajes que las distintas formas de autómatas que hemos visto (AFD, AFND, AFND- ϵ).
- No obstante ofrecen algunas facilidades que los autómatas no, una forma declarativa de expresar las cadenas que queremos aceptar.
- Estos son los aspectos en los que se enfocan las **expresiones regulares**.

Expresiones Regulares (ER)

Formalmente

Las ER utilizan las operaciones de unión (+), concatenación (.) y clausura de Kleene (*).

Una ER sobre algún alfabeto Σ se define recursivamente de la siguiente manera:

Base:

1. \emptyset es una ER, que denota el lenguaje \emptyset , es decir $L(\emptyset) = \emptyset$
2. λ es una ER, que denota el lenguaje $\{\lambda\}$, es decir $L(\lambda) = \{\lambda\}$
3. $a \in \Sigma$ es una ER, que denota el lenguaje $\{a\}$, $L(a) = \{a\}$

Expresiones Regulares (ER)

Formalmente

Inducción:

4. Si E y F son ER, luego E.F es una ER, que denota la concatenación de L(E) y L(F), $L(E.F) = L(E).L(F)$
5. Si E y F son ER, luego E + F es una ER, que denota la unión de L(E) y L(F), $L(E + F) = L(E) \cup L(F)$
6. Si E es una ER, luego E^* es una ER, que denota la clausura de L(E), $L(E^*) = (L(E))^*$
7. Si E es una ER, luego (E) es una ER, que denota el mismo lenguaje que E, $L((E)) = L(E)$

La precedencia de los operadores es de mayor a menor: clausura, concatenación y unión.

La asociatividad es de izquierda a derecha.

Expresiones Regulares

Función de Valuación

Definición: la función $\Phi : L_{ER} \rightarrow 2^{\Sigma^*}$, es denominada la función de valuación. L_{ER} es el conjunto de todas las posibles expresiones regulares. Φ toma como argumento una ER bien formada (según definición anterior) y da como resultado el conjunto de cadenas denotado por dicha ER. Luego:

- ▶ $\Phi(\emptyset) = \{\}$, el conjunto vacío.
- ▶ $\Phi(\lambda) = \{\lambda\}$, el conjunto formado por la cadena de longitud nula.
- ▶ $\Phi(a) = \{a\}$, el conjunto formado por la cadena a .
- ▶ $\Phi(E.F) = \Phi(E).\Phi(F)$, concatenación de conjuntos.
- ▶ $\Phi(E + F) = \Phi(E) \cup \Phi(F)$, la unión de conjuntos.
- ▶ $\Phi(E^*) = \Phi^*(E)$, clausura de un conjunto.
- ▶ $\Phi((E)) = \Phi(E)$.

Expresiones Regulares

Ejemplos

Sea $\Sigma = \{a,b,c\}$, determinar cuál es el lenguaje que denota cada ER:

- 1- a es una ER y denota el conjunto $\{a\}$ ($\Phi(a) = \{a\}$).
- 2- $(a + b)$ es una ER y denota el conjunto $\{a\} \cup \{b\} = \{a, b\}$
- 3- $(ab + bc)^*$, es una ER y denota el conjunto $(\{ab\} \cup \{bc\})^*$
- 4- $(a + b^*) a^* (bc)^*$ es una ER y denota el conjunto $(\{a\} \cup \{b\}^*) \cdot \{a\}^* \{bc\}^*$

Expresiones Regulares

Ejemplos

Sea $\Sigma = \{a, b\}$, construir ER para los siguientes lenguajes:

- 1- El lenguaje de todas las cadenas que empiezan con b y terminan con a.
- 2- El lenguaje de todas las cadenas que tienen exactamente dos a's.
- 3- El lenguaje de todas las cadenas que tienen número par de símbolos.
- 4- El lenguaje de todas las cadenas que tienen un número par de a's.

Ejemplos (Cont.)

