Contents

Notas de Lenguajes Formales y Autómatas

César Bautista Ramos

Fac. Ciencias de la Computación, BUAP

Autómata finito

Estudiaremos lenguajes *formales*, esto es "matemáticos"; no confundirlos con los lenguajes *naturales*, que son los que habla la gente.

1. Alfabetos y lenguajes

Los lenguajes se forman de *palabras* y las palabras se forman de símbolos de un *alfabeto*.

Definición 1. *Un* **alfabeto** Σ *es un conjunto no vacío y finito de símbolos.*

Ejemplo 1. El conjunto $\Sigma = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$ es un alfabeto llamado *alfabeto inglés*.

Ejemplo 2. El conjunto $\Sigma = \{\alpha, \beta, \delta\}$ es un alfabeto.

Ejemplo 3. El conjunto $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ es un alfabeto.

Definición 2. *Una* **palabra ó cadena** *sobre el alfabeto* Σ *es una sucesión finita de símbolos de* Σ*.*

Ejemplo 4. La secuencia *program* es una palabra sobre el alfabeto inglés. También *digit*, *hda* y *quetzal* son palabras sobre el alfabeto inglés, así como *bxweh*.

En las sucesiones el orden es importante. Esto es, las palabras aba y aab se forman de los mismos símbolos: dos a y una b, pero su orden de aparición es diferente, por lo que

$$aba \neq aab.$$

Este es un hecho general.

Así como en la teoría de conjuntos hay que aceptar a la colección vacía como un conjunto (el conjunto vacío $\varnothing$), en la teoría de lenguajes formales hay que aceptar a la *palabra vacía* como una palabra genuina.

Definición 3. *Si Σ es un alfabeto,* **la cadena vacía** ϵ *es una palabra sobre Σ.*

Enseguida definimos nuestro principal objeto de estudio.

Definición 4. *Un* **lenguaje** *A sobre un alfabeto Σ es un conjunto de palabras sobre Σ.*

Ejemplo 5. Sea $A = \{1, 12, 123, 1234, 123456, 0\}$. El conjunto A es un lenguaje sobre el alfabeto $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. El conjunto $B = \{1, 11, 111, 1111, \ldots\}$ es también un lenguaje (infinito) sobre Σ. Resulta que también $\varnothing$ es un lenguaje sobre Σ.

No debemos confundir a la palabra vacía ϵ con el lenguaje vacío:

$$\epsilon \neq \varnothing.$$

Esta no igualdad será evidente cuando estudiemos las operaciones sobre lenguajes. Resulta que la palabra vacía ϵ tiene propiedades muy diferentes a las del lenguaje vacío $\varnothing$.

Ejemplo 6. El conjunto $A = \{a, ab, aab, aaab, \ldots\}$ es un lenguaje (infinito) sobre el alfabeto inglés $\Sigma = \{a, b, \ldots, z\}$. También $B = \{\epsilon\}$ es un lenguajes sobre Σ así como también $\varnothing$ es un lenguaje sobre el alfabeto inglés.

Dado un alfabeto Σ uno puede considerar todas las posibles palabras formadas por tal alfabeto. Se obtiene entonces un *lenguaje universal*.

Definición 5. *Si Σ es un alfabeto, la* **cerradura de** Σ *ó* **lenguaje universal** *sobre Σ se denota con*

$$\Sigma^*$$

y este es el conjunto de todas las palabras sobre Σ:

$$\Sigma^* = \{w \mid w \text{ es una palabra sobre } \Sigma\}.$$

Nótese que

$$\forall \Sigma \text{ alfabeto }, \quad \epsilon \in \Sigma^*.$$

Ejemplo 7. Si $\Sigma = \{1\}$, entonces

$$\Sigma^* = \{\epsilon, 1, 11, 111, 1111, \ldots\}$$

2. Operaciones

Definición 6. *Si w es una cadena sobre el alfabeto Σ, su* **longitud** *se denota con $|w|$.*

Ejemplo 8. *Sea $\Sigma = \{0, 1, \ldots, 9\}$ y $w = 121$. Entonces $|w| = 3$. Además $|\epsilon| = 0$.*

Definición 7. *La* **concatenación de palabras** *es una operación "$\cdot$":*

$$\begin{aligned} \cdot : \Sigma^* \times \Sigma^* &\to \Sigma^* \\ (w, z) &\mapsto w \cdot z = wz \end{aligned}$$

Ejemplo 9. *Si $w = aba$ y $z = bad$ entonces $w \cdot z = ababad$.*

Es común omitir el símbolo de la operación de concatenación. Por ejemplo, en el anterior,

$$wz = ababad$$

Notemos las siguientes propiedades generales

(1) Si w, z son palabras entonces $|wz| = |w| + |z|$.

(2) Si $w \in \Sigma^*$, entonces

   (a) $\epsilon w = w$

   (b) $w\epsilon = w$

(3) En general $wz \neq zw$.

Definición 8 (potencia)**.** *Si $n \in \mathbb{N}$ y $w \in \Sigma^*$, se define*

$$w^n = \begin{cases} \epsilon & si\ n = 0 \\ ww^{n-1} & si\ n > 0 \end{cases}$$

llamada la **n-ésima potencia** *de w.*

Ejemplo 10. Si $w = 122$ entonces

$$\begin{aligned} w^0 &= \epsilon \\ w^1 &= ww^0 = 122\epsilon = 122 \\ w^2 &= ww^1 = 122122 \\ w^3 &= ww^2 = 122122122 \end{aligned}$$

La igualdad entre palabras se podría definir como: si $w, z \in \Sigma^*$, se pone $w = z$ en caso de que $|w| = |z|$ y de que tengan los mismos símbolos en la misma posición.

Definición 9. *Sean $w, x \in \Sigma^*$.*

(1) *Se dice que x es* **prefijo** *de w si $\exists y \in \Sigma^*$ tal que*

$$w = xy$$

(2) *Se dice que x es* **prefijo propio** *de w si x es prefijo de w, pero $w \neq x$.*

Ejemplo 11. *Sea $w = 121$. Entonces*

(1) *$x = 1$ es prefijo (propio) de w;*

(2) *$u = 12$ es prefijo (propio) de w;*

(3) *$w = 121$ es prefijo de w, pues $w = w\epsilon$, pero no es propio.*

Definición 10. *Una cadena $w \in \Sigma^*$ es subpalabra de $z \in \Sigma^*$ si $\exists x, y \in \Sigma^*$ tales que*

$$z = xwy$$

Ejemplo 12.

(1) *Si $w \in \Sigma^*$ entonces w es subpalabra de la misma w, pues*

$$w = \epsilon w \epsilon$$

(2) *$w = 2$ es subpalabra de $z = 121$; a su vez $y = 12$ es subpalabra de z, pues $z = \epsilon y 1$.*

El siguiente concepto será útil para definir nuevos lenguajes.

Definición 11. *Si $w \in \Sigma^*$, la* **inversa o transpuesta** *de w es la imagen reflejada de w que se denota w^I. Esto es:*

$$w^I = \begin{cases} w & \text{si } w = \epsilon \\ y^I a & \text{si } w = ax \text{ con } a \in \Sigma \text{ y } y \in \Sigma^* \end{cases}$$

Ejemplo 13. Si $w = ecos$, entonces

$$\begin{aligned} w^I &= (cos)^I e \\ &= (os)^I ce \\ &= s^I oce \\ &= (s\epsilon)^I oce \\ &= \epsilon^I soce \\ &= \epsilon soce \\ &= soce \end{aligned}$$

Propiedad 1. *Si $w, y \in \Sigma^*$, entonces*

$$(wy)^I = y^I w^I$$

Proof. Por inducción sobre $n = |w|$. Si $n = 0$, entonces $w = \epsilon$, luego

$$\begin{aligned}(wy)^I &= (\epsilon y)^I \\ &= y^I\end{aligned}$$

mientras que

$$\begin{aligned}w^I y^I &= \epsilon y^I, && \text{por definición,} \\ &= y^I\end{aligned}$$

por lo tanto $(wy)^I = w^I y^I$.

Supongamos cierto el resultado para palabras w_0 de longitud n: esto es

$$(w_0 y)^I = y^I w_0^I \tag{1}$$

Ahora tomemos una palabra w de longitud $|w| = n + 1$. Entonces $w = az$ con $a \in \Sigma$ y $z \in \Sigma^*$ con $|z| = n$. Luego

$$\begin{aligned}(wy)^I &= (azy)^I \\ &= (zy)^I a, && \text{por definición de inversa,} \\ &= y^I z^I a, && \text{por hipótesis de inducción (1),} \\ &= y^I (az)^I, && \text{por definición de inversa} \\ &= y^I w^I \quad .\end{aligned}$$

□

3. Operaciones con lenguajes

Así como las palabras se pueden concatenar, también se puede hacer una operación similar sobre lenguajes.

Definición 12. *Si A es lenguaje sobre el alfabeto Σ_1 y B es lenguaje sobre el alfabeto Σ_2, se define el* **lenguaje concatenación** *de A con B como*

$$A \cdot B = \{w \cdot w \mid w \in A \ y \ x \in B\}$$

sobre el alfabeto $\Sigma_1 \cup \Sigma_2$.

Esto es, el lenguaje $A \cdot B$ está formado por todas las posibles concatenaciones de las cadenas de A con las de B. También, como en la concatenación de cadenas, el símbolo de concatenación "·" se acostumbra omitir y se pone $AB = A \cdot B$.

Ejemplo 14. *Si $A = \{casa\}$, $B = \{pajaro, perro\}$, entonces*

$$AB = \{casapajaro, casaperro\}$$

Entre los lenguajes, el lenguje $\{\epsilon\}$ se comporta como 1 con respecto a la operación de concatenación.

Propiedad 2. *Si A es un lenguaje arbitrario, entonces*

$$A\{\epsilon\} = A = \{\epsilon\}A.$$

Proof.

$$\begin{aligned} A\{\epsilon\} &= \{w\epsilon \mid w \in A\}, && \text{por definición de concatenación,} \\ &= \{w \mid w \in A\} \\ &= A; \end{aligned}$$

similarmente se prueba $\{\epsilon\}A = A$. □

También se puede hablar de potencia de un lenguaje:

Definición 13. *(potencia) Sea $n \in \mathbb{N}$:*

$$A^n = \begin{cases} \{\epsilon\}, & si\ n = 0 \\ A \cdot A^{n-1}, & si\ n \geq 1 \end{cases}$$

Ejemplo 15.

(1) $\varnothing^0 = \{\epsilon\}$

(2) Sea $A = \{ab\}$ lenguaje formado por sólo una palabra. Entonces

$$\begin{aligned} &A^0 = \{\epsilon\} \\ &A^1 = A \cdot A^0 = A\{\epsilon\} = A = ab \\ &A^2 = A \cdot A^1 = A\{ab\} = abab \\ &A^3 = A \cdot A^2 = A\{abab\} = ababab \end{aligned}$$

etcétera.

Definición 14. *Sean A, B lenguajes. Entonces*

(1) *El* **lenguaje unión** *es:*

$$A \cup B = \{x \mid x \in A \vee x \in B\}$$

(2) *El* **lenguaje intersección** *es*

$$A \cap B = \{x \mid x \in A \wedge x \in B\}$$

(3) *El* **lenguaje diferencia** *es*

$$A - B = \{x \mid x \in A \wedge x \notin B\}$$

Ejemplo 16. Sea $\Sigma = \{0, 1\}$ y lenguajes $A = \{\epsilon, 0, 10, 11\}$, $B = \{\epsilon, 1, 0110, 11010\}$. Luego,

$$\begin{aligned} A \cup B &= \{\epsilon, 0, 1, 10, 11, 0110, 11010\} \\ A \cap B &= \{\epsilon, 1\} \\ A - B &= \{0, 10, 11\} \\ B - A &= \{0110, 11010\} \end{aligned}$$

En general, como los lenguajes son conjuntos, los lenguajes heredan todas las propiedades y terminología de los conjuntos.

Definición 15. *Sean A, B lenguajes sobre un alfabeto Σ. Si $A \subseteq B$, entonces se dice que A es* **sublenguaje** *de B.*

Ejemplo 17. Sea $A = \{a, aa, aaa, aaaa, aaaaa\}$ y $B = \{a^n \mid n \in \mathbb{N}\}$, entonces A es sublenguaje de B.

En general, si L es un lenguaje sobre un alfabeto Σ entonces L es un sublenguaje de Σ^*.

También, recordemos que de la definición de la igualdad de conjuntos podemos obtener que dos lenguajes A, B son iguales: $A = B$ si y sólo si

(1) $A \subseteq B$
(2) $B \subseteq A$

Esta observación nos ayudará a demostrar el siguiente teorema.

Teorema 1. *Sean A, B, C lenguajes sobre un alfabeto Σ. Se cumple que*

(1) $A \cdot (B \cup C) = (A \cdot B) \cup (A \cdot C)$
(2) $(B \cup C) \cdot A = (B \cdot A) \cup (C \cdot A)$

Proof.

(1) Por contenciones, esto probaremos que
 (a) $A \cdot (B \cup C) \subset (A \cdot B) \cup (A \cdot C)$
 (b) $(A \cdot B) \cup (A \cdot C) \subset A \cdot (B \cup C)$
 (a): Si $x \in A \cdot (B \cup C)$ entonces $x = w \cdot y$ con $w \in A$ y $y \in B \cup C$; luego $y \in B$ ó $y \in C$. Si $y \in B$ entonces $x = wy \in A \cdot B$; y si $y \in C$, entonces $x = w \cdot y \in A \cdot C$, así,

$$x \in (A \cdot B) \cup (A \cdot C)$$

 (b): Si $x \in (A \cdot B) \cup (A \cdot C)$ entonces $x \in A \cdot B$ ó $x \in A \cdot C$. Si $x \in A \cdot B$ entonces $x = wy$ con $w \in A$ y $y \in B \subseteq B \cup C$, luego $x = wy$ con $w \in A$ y $y \in B \cup C$.
 Si $x \in A \cdot C$ entonces $x = wy$ con $w \in A$ y $y \in C \subseteq B \cup C$, entonces $x = wy$ con $w \in A$ y $y \in B \cup C$, luego $x \in A \cdot (B \cup C)$.