Sea $\Sigma = \{a,b\}$, soluciones propuestas a los ejemplos de la página anterior:

1- $b(a + b)^* a$

2- $b^* a b^* a b^*$

3- $(aa + ab + ba + bb)^*$

4- $b^* (a b^* a)^* b^* \text{ ó } b^* (b^* a b^* a b^*) b^* \text{ ó } b^* (\lambda + a b^* a)^*$

Se deja como ejercicio determinar cuál/cuáles de la ER del ítem 4, es/son correcta/s.

Expresiones Regulares

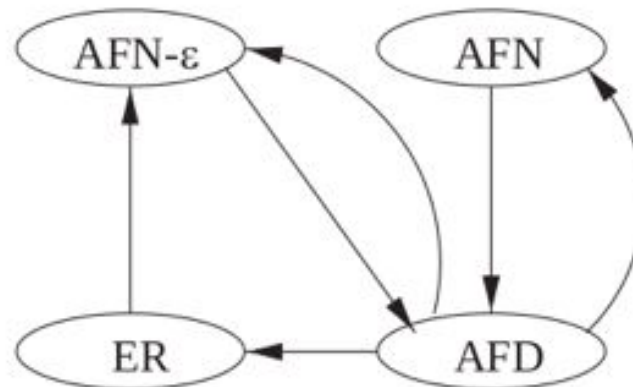
Ejemplos

Se pueden utilizar ER para describir, por ejemplo: nombres de los identificadores, los números, palabras claves, etc.

Equivalencias

Entre ER y AF

- *Para mostrar que las ER definen la misma clase de lenguajes que los AF, es decir los LR, se debe mostrar que:*
- *Todo lenguaje aceptado por un AF es denotado por una ER.*
- *Todo lenguaje denotado por una ER es aceptado por un AF.*



De ER a AF

- ❖ A partir de la demostración del siguiente teorema obtendremos un método para obtener un AF dada una ER:

Teorema: *todo lenguaje denotado por una ER es también aceptado por un AF.*

** Veremos que este AF es un AFND- ϵ*

De ER a AF

Demostración (bibliografía páginas 102 a 107)

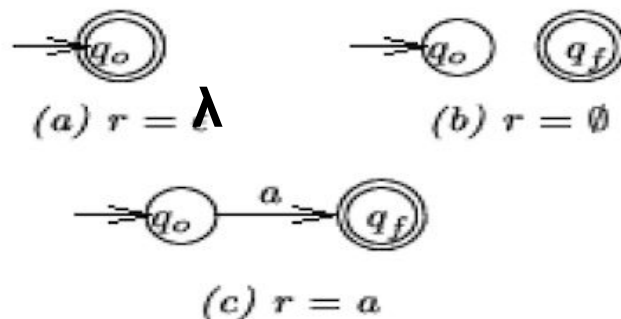
- Supongamos que $L = L(R)$ para una expresión regular R .
- Mostramos que $L = L(R) = L(E)$ para algún AFND- $\subseteq E$, siendo E tal que:
 - ★ *Tiene exactamente un estado final.*
 - ★ *No hay arcos que lleguen al estado inicial.*
 - ★ *Sin arcos que salgan del estado final.*

La prueba es por inducción estructural sobre R (es decir, sobre el número de operadores), siguiendo la definición recursiva de ER.

De ER a AF

Demostración (Cont.)

- Caso base: el número de operadores es 0 (no hay alguno), la ER r corresponde a alguna de \emptyset , λ , a como se muestra en la Figura para los casos base.



- Paso inductivo: Se asume que la afirmación es verdadera para toda ER con menos de i operadores, para $i \geq 1$. Sea r con i operadores:

De ER a AF

Demostración (Cont.)