En cualquier caso

$$x \in A \cdot (B \cup C).$$

(2) Tarea.

□

Notemos que en general, no es cierto que

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

por la culpa del contrajemeplo siguiente: $A = \{a, \epsilon\}$, $B = \{\epsilon\}$, $C = \{a\}$, entonces $B \cap C = \varnothing$ y así

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

mientras que $A \cdot B = A = a, \epsilon$ y $A \cdot C = \{aa, a\}$, por lo que

$$(A \cdot B) \cap (A \cdot C) = \{a\} \neq A \cdot (B \cap C)$$

Uno de los concepto fundamentales de la teoría es el de *cerradura*.

Definición 16. *Sea A lenguaje sobre el alfabeto Σ.*

(1) *La* **cerradura de Kleene** *o* **cerradura estrella** *de A es*

$$A^* = \cup_{n=0}^{\infty} A^n$$

(2) *La* **cerradura positiva** *de A es*

$$A^+ = \cup_{n=1}^{\infty} A^n$$

La cerradura de Kleene se obtiene al hacer cero o más concateneaciones de las palabras de A, mientras que la cerradura positiva se obtiene al hacer una o más concatenaciones.

Ejemplo 18. $A = \{a\}$. Entonces $A^0 = \{\epsilon\}$, $A^1 = a$, $A^2 = aa$, $A^3 = aaa, \ldots$ entonces

$$A^* = \{\epsilon, a, a^2, a^3, \ldots\}$$

mientras que

$$A^+ = \{a, a^2, a^3, \ldots\}$$

Ejemplo 19. Sea Σ un alfabeto. En particular el propio Σ es un alfabeto formado por las palabras de longitud 1. Luego la cerradura de Kleene de Σ es

$$\cup_{n=0}^{\infty} \Sigma^n$$

que es ϵ junto con todas las concatenaciones de palabras sobre Σ que es precisamente el lenguaje universal Σ^*. Este razonamiento muestra que nuestra notación para el lenguaje universal es consistente con la notación de la cerradura de Kleene:

$$\underbrace{\Sigma^*}_{\text{lenguaje universal}} = \underbrace{\Sigma^*}_{\text{cerradura de Kleene}}$$

Propiedad 3. *Si A es un lenguaje sobre Σ, entonces*

(1) $A^* \subseteq \Sigma^*$

(2) $A^+ \subseteq A^*$

Proof.

(1) $\forall n \geq 0,\ A^n \subseteq \Sigma^*$, entonces

$$A^* = \cup_{n=0}^{\infty} \subset \Sigma^*$$

(2) $forallk \geq 1,\ A^k \subseteq \cup_{n=0}^{]} \infty A^n = A^*$, entonces

$$A^+ = \cup_{k=1}^{\infty} \subseteq A^*$$

□

Ejemplo 20. Tenemos que $\varnothing$ es un lenguaje. Entonces

$$\varnothing^0 = \{\epsilon\},\ \varnothing^1 = \varnothing,\ \varnothing^2 = \varnothing, \ldots$$

por lo que

$$\varnothing^* = \{\epsilon\}, \quad \varnothing^+ = \varnothing$$

Uno podría pensar que la diferencia entre la cerradura de Kleene y la cerradura positiva es la palabra vacía ϵ. *Esto no siempre es cierto*, como puede notarse en el siguiente ejemplo.

Ejemplo 21. Sea $\Sigma = \{0, 1, \ldots, 9\}$, y consideremos el lenguaje

$$A = \{w \in \Sigma^* \,|\, w \text{ no contiene ninguno de los dígitos } 0, 1, \ldots, 9\}$$

Luego, $\epsilon \in A$, $0 \in A$, $1 \in A$, $01010100111 \in A$. Nos proponemos demostrar que $A^* = A^+$.

Si $k \geq 1$ y $x \in A^k$, entonces $x = w_1 \cdots w_k$ con cada $w_i \in A$ cadena conteniendo sólo 0's y 1's. Luego x contiene sólo 0's y 1's. Por lo tanto,

$$\forall k \geq 1, A^k \subseteq A \ .$$

Además, si $k \geq 1$ y $x \in A$, entonces $x = \epsilon^{k-1} x \in A^k$, esto es $A \subseteq A^k$:

$$\forall k \geq 1, A^k = A$$

Por lo que

$$A^+ = \cup_{n=1}^{\infty} A^k = A \ .$$

Pero también, como $A^0 = \{\epsilon\} \subseteq A$, se sigue

$$\begin{aligned} A^* &= A^0 \cup A^+ \\ &= A^0 \cup A \\ &= A \end{aligned}$$

por lo tanto

$$A^* = A = A^+$$

Como puede notarse del ejemplo anterior en algunos casos $A^+ = A^*$.

Lema 1. *Sean $A, A_0, A_1, \ldots$ una colección infinita de lenguajes sobre Σ. Entonces*

(1) $A \cdot \cup_{n=0}^{\infty} A_n = \cup_{n=0}^{\infty} A \cdot A_n$

(2) $(\cup_{n=0}^{\infty} A_n) \cdot A = \cup_{n=1}^{\infty} A_n \cdot A$

Proof.

(1) Por demostrar

  (a) $A \cdot \cup_{n=0}^{\infty} A_n \subseteq \cup_{n=0}^{\infty} A \cdot A_n$

  (b) $\cup_{n=0}^{\infty} A \cdot A_n \subseteq A \cdot \cup_{n=1}^{\infty} A_n$

  (a): Si $x \in A \cdot \cup_{n=0}^{\infty} A_n$, entoces $x = w \cdot y$ con $w \in A$ y $y \in \cup_{n=0}^{\infty} A_n$, luego existe k_0 tal que $y \in A_{k_0}$, así $x = wy \in A \cdot A_{k_0}$, lo que implica que

$$x \in \cup_{n=0}^{\infty} A \cdot A_n \quad .$$

  (b): Si $x \in \cup_{n=0}^{\infty} A \cdot A_n$ entonces existe k_0 tal que $x \in A \cdot A_{k_0}$, por lo que $x = wy$ con $w \in A$ y $y \in A_{k_0}$, es decir $y \in \cup_{n=0}^{\infty} A_n$. Así

$$x \in A \cdot \cup_{n=0}^{\infty} A_n$$

  Por lo tanto

$$A \cdot \cup_{n=0}^{\infty} A_n = \cup_{n=0}^{\infty} A \cdot A_n \quad .$$

(2) Tarea.

□

Teorema 2.

$$A^+ = A \cdot A^* = A^* \cdot A$$

Proof.

$$\begin{aligned} A \cdot A^* &= A \cdot \cup_{n=0}^{\infty} A^n \\ &= \cup_{n=0}^{\infty} A \cdot A^n \\ &= \cup_{n=0}^{\infty} A^{n+1} \\ &= \cup_{k=1}^{\infty} A^k \\ &= A^+ \quad . \end{aligned}$$

y similarmente $A^* \cdot A = A^+$. □

Ejemplo 22. Sea $\{A\} = \{ab\}$ lenguaje sobre el alfabeto inglés. Tenemos que

$$A^+ = \{ab, abab, ababab, \ldots\} = \{(ab)^i \mid i \geq 1\}$$

el cual es a su vez un lenguaje. Podemos considerar sus potencias

$$\begin{aligned} (A^+)^2 &= A^+ \cdot A^+ \\ &= \{ab \cdot ab, ab \cdot abab, ab \cdot ababab, \ldots, \quad abab \cdot ab, abab \cdot abab, abab\cdot, ababab, \ldots, \} \end{aligned}$$

el cual es un sublenguaje de A^+: $(A^+)^2 \subseteq A^+$. De forma similar $(A^+)^3 \subseteq A^+$, $(A^+)^4 \subseteq A^+, \ldots$. Este es un hecho general.

Lema 2. *Sea A un lenguaje. Entonces*

$$(A^+)^k \subseteq A^+, \quad \forall k \geq 1$$

Proof. Por inducción sobre k. Si $k = 1$:

$$(A^+)^k = (A^+)^1 = A^+ \subseteq A^+.$$

También el resultado es cierto para $k = 2$:

$$\begin{aligned} (A^+)^2 &= A^+ \cdot A^+ \\ &= A^+ \cdot \cup_{n=1}^{\infty} A^n \\ &= \cup_{n=1}^{\infty} A^+ \cdot A^n \\ &= \cup_{n=1}^{\infty} \left(\cup_{j=1}^{\infty} A^j\right) \cdot A^n \\ &\cup_{n=1}^{\infty} \cup_{j=1}^{\infty} A^j \cdot A^n \\ &= \cup_{n=1}^{\infty} \cup_{j=1}^{\infty} A^{j+n} \\ &\subseteq \cup_{m=1}^{\infty} A^m \\ &= A^+ \end{aligned}$$

Supongamos cierto que

$$(A^+)^k \subseteq A^+ \quad .$$

Por demostrar que $(A^+)^{k+1} \subseteq A^+$. Tenemos que

$$\begin{aligned}(A^+)^{k+1} &= A^+ \cdot (A^+)^k \\ &\subseteq A^+ \cdot A^+ \\ &= (A^+)^2 \\ &\subseteq A^+\end{aligned}$$

□

Tarea 1. *Sea $x \in \Sigma^*$. Demostrar que $(x^I)^I = x$.*

Definición 17. *Si A es un lenguaje, su* **inverso** *es*

$$A^I = \{w^I \mid w \in A\}$$

Propiedad 4. *Si A, B son lenguajes, entonces*

$$(A \cdot B)^I = B^I \cdot A^I$$

Proof. Por contenciones, demostraremos que

(1) $(A \cdot B)^I \subseteq B^I \cdot A^I$
(2) $B^I \cdot A^I \subseteq (A \cdot B)^I$

(1) Sea $z \in (A \cdot B)^I$, entonces $z = x^I$ con $x \in A \cdot B$, por lo que $x = yw$ con $y \in A$ y $w \in B$. Luego

$$\begin{aligned}z &= (yw)^I \\ &= w^I y^I \in B^I \cdot A^I\end{aligned}$$

Por lo tanto $(A \cdot B)^I \subseteq B^I \cdot A^I$.

(2) Sea $z \in B^I \cdot A^I$, entonces $z = w^I \cdot y^I$ con $w \in B$ y $y \in A$. Por lo que

$$z = w^I y^I = (yw)^I \in (A \cdot B)^I \quad .$$

Por lo tanto $B^I \cdot A^I \subseteq (A \cdot B)^I$.

□

4. Numerabilidad

Nos proponemos estudiar lo siguientes problemas:

(1) Dado un lenguaje A y $x \in \Sigma^*$, ¿$x \in A^*$
(2) Dado un lenguaje A, especificar qué palabras lo componen.

Ejemplo 23. Sea $\Sigma = \{a, b\}$.

(1) ¿Cuántas palabras de longitud 0 hay?: ϵ. Sólo una.

(2) ¿Cuántas palabras de longitud 1 hay?: a, b. Dos.

(3) ¿Cuántas palabras de longitud 2 hay?: aa, ab, ba, bb. 4

(4) ¿Cuántas palabras de longitud 3 hay?: 8

(5) ¿Cuántas palabras de longitud n hay?: 2^n.

Podemos numerar las palabras de Σ^* según el siguiente orden

$$a < ab, \quad aa < ab, \quad a \leq baaaa$$

Podemos enumerar las palabras según este orden

$$\begin{aligned}
\epsilon &\mapsto 0 \\
a &\mapsto 1 \\
b &\mapsto 2 \\
aa &\mapsto 3 \\
ab &\mapsto 4 \\
ba &\mapsto 5 \\
bb &\mapsto 6 \\
aaa &\mapsto 7 \\
&\vdots
\end{aligned}$$

Sin embargo, por comodidad (para el caso general) también podemos enumerar usando números en base 3

$$\begin{aligned}
\epsilon &\mapsto 0 \\
a &\mapsto 1 \\
b &\mapsto 2 \\
aa &\mapsto 11_3 = 4 \\
ab &\mapsto 12_3 = 5 \\
ba &\mapsto 21_3 = 7 \\
bb &\mapsto 22_3 = 8 \\
aaa &\mapsto 111_3 = 13 \\
&\vdots
\end{aligned}$$

Supongamos que $\Sigma = a_1, a_2, \ldots, a_n$. Podemos enumerar las palabras de Σ^* con números en base $n+1$ como

$$\begin{aligned}
\epsilon &\mapsto 0 \\
a_1 &\mapsto 1 \\
a_2 &\mapsto 2 \\
&\vdots \\
a_n &\mapsto n \\
a_1a_1 &\mapsto 11_{n+1} \\
a_1a_2 &\mapsto 12_{n+1} \\
&\vdots
\end{aligned}$$

es decir, tenemos una función

$$f : \Sigma^* \to \mathbb{N}$$

la cual es inyectiva, pues cada numero natural tiene una única representación en base. De donde se sigue que Σ^* es enumerable.

Teorema 3. *Si Σ es un alfabeto entonces Σ^* es infinito numerable.*

En contraste todos los lenguajes que se pueden formar con Σ no es numerable. Es decir, hay mucho más lenguajes que palabras. Se puede demostrar esto, usando lo que se llama la *técnica de diagonalización*.

Teorema 4. *Sea Σ un alfabeto. El conjunto de los lenguajes sobre Σ no es numerable.*

Proof. Sea

$$\mathcal{L} = \{A \mid A \text{ es lenguaje sobre } \Sigma\} \ .$$

Procedemos por contradicción. Supongamos que $\mathcal{L}$ es numerable. Entonces

$$\mathcal{L} = \{A_0, A_1, A_2, \ldots\} \ .$$

Sabemos que Σ^* es numerable, entonces podemos poner

$$\Sigma^* = \{w_0, w_1, w_2, \ldots\}$$

definimos entonces el conjunto *diagonal*,

$$D = \{w_i \in \Sigma^* \mid w_i \notin A_i\} \subseteq \Sigma^*$$

D es un lenguaje sobre Σ^*, entonces $D \in \mathcal{L}$, por lo que debe de existir $k \in \mathbb{N}$ tal que

$$D = A_k \ .$$

Tenemos dos casos $w_k \in D$ o $w_k \notin D$.

(1) Si $w_k \in D$ entonces $w_k \notin A_k = D$, i.e., $w_k \notin D$: absurdo.

(2) Si $w_k \notin D = A_k$ entonces $w_k \in D$: absurdo de nuevo

En cualquier caso obtenemos un absurdo. Por lo tanto $\mathcal{L}$ no es numerable. $\square$

5. Lenguajes Regulares y Expresiones Regulares

Definición 18. *Sea Σ un alfabeto. El* **conjunto de los lenguajes regulares** *se define como:*

(1) *$\varnothing$ es un lenguaje regular;*

(2) *$\{\epsilon\}$ es un lenguaje regular;*

(3) *$\forall a \in \Sigma$, $\{a\}$ es un lenguaje regular;*

(4) *Si A, B son lenguajes regulares entonces*

$$A \cup B, \quad A \cdot B, \quad A^*$$

son lenguajes regulares.

Esto es, el conjunto de los lenguajes regulares sobre Σ está formado por el lenguajes vacío, los lenguajes unitarios incluidos $\{\epsilon\}$ y todos aquellos obtenidos de estos por concatenación, unión y la cerradura de Kleene de éstos.

Ejemplo 24. Sea $\Sigma = \{a, b\}$. Entonces

$$\varnothing, \{\epsilon\} \text{ son lenguajes regulares}$$

$$\{a\}, \quad \{b\} \text{ son lenguajes regulares}$$

$$\{a, b\} \text{ es lenguaje regular pues } \{a.b\} = \{a\} \cup \{b\}$$

$$\{ab\} \text{ es regular pues } \{ab\} = a \cdot \{b\}$$

$$\{a, a, b, b\} = \underbrace{\{a, b\}}_{\text{regular}} \cup \underbrace{\{ab\}}_{\text{regular}} \text{ es regular}$$

$$\{a^i \mid i \geq 0\} = \{a\}^* \text{ es regular}$$

$$\{a^i b^j \mid i \geq 0 \text{ y } j \geq 0\} = \underbrace{\{a\}^*}_{\text{regular}} \cdot \underbrace{\{b\}^*}_{\text{regular}} \text{ es regular}$$

$$\{(ab)^i \mid i \geq 0\} = \{ab\}^* \text{ es regular} \ .$$

Ejemplo 25. Sea $\Sigma = \{a, b, c\}$ y A el lenguaje sobre Σ:

$$A = \{w \in \Sigma^* \mid w \text{ no tiene a } ac \text{ como subcadena}\}$$

¿Es A regular?

Sol. Notemos que

$$\{b\}\{c\}^* \subseteq A \text{ y } \{a\}^* \subseteq A$$

luego las palabras formadas por concatenaciones de potencias a^i y bc^j están en A; i.e.,

$$(\{a\} \cup \{b\}\{c\}^*)^* \subseteq A$$

luego

$$\{c\}^*(\{a\} \cup \{b\}\{c\}^*)^* \subseteq A \ .$$

Probaremos que

$$A = \{c\}^*(\{a\} \cup \{b\}\{c\}^*)^*$$

y así resultará que A es regular. Sólo falta comprobar que

$$A \subseteq \{c\}^*(\{a\} \cup \{b\}\{c\}^*)^* \ . \tag{2}$$

Sea $w \in A$, entonces $w = c^i w'$ para algún $i \geq 0$ y w' palabra que no tiene a c como prefijo. Así, w' está formada por a's, b's y c's donde cualquier bloque de c's no puede seguir a a's, en consecuencia, cualquier bloque de c's sigue a $b's$, de donde

$$w' \in (\{a\} \cup b\{c\}^*)^*$$

entonces

$$w = c^i w' \in \{c\}^*(\{a\} \cup \{b\}\{c\}^*)^*,$$

por lo tanto

$$A = \{c\}^*(\{a\} \cup \{b\}\{c\}^*)^*$$

que es un lenguaje regular.

Las expresiones regulares se definen como sigue

Definición 19. *Sea Σ un alfabeto.*

(1) *$\varnothing$ y ϵ son expresiones regulares;*

(2) *Si $a \in \Sigma$ entonces a es una expresión regular;*

(3) *Si r y s son expresiones regulares entonces*

$$r \cup s, \ r \cdot s, \ r^*$$

son expresiones regulares.

Como en las concatenaciones, a veces escribiremos

$$rs = r \cdot s$$

Ejemplo 26. Sea $\Sigma = \{a, b, c\}$. Entonces

$$c^*(a \cup bc^*)^*$$

es una expresión regular.

Proof. Tenemos que b es una expresión regular, así como c, entonces c^* es una expresión regular por lo que bc^* también. Lo es también a, luego $a \cup bc^*$ es expresión regular y en consecuencia $(a \cup bc^*)^*$ es regular. Finalmente $c^*(a \cup bc^*)^*$ es expresión regular. $\square$

as expresiones regulares son *nombres* para los lenguajes regulares.

Definición 20. *Sea Σ un lenguaje. El lenguaje L de una expresión regular sobre Σ se define como:*

(1) $L(\varnothing) = \varnothing$, $L(\epsilon) = \{\epsilon\}$;
(2) *Si* $a \in \Sigma$, $L(a) = \{a\}$;
(3) *Si r, s son expresiones regulares entonces*
    (a) $L(r \cup s) = L(r) \cup L(s)$
    (b) $L(rs) = L(r)L(s)$
    (c) $L(r^*) = L(r)^*$

Para calcular los lenguajes de expresiones regulares se hace uso del orden de precedencia:

(1) cerraduras de Kleene: $*$
(2) concatenaciones :$\cdot$
(3) uniones: $\cup$

Ejemplo 27.

$$L(a \cup ab^*) = L(a) \cup (L(a) \cdot L(b)^*)$$

Definición 21. *Si r es una expresión regular entonces*

$$r^+ = rr^*$$

Propiedad 5.

$$L(r^+) = L(r)^+$$

Proof. Por definición

$$\begin{aligned} L(r^+) &= L(rr^*) \\ &= L(r)L(r^*) \\ &= L(r)L(r)^* \\ &= L(r)^+ \end{aligned}$$

□

Definición 22. *Sean r, s expresiones regulares sobre Σ. Se dice que r y s son equivalentes si y sólo si $L(r) = L(s)$ en tal caso se escribe $r = s$. Es decir,*

$$r = s \Leftrightarrow L(r) = L(s)$$

Notemos que $r = s \Leftrightarrow$

(1) $L(r) \subseteq L(s)$
(2) $L(s) \subseteq L(r)$

También es fácil ver que si r es una expresión regular entonces $L(r)$ es un lenguaje regular.

Ejemplo 28.

$$(a^*b)^* = \epsilon \cup (a \cup b)^* b$$

Proof. El alfabeto bajo consideración es $\Sigma = \{a, b\}$. Por definición

$$L((a^*b)^*) = (\{a\}^*\{b\})^*$$

que es el lenguaje formado por 0 o más concatenaciones de palabras de $\{a\}^*b$ esto es, palabras del tipo

$$\epsilon$$

$$a^{j_1} b \cdots a^{j_k} b$$

esto es, la palabra vacía junto con palabras que terminan en b. Este lenguaje es la descripción exactamente del lenguaje regular siguiente

$$\{\epsilon\} \cup (\{a, b\}^* \cdot \{b\}) = L(\epsilon \cup (a \cup b)^* b)$$

de donde

$$L((a^*b)^*) == L(\epsilon \cup (a \cup b)^* b)$$

por lo tanto

$$(a^*b)^* = \epsilon \cup (a \cup b)^* b$$

□

Ejemplo 29. Sea r ena expresión regular, entonces

$$r^+ = r^* r$$

Proof. Por la propiedad 5

$$\begin{aligned} L(r^+) &= L(r)^+ \\ &= L(r)^* L(r) \\ &= L(r^* r) \end{aligned}$$

lo que implica que $r^+ = r^* r$. □

El álgebra de las expresiones regulares viene descrita en el siguiente teorema.

Teorema 5. *Sean r, s, t expresiones regulares sobre Σ. Entonces*

(1) $r \cup s = s \cup r$
(2) $r \cup \varnothing = r = \varnothing \cup r$
(3) $r \cup r = r$
(4) $(r \cup s) \cup t = r \cup (s \cup t)$
(5) $r\epsilon = \epsilon r = r$
(6) $r\varnothing = \varnothing = \varnothing r$

(7) $r(st) = (rs)t$

(8) $r(s \cup t) = rs \cup rt$ *y* $(r \cup s)t = rs \cup st$

(9) $r^* = r^{**} = r^*r^* = (\epsilon \cup r)^* = r^*(r \cup \epsilon) = (r \cup \epsilon)r^* = \epsilon \cup rr^*$

(10) $(r \cup s)^* = (r^* \cup s^*)^* = (r^*s^*)^* = (r^*s)^*r^* = r^*(sr^*)^*$

(11) $r(sr)^* = (rs)^*r$

(12) $(r^*s)^* = \epsilon \cup (r \cup s)^*s$

(13) $(rs^*)^* = \epsilon \cup r(r \cup s)^*$

(14) $s(r \cup \epsilon)^*(r \cup \epsilon) \cup s = sr^*$

(15) $rr^* = r^*r$

Proof. Sólo haremos la demostración de una de estas equivalencias. La demás son similares.

(11) Por demostrar que

$$L(r(sr)^*) = L((rs)^*r) \tag{3}$$

Sea $w \in L(r(sr)^*) = L(r)(L(s)L(r))^*$ entonces

$$w = r_0 s_1 r_1 s_2 r_2 \cdots s_n r_n$$

con $r_0 \in L(r)$ y cada $s_i \in L(s)$, $r_i \in L(r)$. Podemos escribir

$$\begin{aligned} w &= (r_0 s_1) \cdot (r_1 s_2) \cdots (r_{n-1} s_{n-1}) r_n \\ &\in (L(r) \cdot L(s))^* L(r) = L((rs)^*r) \end{aligned}$$

Por lo tanto,

$$L(r(sr)^*) \subseteq L((rs)^*r) \ .$$

Similarmente se prueba que $L((rs)^*r) \subseteq L(r(sr)^*)$. Se sigue entonces que la ecuación (3) es cierta. Se concluye entonces que

$$r(sr)^* = (rs)^*r$$

□

Como un ejemplo del uso de ésta álgebra es la siguiente propiedad.

Propiedad 6. *Si* $r = s^*t$ *entonces* $r = sr \cup t$

Proof.

$$\begin{aligned} r = s^*t &= (\epsilon s^+) && \text{pues } L(s^*) = L(\epsilon) \cup L(s^+) \\ &= (\epsilon \cup ss^*)t && \text{por definición de } s^+ \\ &= \epsilon t \cup ss^*t && \text{por (5) e hipótesis} \\ &= sr \cup t && \text{por (1)} \end{aligned}$$

□

Tarea 2.

(1) *¿De que conjunto de símbolos se derivan las frases inglesas?*

(2) *¿Por qué el lenguaje vacío no es el mismo que* $\{\epsilon\}$*?*

(3) *¿Sea* $\Sigma = \{1\}$*. ¿Se puede decir que para todo número natural* n *hay alguna palabra* $w \in \Sigma^*$ *para la cual* $|w| = n$*? ¿es única? ¿Qué ocurriría si* $\Sigma = \{1, 2\}$*?*

(4) *Para una palabra* w*, ¿se puede decir que*

$$|w^{i+j}| = |w^i| + |w^j|?$$

Encontrar una expresión para $|w^{i+j}|$ *en términos de* i, j *y* $|w|$*.*

(5) *¿La cadena vacía es un prefijo de sí misma?*

(6) *Definir las nociones de sufijo y sufijo propio de una cadena sobre un alfabeto. Dar ejemplos.*

(7) *Obtener todos los prefijos, sufijos y subpalabras de la palabra* $w =$ *bar sobre el alfabeto inglés.*

Tarea 3.