1. $r = r_1 + r_2$, ambos r_1 y r_2 tienen menos de i operadores. Aplicando la HI: existen AFND- ϵ

$$M_1 = \langle Q_1, \Sigma_1, \delta_1, q_{o1}, \{f_1\} \rangle$$

$$M_2 = \langle Q_2, \Sigma_2, \delta_2, q_{o2}, \{f_2\} \rangle$$

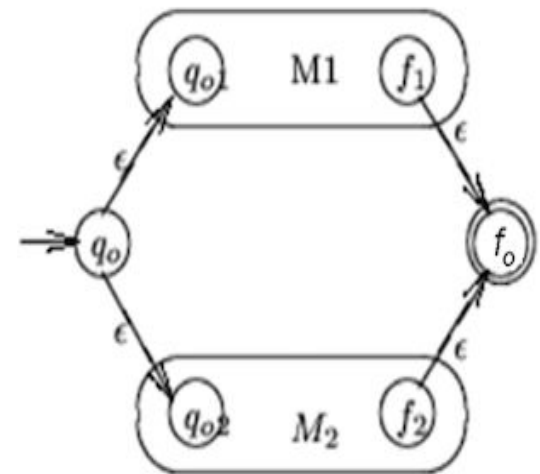
tales que $L(M_1) = L(r_1)$ y $L(M_2) = L(r_2)$. Se asume que Q_1 y Q_2 son disjuntos.

El AFND- ϵ resultante será

$$M = \langle Q_1 \cup Q_2 \cup \{q_o, f_o\}, \Sigma_1 \cup \Sigma_2, \delta, q_o, \{f_o\} \rangle$$

Por tanto :

$$L(M) = L(M_1) \cup L(M_2)$$



De ER a AF

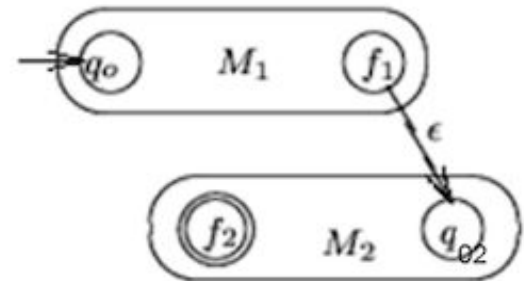
Demostración (Cont.)

2. $r = r_1 r_2$. Sean M_1, M_2 como el caso anterior.
Entonces

$$M = \langle Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta, q_{o1}, \{f_2\} \rangle$$

Por tanto:

$$L(M) = L(M_1)L(M_2)$$

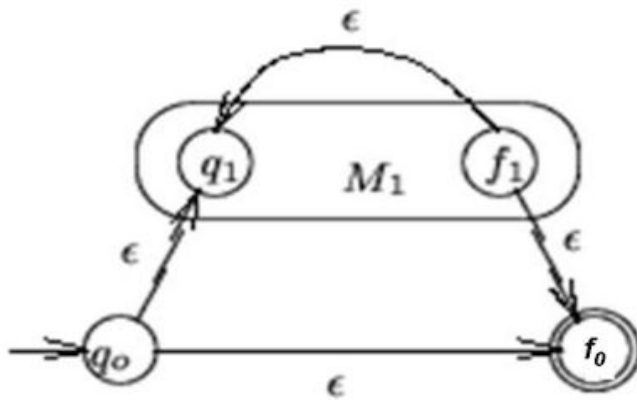


De ER a AF

Demostración (Cont.)

3. $r = r_1^*$. Sea M_1 como los casos anteriores

$$M = \langle Q_1 \cup \{q_o, f_o\}, \Sigma_1, \delta_1, q_o, \{f_o\} \rangle$$



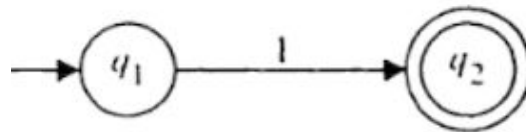
Por lo tanto: $L(M) = L(M_1)^*$

De ER a AF

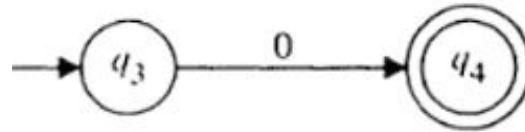
Ejemplo

Construir un AFND- ϵ para la ER $01^* + 1$, lo haremos paso a paso:

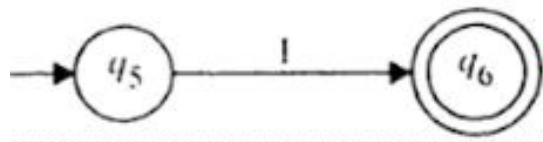
$r1 = 01^*$ y $r2 = 1$. El AFND- ϵ para $r2$ es:



Ahora, $r1 = r3r4$, donde $r3 = 0$ y $r4 = 1^*$. El AFND- ϵ para $r3$ es:

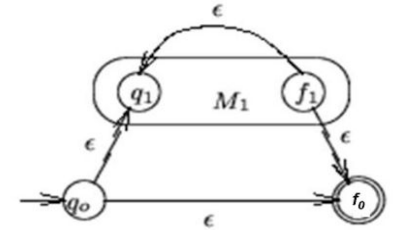


Ahora, $r4 = r5^*$, donde $r5 = 1$. El AFND- ϵ para $r5$ es:

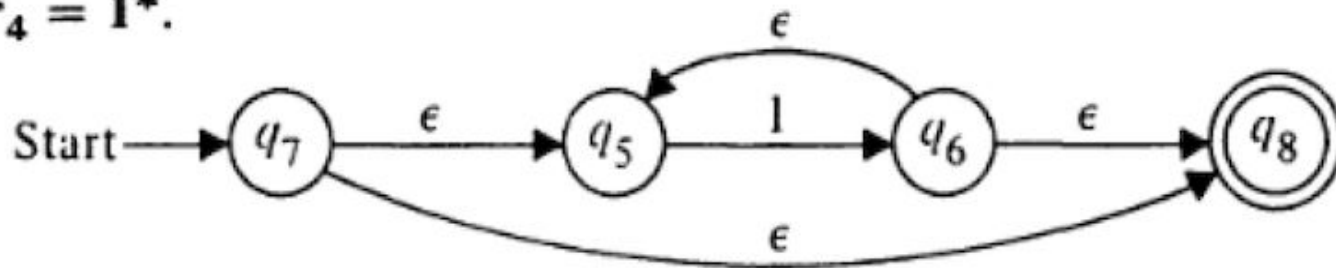


De ER a AF

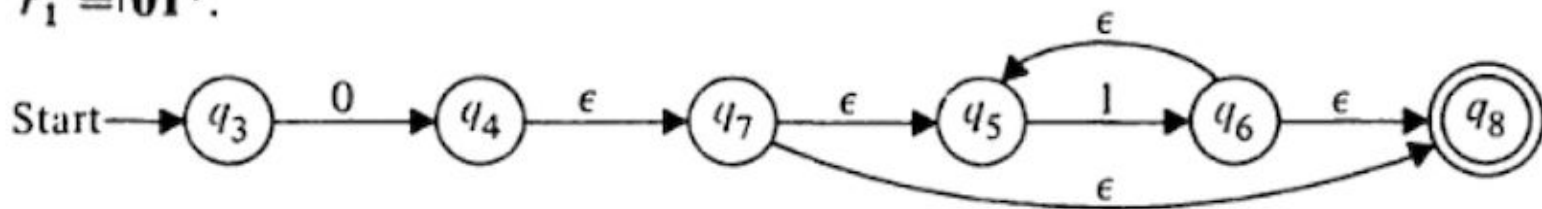
Ejemplo (Cont.)



$r_4 = 1^*$.

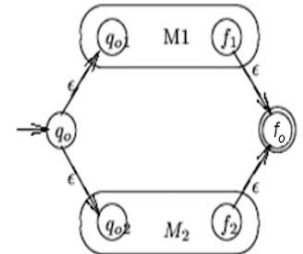


$r_1 = |01^*$.