(1) *Sea* $x \in \Sigma^*$*. Probar que* $(x^I)^I = x$*.*

(2) *Para un lenguaje arbitrario* A*, ¿qué es* $A \cdot \varnothing$*?*

(3) *Sean* $A = \{el, mi\}$ *y* $B = \{caballo, casa, herradura\}$ *lenguajes sobre el alfabeto ingés. Obtener* $A \cdot B$*,* $A \cdot A$ *y* $A \cdot B \cdot B$*.*

(4) *Suponer que* $A = \{\epsilon, a\}$*. Obtener* A^n *para* $n = 0, 1, 2, 3$ *¿Cuántos elementos tiene* A^n *para* n *arbitrario? ¿Cuáles son las cadenas de* A^n *para* n *arbitrario?*

(5) *Sea* $A = \{\epsilon\}$*. Obtener* A^n *para* n *arbitrario.*

(6) *Sean* $A = \{\epsilon, ab\}$ *y* $B = \{cd\}$ *¿Cuántas cadenas hay en* $A^n \cdot B$ *para* n *arbitrario?*

(7) *Sean* $A = \{a\}$*,* $B = \{b\}$*. Obtener* A^nB*,* AB^n *y* $(AB)^n$*.*

(8) *Sean* $A = \{\epsilon\}$*,* $B = \{aa, ab, bb\}$*,* $C = \{\epsilon, aa, ab\}$ *y* $D = \varnothing$ *el lenguaje vacío. Obtener* $A \cup B$*,* $A \cup C$*,* $A \cup D$ *y* $A \cap B$*,* $B \cap C$*,* $C \cap D$*,* $A \cap D$*. Suponer que* F *es un lenguaje cualquiera. Obtener* $F \cup D$ *y* $F \cap D$*.*

(9) *¿Bajo qué condiciones* $A^* = A^+$*?*

(10) *Obsérvese que para todo lenguaje* A *se tiene que* $\epsilon \in A^*$ *¿cuándo* $\epsilon \in A^+$*?*

(11) *Probar que* $\{\epsilon\}^* = \{\epsilon\} = \{\epsilon\}^+$*.*

(12) *Antes se obtuvo que* $A^* = A^0 \cup A^+ = \{\epsilon\} \cup A^+$. *Cabría esperar que* $A^+ = A - \{\epsilon\}$. *Probar que, en general, ésta expresión no es cierta. ¿Cuándo se cumplirá que* $A^+ = A^* - \{\epsilon\}$.

(13) *Obtener lenguajes* A, B, C *tales que* $A \cdot (B - C) \neq A \cdot B - A \cdot C$.

(14) *Probar que*
   (a) $(A^*)^* = A^*$
   (b) $(A^*)^+ = A^*$
   (c) $(A^+)^* = A^*$

(15) *Demostrar que se cumplen las siguientes igualdades para los lenguajes* A *y* B *sobre el alfabeto* Σ*:*
   (a) $(A \cup B)^I = A^I \cup B^I$
   (b) $(A \cap B)^I = A^I \cap B^I$
   (c) $(A^+)^I = (A^I)^+$
   (d) $(A^*)^I = (A^I)^*$

Tarea 4.

(1) *Sea* $\Sigma = \{a, b\}$. *Lo siguiente es una definición recursiva del lenguaje* A*:*
   (a) $\epsilon \in A$.
   (b) *Si* $x \in A$, *entonces* axb *y* bxa *pertenecen a* A.
   (c) *Si* x *e* y *pertencen a* A, *entonces* xy *pertenece a* A.
   (d) *No hay nada más en* A.

   Probar que
   (a)
$$A = \{w \in \Sigma^* \mid w \textit{ tiene el mismo número de aes que de bes}\}$$
   (b) *Si* b *y* ϵ *están en* A *¿qué más palabras hay en* A*?*
   (c) *Dar una definición recursiva para que* $A \subseteq \{a, b\}^*$ *contenga todas las palabras que tienen el doble de aes que bes*

(2) *Un palíndromo es una cadena que se lee igual hacia adelante que hacia atrás. Por ejemplo, la palabra "a" es un paíndromo, al igual que la cadena "radar". Dar una definición recursiva de un palíndromo (obsérvese que* ϵ *es un palíndromo).*

(3) *Probar que para los lenguajes* A *y* B, $(A \cup B)^* = (A^* B^*)^*$.

Tarea 5.

(1) *Verificar, aplicando la definición de lenguaje regular, que los siguientes son lenguajes regulares sobre* $\Sigma = \{a, b\}$*:*
   (a) $\{a^i \mid i > 0\}$.
   (b) $\{a^i \mid i > n\}$ *para* $n \geq 0$ *fijo.*
   (c) $\{w \in \Sigma^* \mid w \textit{ termina con } a\}$.

(2) *Verificar que el lenguaje de todas las cadenas de ceros y unos que tienen al menos dos ceros consecutivos, es un lenguaje regular.*

(3) *Los identificadores de Pascal son cadenas de longitud arbitraria compuestas por caracteres alfabéticos y por dígitos. Los identificadores de Pascal deben empezar con un carácter alfabético. ¿Es este un lenguaje regular?*

(4) *Obtener una expresión regular que represente el lenguaje de los identificadores de Pascal.*

(5) (a) *Probar que* $(r \cup \epsilon)^* = r^*$.
   (b) *Probar que* $(b \cup aa^*b) \cup (b \cup aa^*b)(a \cup ba^*b)^*(a \cup ba^*b)$ *y* $a^*b(a \cup ba^*b)^*$ *son equivalentes.*
   (c) *Sobre* $\Sigma = \{a, b, c\}$ *¿son equivalentes las parejas de expresiones regulares de cada apartado?*
   (d) $(a \cup b)^*a^*$ *y* $((a \cup b)a)^*$.
   (e) $\varnothing^{**}$ *y* ϵ.
   (f) $((a \cup b)c)^*$ *y* $(ac \cup bc)^*$.
   (g) $b(ab \cup ac)$ *y* $(ba \cup ba)(b \cup c)$.

(6) *Simplificar:*
   (a) $\varnothing^* \cup a^* \cup b^* \cup (a \cup b)^*$.
   (b) $((a^*b^*)^* \cdot (b^*a^*)^*)^*$.
   (c) $(a^*b)^* \cup (b^*a)^*$.
   (d) $(a \cup b)^*a(a \cup b)^*$.

(7) *Probar que* $(aa)^*a = a(aa)^*$.

(8) *Simplificar las siguientes expresiones regulares:*
   (a) $(\epsilon \cup aa)^*$.
   (b) $(\epsilon \cup aa)(\epsilon \cup aa)^*$.
   (c) $a(\epsilon \cup aa)^*a \cup \epsilon$.
   (d) $a(\epsilon \cup aa)^*(\epsilon \cup aa) \cup a$.
   (e) $(a \cup \epsilon)a^*b$.
   (f) $(\epsilon \cup aa)^*(\epsilon \cup aa)a \cup a$.
   (g) $(\epsilon \cup aa)(\epsilon \cup aa)^*(\epsilon \cup aa) \cup (\epsilon \cup aa)$.
   (h) $(\epsilon \cup aa)(\epsilon \cup aa)^*(ab \cup b) \cup (ab \cup b)$.
   (i) $(a \cup b)(\epsilon aa)^*(\epsilon \cup aa) \cup (a \cup b)$.
   (j) $(aa)^*a \cup (aa)^*$.
   (k) $a^*b((a \cup b)a^*b)^* \cup a^*b$.
   (l) $a^*b((a \cup b)a^*b)^*(a \cup b)(aa)^* \cup a(aa)^* \cup a^*b((a \cup b)a^*b)^*$.

6. Autómatas finitos deterministas

Nuestro problema principal es determinar si una palabra pertenece ó no a un lenguaje. Por ejemplo, si A es el lenguaje de la expresión regular $c^*(a \cup bc^*)^*$

entonces

$$abc^5c^3ab \in A, \quad cabac^3bc \notin A,$$

el análisis se puede hacer letra por letra según sus posiciones. Para ayudar a tal anásisis se hace uso de *grafos dirigidos* llamados *diagramas de transición*. Los nodos de tales grafos se llaman *estados*, las flechas se llaman *transiciones* y se etiquetan éstas flechas con símbolos del alfabeto.

Hay símbolos especiales: *estado inicial* que se marca con $\rightarrow$ y *estados finales* ó de *aceptación* que se marcan con un círculo: $\odot$

Por ejemplo:

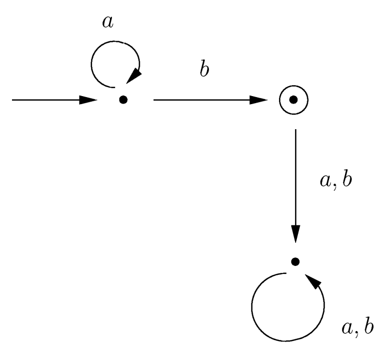
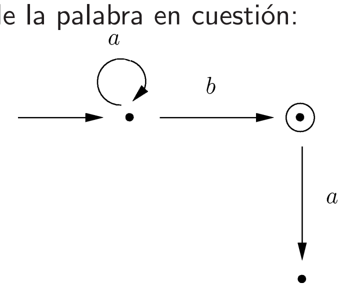


Figure 1. Un autómata finito determinista

Tales grafos se llaman *autómatas finitos deterministas* (AFD).

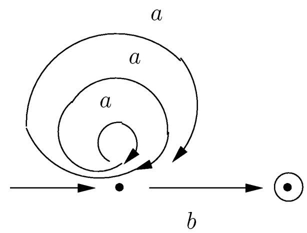
Una palabra se dice *aceptada* ó *legal* con respecto a un AFD si partiendo del estado marcado como inicial, se llega a un estado de aceptación mediante el siguiente procedimiento:

¿la cadena aba es aceptada por el AFD de la figura 6? comenzando del nodo marcado como estada inicial seguimos el camino indicado por las flechas con etiquetas las letras de la palabra en cuestión:



se arriba entonces a un estado que no es de aceptación, por lo que la palabra aba se rechaza.

¿a^3b es aceptada? veamos el diagrama:



como se puede notar, tal palabra nos hace llegar al estado de aceptación, por lo que la palabra a^3b es aceptada.

De donde es claro que el autómata finito determinista de la figura 6 acepta las palabras del lenguaje de a^*b.

Ejemplo 30. Sea $\Sigma = \{a, b\}$. Consideremos el lenguaje

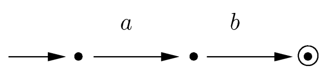
$$A = \{(ab)^i \mid i \geq 1\} \ .$$

Construir un AFD que acepte únicamente a las palabras de A.

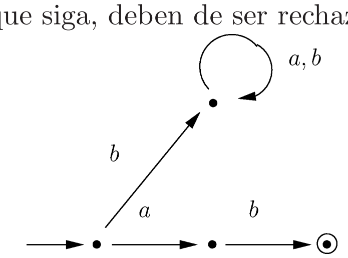
Sol. Recordemos que

$$A = \{ab, abab, ababab, \dots\}$$

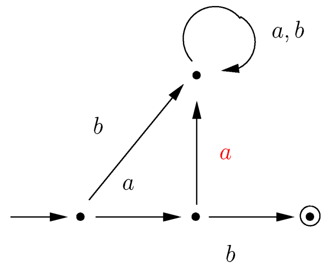
de donde al menos la palabra ab debe, en el autómata que construyamos, conducir a un estado de aceptación. Lo que sugiere que consideremos el diagrama



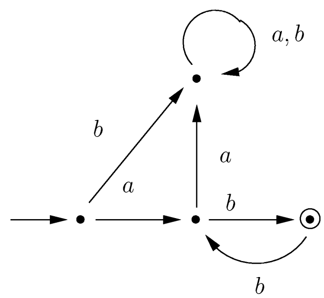
éste aún no es un AFD, puesto que se requiere que el autómata, en cada estado, sepa a qué estado nuevo se transita ante la aparición de cualquier letra del alfabeto, en nuestro caso a y b. Notemos que en nuetro primer diagrama el autómata, en el estado inicial no sabrá que hacer si aparece una b. Ninguna palabra de A tiene prefijo b, luego las palabras que comiencen con b, sin importar lo que siga, deben de ser rechazadas. Esto sugiere:



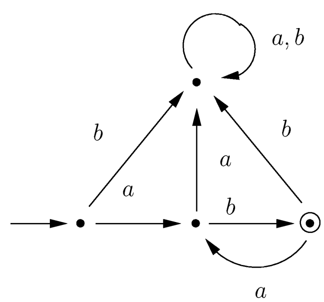
Otras palabras a rechazar son aquellas que después de a, en lugar de continuar con b, continuen con a, lo que sugiere



Otras palabras que deben de ser aceptadas son por ejemplo $abab$, $ababab$. La manera de crear repeticiones es introducir ciclos en el grafo:



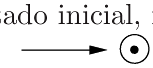
con lo cual aceptamos las palabras del tipo $(ab)^+$. Pero aún debemos rechazar las palabras que al tener prefijo ab continuen con b:



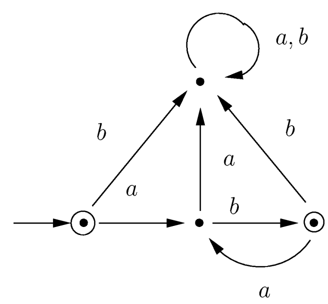
lo que completa nuestro autómata finito determinista que acepta solamente las palabras del lenguaje de $(ab)^+$.

Ejemplo 31. Lo mismo que el anterior para $A = (ab)^*$.

Sol. Ahora la palabra vacía también tiene que ser aceptada. La técnica para aceptar a ϵ es hacer al estado inicial, final también:

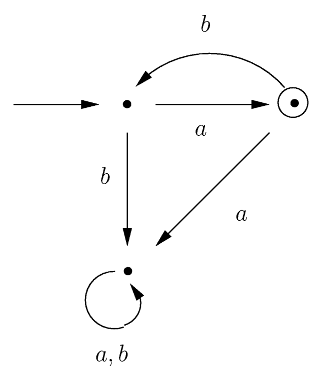


Por lo que el autómata pedido es

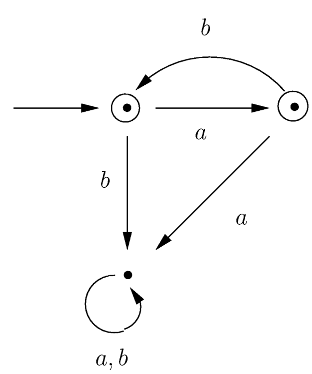


Nótese que ahora tenemos *dos* estados finales.

En general cuando en un AFD se hace del estado inicial un estado final, no sólo se va ha aceptar a la palabra vacía, puede que se acepten otras indeseables. Por ejemplo, en el AFD,

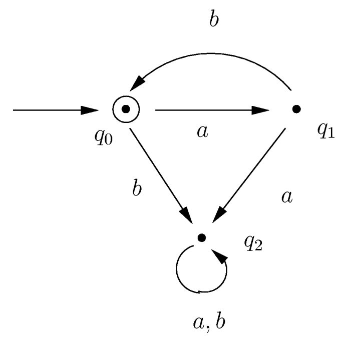


se acepta sólo a al lenguaje $a(ba)^*$ y no a la palabra vacía. Si ponemos al estado inicial como final obtenemos



que acepta no sólo a la palabra vacía ϵ, sino que se "cuela" todo el lenguaje $(ab)^*$. Es decir, el nuevo autómata acepta a $(ab)^* \cup a(ba)^*$.

Ejemplo 32. A veces, el útil etiquetar los estados. Por ejemplo



Entonces se puede representar la *dinámica* de los estados mediante una tabla

δ	a	b
q_0	q_1	q_2
q_1	q_2	q_0
q_2	q_2	q_2

Table 1. Tabla de transiciones

que es una forma de representar a una función $\delta : Q \times \Sigma \to Q$, donde $Q = \{q_0, q_1, q_2\}$ el el conjunto de estados.

Formalmente, un AFD es:

Definición 23 (AFD)**.** *Un* **autómata finito determinista** M *es una 5-upla:*

$$M = (Q, \Sigma, s, F, \delta)$$

donde

(1) $Q = \{q_0, q_1, \dots, q_n\}$ *es un conjunto finito de elementos llamados* **estados***.*

(2) Σ *es un alfabeto.*

(3) $s \in Q$ *un elemento llamado* **estado final***.*

(4) $F \subseteq Q$ *un subconjunto de estados llamados* **estados finales***.*

(5) *Una función* $\delta : Q \times \Sigma \to Q$*, donde* $\delta(q_i, \sigma)$ *es el estado siguiente a* q_i*.*

Ejemplo 33. En el ejemplo inmediato anterior , el autómata finito determinista es:

(1) $Q = \{q_0, q_1, q_2\}$

(2) $\Sigma = \{a, b\}$

(3) $s = q_0$

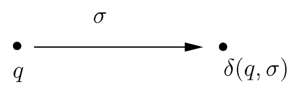
(4) $F = \{q_0\}$

y δ es la función definida por la tabla 1.

Recíprocamente, dado M un AFD, $M = (Q, \Sigma, s, F, \delta)$, se puede construir su diagrama de transiciones como:

(1) nodos: $q \in Q$

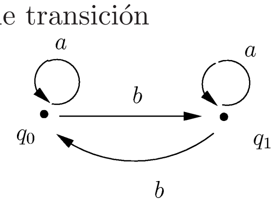
(2) flechas: si $q \in Q$ y $\sigma \in \Sigma$, entonces se pone



Ejemplo 34. Para el autómata $M = (Q, \Sigma, s, F, \delta)$ con $Q = \{a, b\}$, $\Sigma = \{a, b\}$, $s = q_0$, $F = \{q_0\}$ y δ definida por

δ	a	b
q_0	q_0	q_1
q_1	q_1	q_0

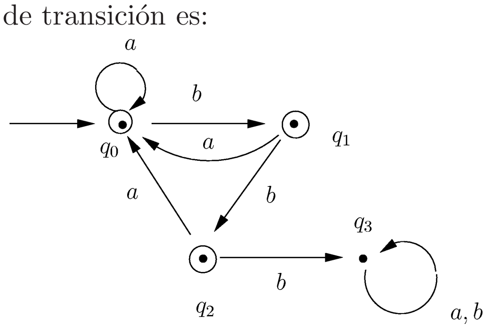
le corresponde diagrama de transición



Ejemplo 35. Sea $M = (Q, \Sigma, s, F, \delta)$ con $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$, $s = q_0$, $F = \{q_0, q_1, q_2\}$ y tabla de transiones

δ	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_0	q_3
q_3	q_3	q_3

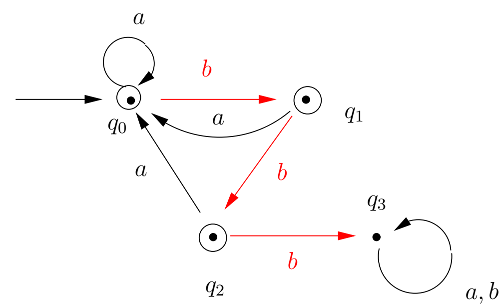
luego el diagrama de transición es:



Definición 24. *Sea M un AFD. El* **lenguaje aceptado por** *M es*

$$L(M) = \{w \in \Sigma^* \mid w \text{ es aceptada por } M\}$$

Ejemplo 36. Consideremos M como en el ejemplo inmediato anterior. Puede notarse que todos los estados son de aceptación excepto uno; luego todas las palabras son aceptadas excepto cuando se llega a q_3. Y la única forma de llegar a q_3 es con tres b's consecutivas:



esto es

$$L(M) = \{w \in \Sigma^* \mid w \text{ no tiene a } b^3 \text{ como subpalabra}\}$$

Si $\sigma_1, \sigma_2, \sigma_3 \in \Sigma$ y q_0 es estado inicial, el estado resultante de analizar la cadena es, formalmente,

$$\delta(\delta(\delta(q_0, \sigma_1), \sigma_2), \sigma_3)$$

esta aplicación se abreviará como

$$\delta(q_0, \sigma_1\sigma_2\sigma_3)$$

más generalmente:

Definición 25. *Sea q_i un estado. Se definen*

(1) $\delta(q_i, \epsilon) = q_i$

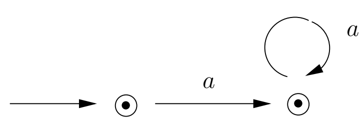
(2) *Si $w \in \Sigma^*$ y $w = aw'$ con $a \in \Sigma$ y $w' \in \Sigma^*$, entonces*

$$\delta(q_, aw') = \delta(\delta(q_i, a), w')$$

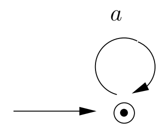
Definición 26. *Sean M_1, M_2 dos AFD. Se dice que M_1 es equivalente a M_2 si*

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

Ejemplo 37. Pongamos $\Sigma = \{a\}$. Sea M_1 el autómata finito determinista dado por



y M_2 el dado por



Es fácil ver que $L(M_1) = a^*$. También $L(M_2) = a^*$. Luego $L(M_1) = L(M_2)$ y así M_1 es equivalente a M_2

Tarea 6.

(1) *Obtener la expresión regular que representa el lenguaje formado por todas las cadenas sobe* $\{a, b\}$ *que tienen un número par de bes. Construir el diagrama de transición para este lenguaje.*

(2) *Construir el diagrama de transición para el lenguaje dado por* $c^*(a \cup bc^*)^*$*. Convertir el diagrama en una tabla, etiquetando los estados* $q_0, q_1, \ldots$

(3) *Sea* $M = (Q, \Sigma, s, F, \delta)$ *dado por*

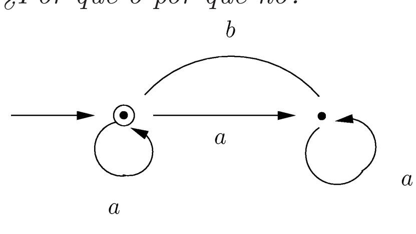
$$Q = \{q_0, q_1, q_2, q_3\}$$
$$\Sigma = \{0, 1\}$$
$$F = \{q_0\}$$
$$s = q_0$$

y δ *dada por la tabla*

δ	*0*	*1*
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

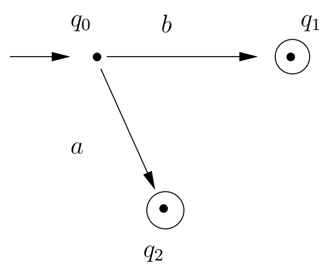
Construir el diagrama de transición. Obtener la secuencia de estados por lo que se pasa para aceptar la cadena 110101 *(el carácter del extremo izquierdo es el primero en ser analizado).*

(4) *¿ La siguiente figura es un diagrama de transición correspondiente a un AFD? ¿Por qué o por qué no?*

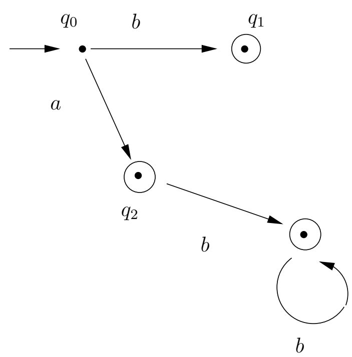


7. Automatas finitos no deterministas

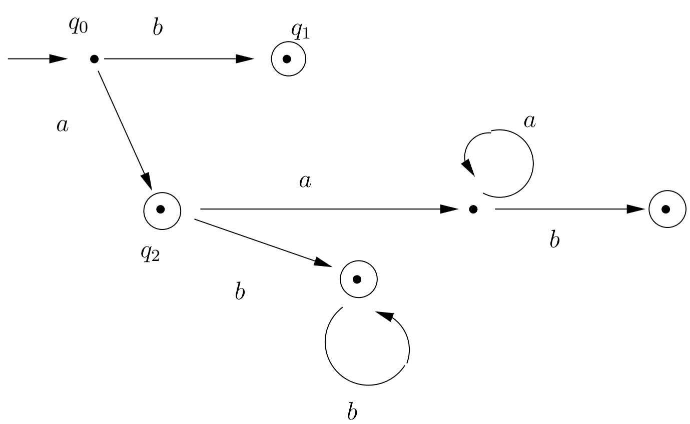
Ejemplo 38. Diseñemos un AFD que sólo acepte a $a^*b \cup ab^*$. Notemos que el lenguaje $a^*b \cup ab^*$ está formado por las palabras w que tienen sufijo b o prefijo a. Las palabras a y b deben de ser aceptadas, lo que sugiere



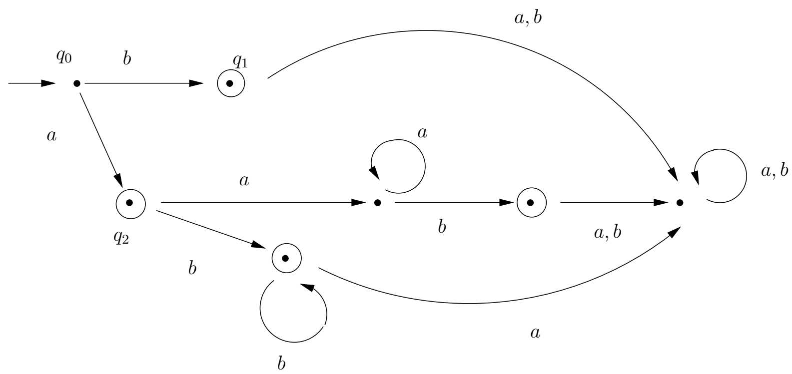
también las palabras que después de a le siguen alguna potencia de b deben de ser aceptadas. Por lo que añadimos



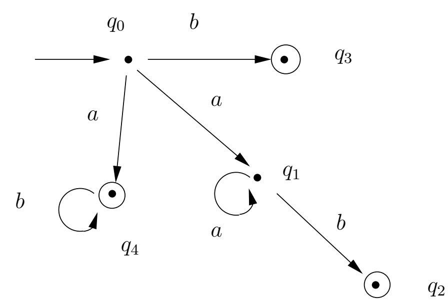
También las palabras del tipo $a^n b$ con $n \geq 2$ deben de ser aceptadas; se añade



Cualesquiera otras palabras diferentes a las de los modelos anteriores deben de ser rechazadas



Nótese que no es claro que el lenguaje $a^*b \cup ab^*$ sea exactamente el aceptado por éste autómata. Sería más fácil si se permitiera:



pero éste no en un AFD sino un *autómata finito no determinista.*

Definición 27. *Un* **autómata finito no determinista** *(AFN) es $M = (Q, \Sigma, s, F, \Delta)$ donde*

(1) $Q = \{q_0, \dots, q_n\}$ *es conjunto finito de estados.*

(2) $|Sigma$ *un alfabeto.*

(3) $s \in Q$ *estado inicial.*

(4) $F \subset Q$ *estados finales.*

(5) Δ *es una relación de $Q \times \Sigma$ en Q, es decir,*

$$\Delta \subseteq (Q \times \Sigma) \times Q$$

Lo anterior significa que Δ no es una función, pero casi lo es; queremos decir, que si $q \in Q$ y $\sigma \in \Sigma$, entonces $\Delta(q, \sigma)$ no es un sólo elemento, sino todo un conjunto:

$$\Delta(q, \sigma) \subseteq Q \ .$$

Ejemplo 39. En el autómata finito no determinista anterior (ejemplo 38) tenemos que

$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$
$$\Sigma = \{a, b\}$$
$$s = q_0$$
$$F = \{q_2, q_3, q_4\}$$

y Δ está descrita por lo siguiente tabla:

Δ	a	b
q_0	$\{q_1, q_4\}$	$\{q_3\}$
q_1	$\{q_2\}$	q_2
q_2	$\varnothing$	$\varnothing$
q_3	$\varnothing$	$\varnothing$
q_4	$\varnothing$	$\{q_4\}$

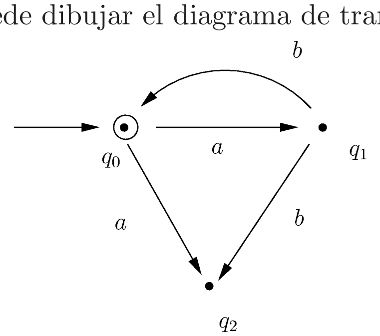
Ejemplo 40. Sea $M = (Q, \Sigma, s, F, \Delta)$ un AFN dado por

$$Q = \{q_0, q_1, q_2\}, \quad \Sigma = \{a, b\}$$
$$s = q_0, \quad F = \{q_0\}$$

y relación de transición dada por

Δ	a	b
q_0	$\{q_1\}$	$\varnothing$
q_1	$\varnothing$	$\{q_0, q_2\}$
q_2	$\{q_0\}$	$\varnothing$

Con estos datos se puede dibujar el diagrama de transición:



de donde se puede ver que M acepta a $(ab)^*$ y también a $(aba)^*$. Aún más, acepta a $((ab)^*(aba)^*)^* = (ab \cup aba)^*$. Se puede mostrar que

$$L(M) = ((ab)^*(aba)^*)^*$$

Después, basados en el lema de Arden, daremos un algoritmo para comprobar igualdades de este tipo.