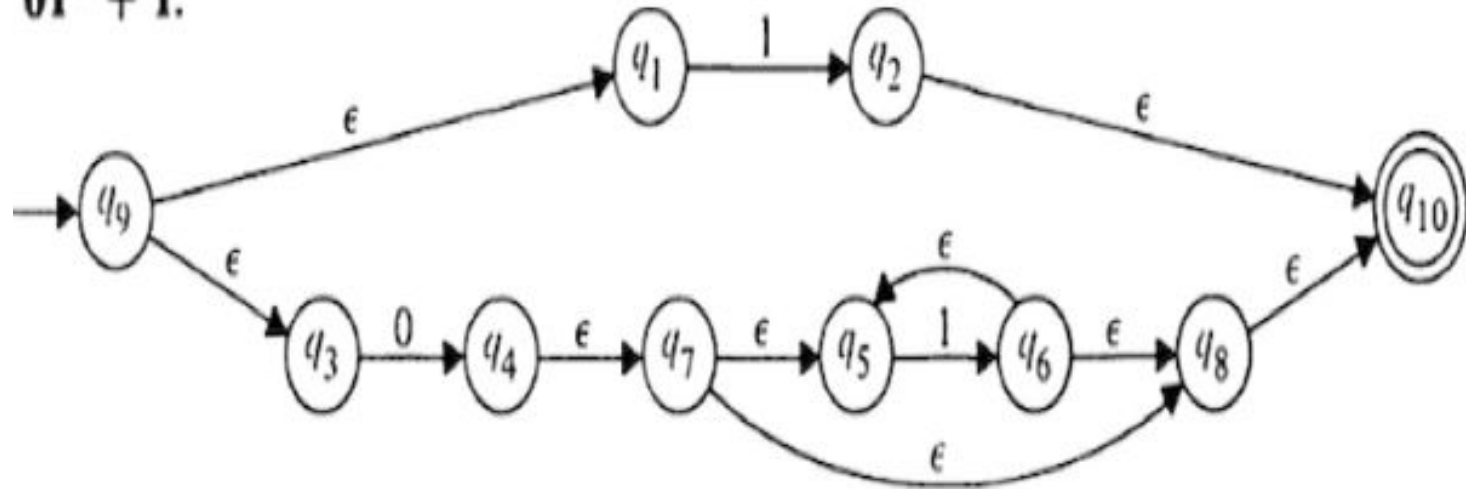


De ER a AF

Ejemplo (Cont.)



$01^* + 1.$



Equivalencias (Cont.)

Entre ER y AF

Si bien la construcción de como obtener la ER a partir del AF requiere un poco de esfuerzo, desde el punto de vista práctico, es obviamente importante desde el punto de vista teórico, para completar la prueba de que los lenguajes aceptados por AF son los mismos que los denotados por ER, es decir los LR.

Teorema: todo lenguaje aceptado por un AFD es también denotado por una ER.

Leyes Algebraicas de las ER

Sean α , β y γ expresiones regulares, entonces:

1. $\alpha + \beta = \beta + \alpha$
2. $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$
3. $(\alpha\beta)\gamma = \alpha(\beta\gamma)$
4. $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$
5. $(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma$
6. $\emptyset + \alpha = \alpha + \emptyset = \alpha$
7. $\lambda\alpha = \alpha\lambda = \alpha$
8. $\emptyset\alpha = \alpha\emptyset = \emptyset$
9. $\alpha + \alpha = \alpha$
10. $\alpha\alpha^* = \alpha^*\alpha$
11. $\alpha\alpha^* + \lambda = \alpha^*$
12. $(\alpha^*\beta^*)^* = (\alpha + \beta)^*$
13. $(\alpha^*)^* = \alpha^*$

Gramáticas Regulares

- Definición:

En una *gramática regular* $G = (N, \Sigma, P, S)$, las producciones en P pueden ser solamente de la siguiente forma:

1. $X \rightarrow aY$, con $X, Y \in N$ y $a \in \Sigma$
2. $X \rightarrow a$, con $X \in N$ y $a \in \Sigma$

Con la siguiente excepción. Si λ debe ser generada por la gramática, se permite la producción, $S \rightarrow \lambda$ mientras el símbolo distinguido no aparezca en la parte derecha de alguna producción.

La familia de lenguajes aceptados por AFs es equivalente a la familia de lenguajes generados por gramáticas regulares y a los denotados por ERs.