La definición formal de las palabras aceptadas es:

Definición 28. *$w \in \Sigma^*$ es* **aceptada** *si $\Delta(s,w)$ contiene al menos un estado de aceptación, i.e., si*

$$\Delta(s,w) \cap F \neq \varnothing \ .$$

Definición 29. *Sea M un AFN. El* **lenguaje aceptado por** *M es*

$$L(M) = \{w \in \Sigma^* \mid w \text{ es aceptada por } M\}$$

Las transiciones de estados pueden describirse de forma similar a los AFD con δ.

Definición 30. *Sea $M = (Q, \Sigma, s, F, \Delta)$ un AFN.*

(1) *Si $X \subseteq Q$ y $\sigma \in Q$ se define*

$$\Delta(X,\sigma) = \begin{cases} \varnothing, & \text{si } X = \varnothing \\ \bigcup_{q \in X} \Delta(q,\sigma), & \text{si } X \neq \varnothing \end{cases}$$

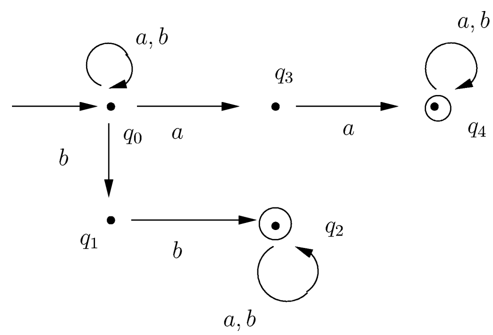
(2) *Si $w \in \Sigma^*$ con $w = \sigma w'$ con $\sigma \in \Sigma$ y $|w'| > 0$ entonces*

$$\Delta(q,w) = \Delta(\Delta(q,\sigma), w')$$

Ejemplo 41. *Si $\Sigma = \{a,b\}$;*

$$\begin{aligned} \Delta(q_0, abaab) &= \Delta(\Delta(q_0,a), baab) \\ &= \Delta(\Delta(\Delta(\Delta(\Delta(q_0,a),b),a),a),b) \end{aligned}$$

Ejemplo 42. Consideremos M el AFN con alfabeto $\Sigma = \{a,b\}$ y diagrama de transición



entonces

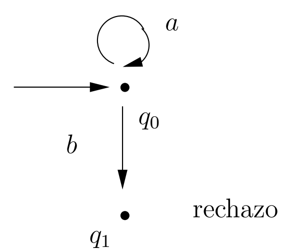
Δ	a	b
q_0	$\varnothing$	$\{q_2\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_4\}$	$\varnothing$
q_3	$\{q_4\}$	$\{q_4\}$

y así,

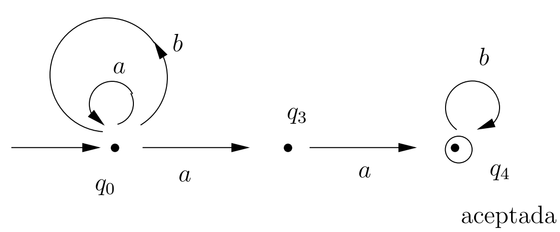
$$\begin{aligned}\Delta(q_0, ab) &= \Delta(\Delta(q_0, a), b) \\ &= \Delta(\{q_0, q_3\}, b) \\ &= \Delta(q_0, b) \cup \Delta(q_3, b) \\ &= \{q_0, q_3\} \cup \varnothing \\ &= \{q_0, q_3\}\end{aligned}$$

que son los posibles estados que se obtienen a partir de q_0 con transición ab.

Examinemos la palabra $abaab$:



por lo que $abaab$ *aún* no se acepta. Pero



lo que nos lleva a un estado de aceptación. De aquí que $abaab$ se acepta, i.e.,

$$abaab \in L(M)$$

Estos diagramas realmente corresponden a las siguientes ecuaciones

$$\begin{aligned}
\Delta(q_0, abaab) &= \Delta(\Delta(q_0, a), baab) \\
&= \Delta(\{q_0, q_3\}, baab) \\
&= \Delta(q_0, baab) \cup \Delta(q_3, baab) \\
&= \Delta(\Delta(q_0, b), aab) \cup \Delta(\Delta(q_3, b), aab) \\
&= \Delta(\{q_0, q_1\}, aab) \cup \underbrace{\Delta(\varnothing, aab)}_{\varnothing} \\
&= \Delta(q_0, aab) \cup \Delta(q_1, aab) \\
&= \Delta(\Delta(q_0, a), ab) \cup \Delta(\Delta(q_1, a), ab) \\
&= \Delta(\{q_0, q_3\}, ab) \cup \Delta(\varnothing, ab) \\
&= \Delta(q_0, ab) \cup \Delta(q_3, ab) \\
&= \Delta(\Delta(q_0, a), b) \cup \Delta(\Delta(q_3, a), b) \\
&= \Delta(\{q_0, q_3\}, b) \cup \Delta(q_4, b) \\
&= \{q_0, q_1\} \cup \varnothing \cup \{q_4\} \\
&= \{q_0, q_1, q_4\}
\end{aligned}$$

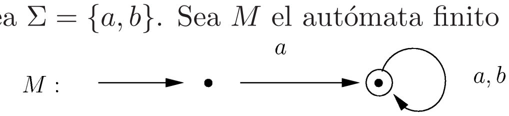
que contiene al estado de aceptación $q_4 \in F$, por lo que, como antes observamos, $abaab \in L(M)$.

8. Equivalencia entre AFD y AFN

Lo que realmente importa de los autómatas no es su diagrama de transición, sino el lenguaje que aceptan.

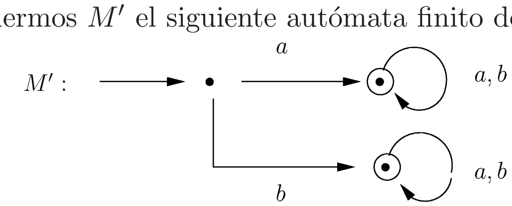
Definición 31. *Sea M un AFD ó AFN, sea M' un AFD ó AFN. Se dice que M es* **equivalente** *a M' si $L(M) = L(M')$*

Ejemplo 43. Sea $\Sigma = \{a, b\}$. Sea M el autómata finito no determinista



es fácl ver que $L(M) = a(a \cup b)^*$.

Ahora considermos M' el siguiente autómata finito determinista



También tenemos que $L(M') = a(a \cup b)^*$. Por lo tanto M es equivalente a M'.

En general, si M es un AFD, entonces es un AFN, pues δ función es en particular una relación. Queremos probar lo recíproco; esto es, si M es un AFN entonces existe M' un AFD equivalente a M.

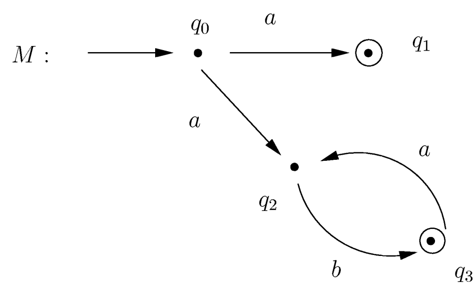
La idea es la siguiente: como Δ es una relación, entonces

$$\Delta(q, \sigma) = \{q_{i_1}, \ldots, q_{i_s}\}$$

esto es, Δ induce una función, no de estados a estados, sino de conjuntos de estados a conjuntos de estados:

$$\Delta' : E \times \Sigma \to E, \quad E \subseteq 2^Q$$

Ejemplo 44. Sea M el AFN

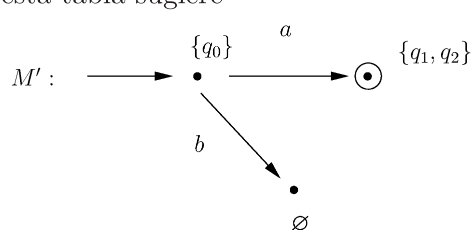


Tenemos que $L(M) = a \cup (ab)^+$.

Construiremos un AFD M' tal que $L(M') = a \cup (ab)^+$. Resulta que

Δ	a	b
q_0	$\{q_1, q_2\}$	$\varnothing$
q_1	$\varnothing$	$\varnothing$
q_2	$\varnothing$	$\{q_3\}$
q_3	$\{q_2\}$	$\varnothing$

La primera fila de esta tabla sugiere

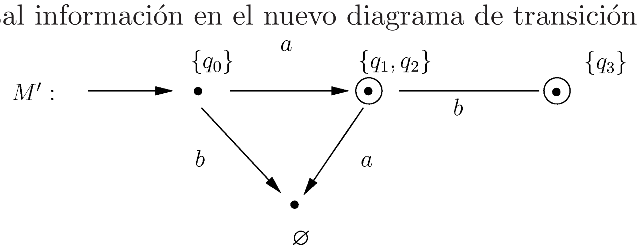


donde hay nuevos estados marcados por $\{q_0\}$, $\{q_1, q_2\}$ y $\varnothing$. Necesitamos calcular las transiciones de estos nuevos estados. Las del estado $\{q_1, q_2\}$ son

$$\begin{aligned} \Delta(\{q_1, q_2\}, a) &= \Delta(q_1, a) \cup \Delta(q_2, b) \\ &= \varnothing \cup \varnothing = \varnothing \end{aligned}$$

$$\begin{aligned} \Delta(\{q_1, q_2\}, b) &= \Delta(q_1, b) \cup \Delta(q_2, b) \\ &= \varnothing \cup \{q_3\} = \{q_3\} \end{aligned}$$

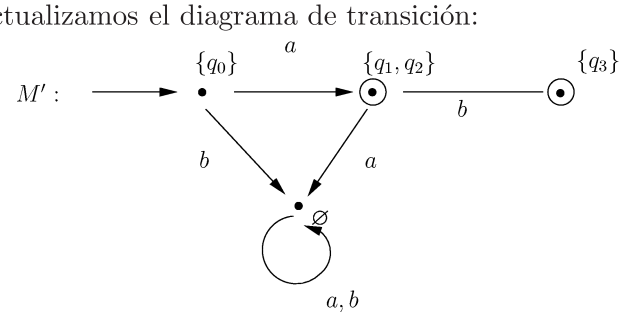
Agregamos tal información en el nuevo diagrama de transición:



Las transiciones desde $\varnothing$ son hacia $\varnothing$:

$$\Delta(\varnothing, a) = \varnothing, \quad \Delta(\varnothing, b) = \varnothing \ .$$

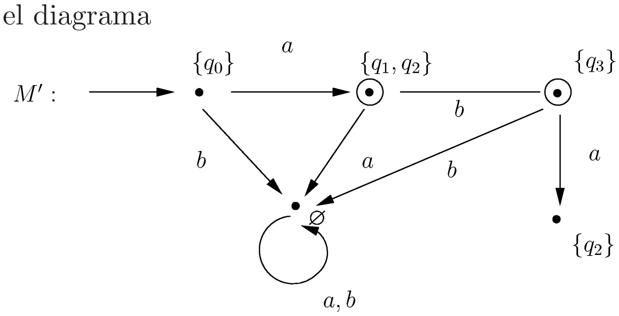
De nuevo, actualizamos el diagrama de transición:



Ahora necesitamos calcular las transiciones del nuevo estado $\{q_3\}$:

$$\Delta(\{q,3\}, a) = \{q_2\}, \quad \Delta(q_3, b) = \{q_3\}$$

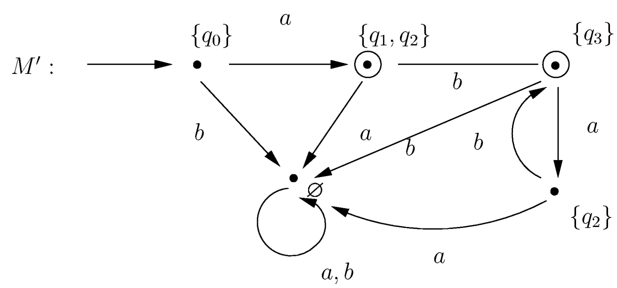
queda ahora el diagrama



Finalmente, necesitamos las transiciones del nuevo estado $\{q_2\}$:

$$\Delta(\{q_2\}, a) = \varnothing, \quad \Delta(\{q_2\}, b) = \{q_3\}$$

lo que completa la construcción de M' que es un AFD:



el cual acepta el lenguaje

$$L(M') = a \cup b \cup (ab)^+ \ .$$

Nótese que los nuevos estados iniciales on aquellos que contienen a estados inicales del autómata original.

Hemos construido un nuevo AFD $M' = (Q', \Sigma', s', F', \delta')$ donde

$$Q' = \{\{q_0\}, \varnothing, \{q_1, q_2\}, \{q_3\}, \{q_2\}\}$$
$$\Sigma' = \Sigma \text{ el alfabeto inicial}$$
$$s' = \{s\} F' = \{\{q_1, q_2\}, \{q_3\}\}$$

y la función de transición es

δ'	a	b
$\{q_0\}$	$\{q_1, q_2\}$	$\varnothing$
$\varnothing$	$\varnothing$	$\varnothing$
$\{q_1, q_2\}$	$\varnothing$	$\{q_3\}$
$\{q_3\}$	$\{q_2\}$	$\varnothing$
$\{q_2\}$	$\varnothing$	$\{q_3\}$

Donde M' es equivalente a M.

Teorema 6. *Sea* $M = (Q, \Sigma, s, F, \Delta)$ *un AFN. Entonces existe* $M' = (Q', \Sigma', s', F', \delta')$ *un AFD equivalente a* M*.*

Proof. Sea 2^Q el conjunto potencia de Q, esto es 2^Q es la colección de todos los subconjuntos de Q. Se define M' como sigue:

$$Q' = 2^Q$$
$$\Sigma' = \Sigma$$
$$s' = \{s\}$$
$$F' = \{S \subseteq Q \,|\, S \cap F \neq \varnothing\}$$

$$\delta' : Q' \times \Sigma \to \Sigma, \quad \delta'(S, \sigma) = \Delta(S, \sigma)$$

Por demostrar que $L(M') = L(M)$. Probaremos que

$$w \in \Sigma^* \text{ es aceptada por } M' \Leftrightarrow w \text{ es aceptada por } M$$

Una cadena $w \in \Sigma^*$ es aceptada por $M' \Leftrightarrow \delta'(s', w)$ es un estado de aceptación de $M' \Leftrightarrow \delta(s', w) \cap F \neq \varnothing \Leftrightarrow \Delta(s, w) \cap F \neq \varnothing \Leftrightarrow w \in L(M)$. □

Tarea 7. *El las siguientes tablas, los estados iniciales están marcados con una flecha y los estados finales con un asterisco:*

(1) *Convertir el siguiente AFN a AFD:* $\Sigma = \{0, 1\}$

Δ	0	1
$\to p$	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	$\varnothing$
$*s$	$\{s\}$	$\{s\}$

(2) *Convertir el siguiente AFN a AFD:* $\Sigma = \{0, 1\}$

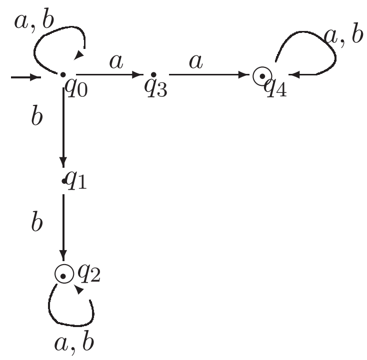
Δ	0	1
$\to p$	$\{q, s\}$	$\{q\}$
$*q$	$\{r\}$	$\{q, r\}$
r	$\{s\}$	$\{p\}$
$*s$	$\varnothing$	$\{p\}$

(3) *Convertir el siguiente AFN a AFD y describir informalmente el lenguaje que acepta:* $\Sigma = \{0, 1\}$*:*

Δ	0	1
$\to p$	$\{p, q\}$	$\{p\}$
q	$\{r, s\}$	$\{t\}$
r	$\{p, r\}$	$\{t\}$
$*s$	$\varnothing$	$\varnothing$
$*t$	$\varnothing$	$\varnothing$

Tarea 8.

(1) *Calcule todas las transiciones (desde el estado inicial) dadas por las cadenas babba y aabaaaba para determinar si son aceptadas por el autómata*

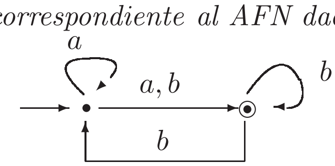


(2) *Sea M el AFN dado por $Q = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $s = q_0$, $F = \{q_1\}$ y Δ dada por*

Δ	a	b
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	$\varnothing$	$\{q_0, q_1\}$

determinar si a^2b, ba y b^2a están en $L(M)$. Dibujar el diagrama de transición para M.

(3) *Construir el AFD correspondiente al AFN dado por*



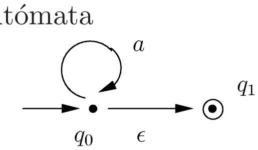
¿qué lenguaje acepta dicho autómata?

(4) *Supongamos que M es un AFN que ya es determinista. ¿Qué se obtendrá si tratamos de convertirlo en un AFD, según el algoritmo expuesto?*

9. ϵ-transiciones

Se puede extender la definición de los AFN para incluir transiciones que no dependan de ninguna entrada y sin consumir ningún símbolo. Tales se llaman *ϵ-transiciones.*

Ejemplo 45. Sea M el autómata



entonces $a^2 \in L(M)$ pues

$$\begin{aligned}
\Delta(q_0, a^2) &= \Delta(\Delta(q_0, a), a) \\
&= \Delta(\{q_0\}, a) \\
&= \Delta(\{q_0\}, a\epsilon) \\
&= \Delta(\Delta(\{q_0\}, a), \epsilon) \\
&= \Delta(\{q_0\}, \epsilon) \\
&= \{q_1\}
\end{aligned}$$

Hemos usado que $a^2 = a^2\epsilon$.

Pero también $a^2 = a\epsilon a$; así

$$\begin{aligned}
\Delta(q_0, a^2) &= \Delta(q_0, a\epsilon a) \\
&= \Delta(\Delta(q_,0, a), \epsilon a) \\
&= \Delta(\Delta(q_0, a), \epsilon a) \\
&= \Delta(\{q_0\}, \epsilon a) \\
&= \Delta(\Delta(\{q_0\}, \epsilon), a) \\
&= \Delta(\{q_1\}, a) \\
&= \varnothing
\end{aligned}$$

También $a^2 = \epsilon a \epsilon a$ ó $a^2 = aa\epsilon$, etcétera. Así, siempre en cualquier palabra w se puede introducir ϵ y en su análisis de aceptación w puede no consumir, por ϵ, ningún símbolo del alfabeto.

El precio a pagar por permitir tales ϵ-transiciones es la indefinición de los estados siguientes. Por ejemplo, en los cáculos anteriores obtuvimos que $\Delta(q_0, a^2) = \{q_1\}$ y también que $\Delta(q_0, a^2) = \varnothing$. ¿Cuáles son entonces los estados siguientes? Se puede resolver tal indefinición si se definen los estados siguientes de manera más cuidadosa.

Definición 32. *Un* **AFN con ϵ-transiciones** M *es* $M = (Q, \Sigma, s, F, \Delta)$ *donde*

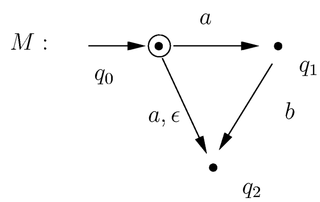
(1) Q *es un conjunto finito (de estados).*

(2) $s \in Q$ *estado inicial*

(3) $F \subseteq Q$ *estados de aceptación*

(4) Δ *es una relación de* $Q \times (\Sigma \cup \{\epsilon\})$ *en* $Q.e$

Ejemplo 46. Sea el autómata

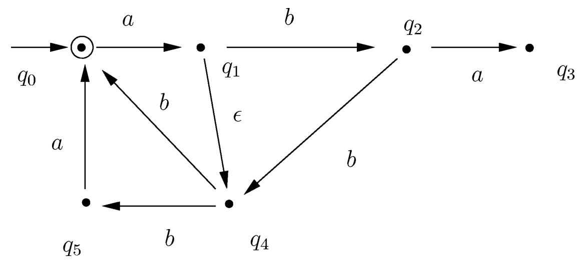


M es un AFN con ϵ-transiciones. Nótese que se puede transitar del estado q_2 a q_0 sin consumir ninguna letra del alfabeto, por lo que ab es aceptada por M. Aún más, los estados siguientes a q_0 con entrada ab deben de ser $\{q_0, q_2\}$.

Se puede poner Δ en una tabla:

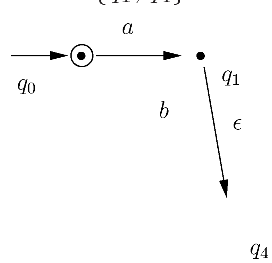
Δ	a	b	ϵ
q_0	$\{q_1\}$	$\varnothing$	$\varnothing$
q_1	$\varnothing$	$\{q_2\}$	$\varnothing$
q_2	$\{q_2\}$	$\varnothing$	$\{q_0\}$

Para obtener los estados siguientes a un estado dado se deben de tener en cuenta a los estados siguientes de las ϵ-transiciones. Por ejemplo

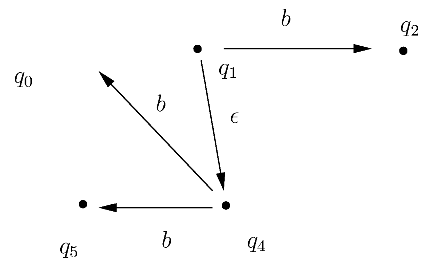


Los estados siguientes a q_0 con entrada a son

$$\{q_1, q_4\}$$



mientras que los estados siguentes a q_1 con entrada b son $\{q_2, q_0, q_5\}$



En general, se pueden calcular los estados siguientes con lo siguiente:

Definición 33. *Sea q un estado. La* **ϵ-cerradura** *de q es*

$(\epsilon{-}c)(q) = \{p \,|\, p$ *es accesible desde q sin consumir ningún símbolo de Σ en la entrada*$\}$

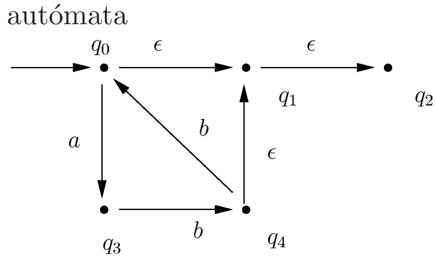
Si $q_{i_1}, \ldots q_{i_n}$ son estados se define

$$(\epsilon - c)\{q_{i_1}, \ldots q_{i_n}\} = \bigcup_{k=1}^{n} (\epsilon - c)(q_{i_k})$$

Por definición, todo estado es accesible desde sí mismo sin consumir ningún símbolo de entrada. Esto es,

$$\forall q \in Q, q \in (\epsilon - c)(q)$$

Ejemplo 47. En el autómata



entonces

$$(\epsilon - c)(q_3) = \{q_3\}$$
$$(\epsilon - c)(q_0) = \{q, q_1, q_2\}, \quad (\epsilon - c)(q_4) = \{q_4, q_1, q_2\}$$

Definición 34. *Sea q un estado y $\sigma \in \Sigma$. Se definen los* **estados que siguen directamente a q pasando por** *σ como el conjunto*

$$d(q, \sigma) = \{p \in Q \,|\, \exists \text{ una transciión de } q \text{ a } p \text{ etiquetada por } \sigma\}$$

y si $q_{i_1}, \ldots, q_{i_k}$ son varios estados, se define

$$d(\{q_{i_1}, \ldots, q_{i_k}\}, \sigma) = \bigcup_{j=1}^{k} d(q_{i_j}, \sigma)$$

Ejemplo 48. *En el AFN del ejemplo anterior* **??** *tenemos que*

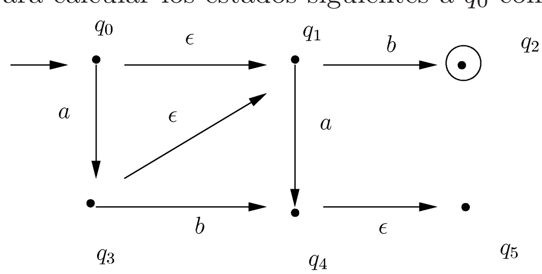
$$d(q_0, a) = \{q_3\}, \quad d(\{q_3, q_4\}, b) = d(q_3, b) \cup d(q_4, b) = \{q_4, q_0\}$$
$$d(q_0, b) = \varnothing$$

Notemos que

- $(\epsilon - c)(d(q, \sigma))$ son los estados accesibles desde q tomando primero una transición sobre σ y luego tomando una o más ϵ-transiciones.
- $d((\epsilon - c)(q), \sigma)$ son los estados accesibles desde q tomando una o más ϵ-transiciones y luego una transición sobre σ.
- $(\epsilon - c)(d((\epsilon - c)(q), \sigma))$ son los estados accesibles desde q primero tamando una o más ϵ-transiciones luego siguiendo con una transción σ y luego tomando una o más ϵ-transiciones. Así:

  $(\epsilon - c)(d((\epsilon - c)(q), \sigma))$ *son los estados siguientes a* q *con entrada* σ*.*

Ejemplo 49. Para calcular los estados siguientes a q_0 con entrada a,



primero calculamos su ϵ-cerradura:

$$(\epsilon - c)(q_0) = \{q_0, q_1\}$$

ensieguida los estados que siguen directamente pasando por a

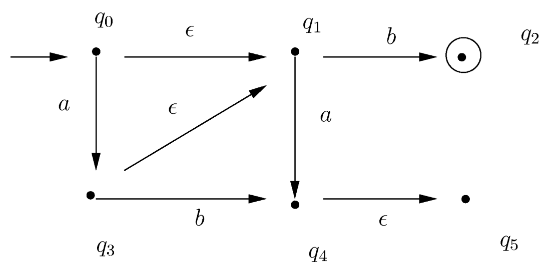
$$d((\epsilon - c)(q_0), a) = d(q_0, a) \cup d(q_1, a) = \{q_3, q_4\}$$

y finalmente la ϵ-cerradura de éstos:

$$\begin{aligned}(\epsilon - c)(d((\epsilon - c)(q_0), a)) &= (\epsilon - c)(q_3) \cup (\epsilon - c)(q_4) \\ &= \{q_3, q_1\} \cup \{q_4, q_4\} \\ &= \{q_1, q_3, q_4, q_5\}\end{aligned}$$

A partir de un AFN M con ϵ-transiciones se puede definir un AFN M' sin ϵ-transiciones tal que $L(M') = L(M)$.