Gramáticas Regulares

Relación Deriva

- ▶ Sean $\alpha, \beta, \delta, \gamma \in (\Sigma \cup N)^*$, definimos la relación deriva (\Rightarrow) sobre $(\Sigma \cup N)^*$ como sigue:

$$\delta\alpha\gamma \Rightarrow \delta\beta\gamma \text{ si } \exists \alpha \rightarrow \beta \in P$$

Esto significa que $\delta\beta\gamma$ es obtenida a partir de $\delta\alpha\gamma$ por la aplicación de la regla o producción $\alpha \rightarrow \beta \in P$

- ▶ Si $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \dots \Rightarrow \alpha_n$ con $n \geq 0$ y $\alpha_i \in (N \cup \Sigma)^*$ para $i \in \{0, 1, \dots, n\}$, luego decimos que $\alpha_0 \xRightarrow{*} \alpha_n$, es decir la clausura reflexo- transitiva o que α_0 deriva en 0 o más pasos en α_n .

Gramáticas Regulares

Relación Deriva (Cont.)

- ▶ Si $\alpha_n \in (\Sigma \cup N)^*$, luego α_n es denominado una *forma sentencial*.
- ▶ Si $\alpha_n \in \Sigma^*$, es una *sentencia* del lenguaje.

Lenguaje generado:

Definición: El lenguaje generado por una gramática G denotado $L(G)$ es el conjunto, $L(G) = \{x \in \Sigma^* \mid S \xRightarrow{*} x\}$.

Gramáticas Regulares

Ejercicio

Determinar usando la relación deriva (\Rightarrow) qué cadenas genera la gramática, para luego obtener cuál es el lenguaje que generado por la misma:

1. $S \rightarrow aA$
2. $S \rightarrow bA$
3. $A \rightarrow aB$
4. $A \rightarrow bB$
5. $A \rightarrow a$
6. $B \rightarrow aA$
7. $B \rightarrow bA$

*El lenguaje generado por la gramática es el de las cadenas sobre $\Sigma = \{a, b\}$, **de longitud par terminadas en a.***

Gramáticas Regulares

Ejercicio

Construir una gramática regular que genere el lenguaje de las palabras, sobre $\Sigma = \{a, b\}$, que contienen la subcadena bb .

1. $A \rightarrow aA$
2. $A \rightarrow bB$
3. $B \rightarrow aA$
4. $B \rightarrow bC$
5. $B \rightarrow b$
6. $C \rightarrow aC$
7. $C \rightarrow bC$
8. $C \rightarrow a$
9. $C \rightarrow b$

Equivalencia de GR a AFND

Teorema: La clase de los lenguajes generados por gramáticas regulares es exactamente la de los lenguajes regulares.

La prueba de este teorema consiste en dar un procedimiento para, a partir de una gramática dada, construir un autómata finito y viceversa.

Dicho procedimiento es directo, y consiste en asociar a los símbolos no terminales de la gramática, los estados del autómata:

Es decir, para cada regla $A \rightarrow bC$ en la gramática, tendremos $\delta(A, b) = C$, en el autómata.

Pero qué sucede con las reglas del tipo $A \rightarrow b$, para estos casos tendremos las transiciones $\delta(A, b) = Z$, donde Z es un nuevo estado, que no está asociado a ningún no terminal y además es el único estado final del autómata.

El estado inicial es el rotulado con el símbolo distinguido.

Equivalencia de AFD a GR

De manera similar, a la construcción anterior, se puede obtener a partir de un AFD $M = (Q, \Sigma, q_0, \delta, F)$, la gramática regular que genere el mismo lenguaje.

La idea es, para cada transición $\delta(p, \sigma) = q$, habrá en la gramática una regla $X_p \rightarrow \sigma X_q$, con X_i el no terminal de la gramática que corresponde al estado i del AFD.

Pero falta determinar como obtener las reglas $X_p \rightarrow \sigma$, que son las que permiten terminar una derivación.

La aplicación de este tipo de reglas debe corresponder al consumo del último carácter de una palabra aceptada en el AFD, pero para ello necesariamente nos debemos encontrar en un estado final.

¿DUDAS?