Ejemplo 50. Sea M el siguiente



sea Δ su relación de transición. Vamos a definir un M' con relación de transición Δ': los estados siguientes:

- $\Delta'(q_0, a) = \{q_1, q_3, q_4, q_5\}$
- $\Delta'(q_0, b) = (\epsilon - c)(d(\epsilon - c)(q_1, b))$; donde

  $$(\epsilon - c)(q_1) = \{q_1\}, \quad d(\{q_1\}, b) = \{q_2\}, \quad (\epsilon - c)(q_2) = \{q_2\}$$

  por lo que

  $$\Delta'(q_0, b) = \{q_2\}$$

- $\Delta'(q_1, a)$:

  $$(\epsilon - c)(q_1) = \{q_1\}$$
  $$d(q_1, a) = \{q_4\}$$
  $$(\epsilon - c)(q_4) = \{q_4, q_5\}$$

  $$\Delta'(q_1, a) = \{q_4, q_5\}$$

- $\Delta'(q_1, b)$:

  $$(\epsilon - c)(q_1) = \{q_1\}$$
  $$d(q_1, b) = \{q_2\}$$
  $$(\epsilon - c)\{q_2\} = \{q_2\}$$

  $$\Delta'(q_1, b) = \{q_2\}$$

- $\Delta'(q_2, a)$:

  $$(\epsilon - c)(q_2) = \{q_2\}$$
  $$d(q_2, a) = \varnothing$$
  $$(\epsilon - c)\varnothing = \varnothing$$

  $$\Delta'(q_2, a) = \varnothing.$$

- $\Delta'(q_2, b) = \varnothing$.

- $\Delta'(q_3, a)$:

$$(\epsilon - c)(q_3) = \{q_3, q_1\}$$
$$d(\{q_3, q_1\}, a) = d(q_3, a) \cup d(q_1, a) = \varnothing \cup \{q_4\} = \{q_4\}$$
$$(\epsilon - c)(q_4) = \{q_4, q_5\}$$
$$\Delta'(q_3, a) = \{q_4, q_5\}$$

- $\Delta'(q_3, b)$:

$$(\epsilon - c)(q_3) = \{q_3, q_1\}$$
$$d(\{q_3, q_1\}, b) = d(q_3, b) \cup d(q_1, b) = \{q_4\} \cup \{q_2\} = \{q_4, q_2\}$$
$$(\epsilon - c)\{q_4, q_2\} = (\epsilon - c)(q_4) \cup \underbrace{(\epsilon - c)(q_2)}_{\varnothing} = \{q_4, q_5\}$$
$$\Delta'(q_3, b) = \{q_4, q_5\}$$

- $\Delta'(q_4, a)$:

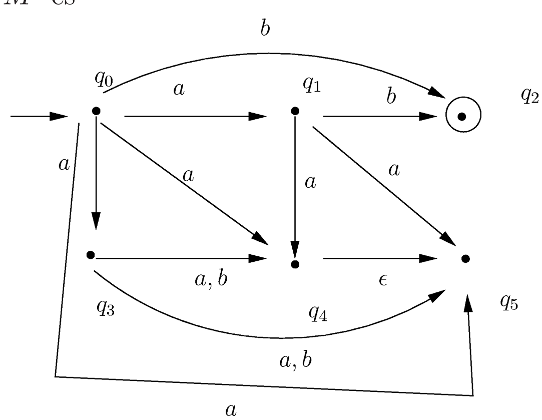
$$(\epsilon - c)(q_4) = \{q_4, q_5\}$$
$$d(q_4, a) \cup d(q_5, a) = \varnothing$$
$$\Delta'(q_4, a) = \varnothing.$$

- $\Delta'(q_4, b)$:

$$(\epsilon - c)(q_4) = \{q_4, q_5\}$$
$$d(q_4, b) \cup d(q_5, b) = \varnothing$$
$$\Delta'(q_4, b) = \varnothing$$

- $\Delta'(q_5, b) = \Delta'(q_5, a) = \varnothing$.

Obtenemos que M' es



Nótese que

$$L(M) = \{b, ab\} = L(M').$$

Teorema 7. *Sea*

$$M = (Q, \Sigma, s, F, \Delta)$$

un AFN con ϵ-transiciones. Entonces existe

$$M' = (Q', \Sigma', s', F', \Delta')$$

un AFN sin ϵ-transiciones tal que

$$L(M') = L(M).$$

Proof. Se definen

$$Q' = Q, \quad \Sigma' = \Sigma, s' = s$$

$$F' = \{q \in Q \,|\, (\epsilon - c)(q) \cap F \neq \varnothing\}$$

y si $q \in Q$ y $\sigma \in \Sigma$ entonces

$$\Delta'(q, \sigma) = (\epsilon - c)\big(d((\epsilon - c)(q), \sigma)\big)$$

Tenemos que demostrar que

$$w \in L(M') \Leftrightarrow w \in L(M).$$

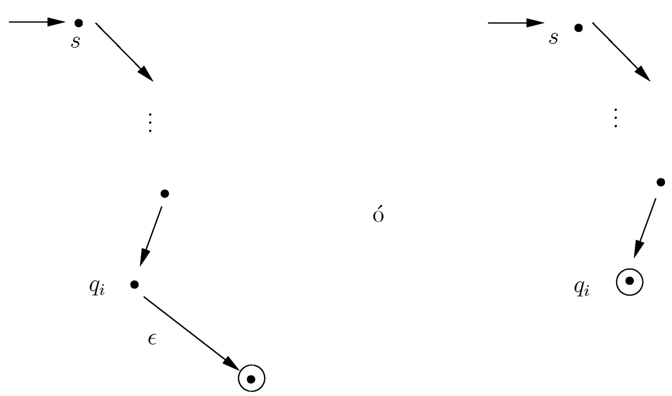
Si $w \in L(M')$ entonces $\Delta'(s', w) \cap F' \neq \varnothing$, esto es

$$\Delta'(s, w) \cap \{q \,|\, (\epsilon - c)(q) \cap F \neq \varnothing\} \neq \varnothing$$

por lo que existe $q_i \in Q$ tal que

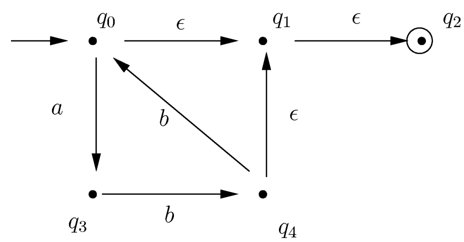
$$q_i \in \Delta'(s, w) \text{ y } (\epsilon - c)(q_i) \cap F \neq \varnothing$$

lo segundo indica que $q_i \in F$ o a q_i le sigue un estado final después de una o más ϵ-transiciones:

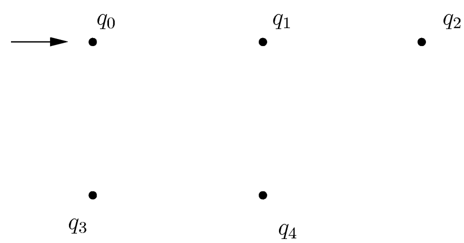


Lo que implica que $w \in L(M)$. Recíprocamente es similar. □

Ejemplo 51. Sea



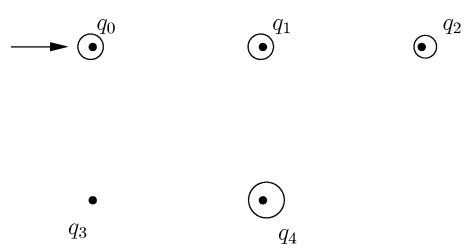
podemos encontrar M' un AFN sin ϵ-transiciones equivalente a M: por construcción, los estados de M' son los mismos que los de M:



los estados finales de M' son $F' = \{q \,|\, (\epsilon - c)(q) \cap F \neq \varnothing\}$:

q	$(\epsilon - c)(q)$	$(\epsilon - c)(q) \cap F \neq \varnothing$
q_0	$\{q_0, q_1, q_2\}$	$\{q_2\}$
q_1	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_3\}$	$\varnothing$
q_4	$\{q_4, q_1, q_2\}$	$\{q_2\}$

de donde $F' = \{q_0, q_1, q_2, q_4\}$. Actualizamos nuestro diagrama de transición:



Ahora, recordemos que

$$\Delta'(q, \sigma) = (\epsilon - c)(d((\epsilon - c)(q), \sigma))$$

- $\Delta'(q_0, a)$:

$$(\epsilon - c)(q) = \{q_0, q_1, q_2\}$$
$$d(\{q_0, q_1, q_2\}, a) = d(q_0, a) \cup d(q_1, a) \cup d(q_2, a) = \{q_3\} \cup \varnothing\varnothing = \{q_3\}$$
$$(\epsilon - c)(q_3) = \{q_3\}$$
$$\Delta'(q_0, a) = \{q_3\}$$

- $\Delta'(q_0, b)$:

$$(\epsilon - c)(q_0) = \{q_0, q_1, q_2\}$$
$$d(\{q_0, q_1, q_2\}, b) = \varnothing \cup \varnothing\varnothing = \varnothing,$$
$$\Delta'(q_0, b) = \varnothing.$$

- $\Delta'(q_1, a)$:

$$(\epsilon - c)(q_1) = \{q_1, q_2\}$$
$$d(\{q_1, q_2\}, a) = d(q_1, a) \cup d(q_2, a) = \varnothing \cup \varnothing = \varnothing,$$
$$\Delta'(q_1, a) = \varnothing.$$

- $\Delta'(q_1, b) = \varnothing$.
- $\Delta'(q_2, a) = \varnothing$.
- $\Delta'(q_2, b) = \varnothing$.
- $\Delta'(q_3, a)$:

$$(\epsilon - c)(q_3) = \{q_3\}$$
$$d(q_3, a) = \varnothing$$
$$\Delta'(q_3, a) = \varnothing.$$

- $\Delta'(q_3, b)$:

$$d(q_3, b) = \{q_4\}$$
$$(\epsilon - c)(q_4) = \{q_4, q_1, q_2\}$$
$$\Delta'(q_3, b) = \{q_4, q_1, q_2\}.$$

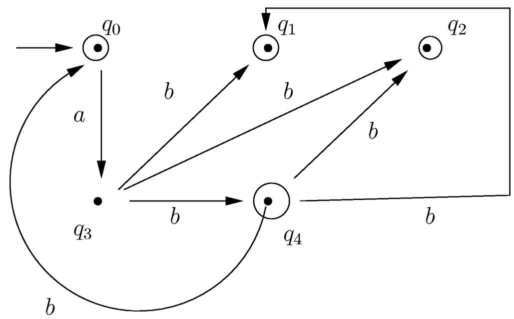
- $\Delta'(q_4, a)$:

$$(\epsilon - c)(q_4) = \{q_4, q_1, q_2\}$$
$$d(\{q_4, q_1, q_2\}, a) = d(q_4, a) \cup d(q_1, a) \cup d(q_2, a) = \varnothing \cup \varnothing\varnothing$$
$$\Delta'(q_4, a) = \varnothing.$$

- $\Delta'(q_4, b)$:

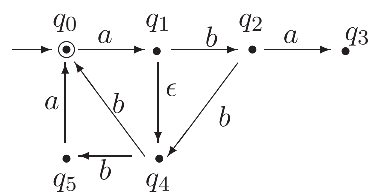
$$d(\{q_4, q_1, q_2\}, b) = d(q_4, b) \cup d(q_1, b) \cup d(q_2, b) = \{q_0\}$$
$$(\epsilon - c)(q_0) = \{q_0, q_1, q_2\}$$
$$\Delta'(q_4, b) = \{q_0, q_1, q_2\}.$$

Hemos obtenido M':

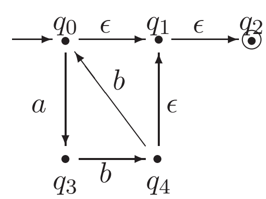


Tarea 9.

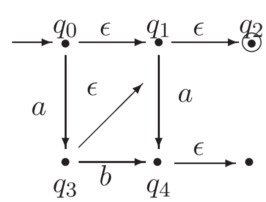
(1) *Calcular* $\Delta(q_0, abb)$ *y* $\Delta(q_0, aba^2b)$ *para el AFN siguiente*



(2) *Obtener* $(\epsilon - c)(\{q_1, q_4\})$ *y* $(\epsilon - c)(d(q_3, b))$ *para el AFN siguiente*



(3) *Usar la técnica estudiada para calcular* $\Delta(q_3, b)$ *en*

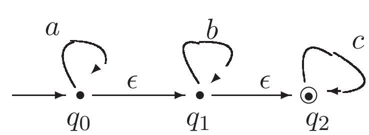


(4) *Para el AFN dado en la figura siguiente*

(a) *obtener la tabla de transición para* Δ

(b) *obtener la* ϵ*-cerradura de* q_i *para* $i = 0, 1, 2$

(c) *calcular* $\Delta(q_0, a)$, $\Delta(q_0, b)$ *y* $\Delta(q_0, c)$.



(5) *Para el AFN del ejercicio inmediato anterior, obtener el AFN que se obtiene al eliminar las ϵ-transiciones. Dar la tabla para Δ'.*

10. Autómatas finitos y expresiones regulares

Se demostrará que (teorema de Kleene):

(1) Si M es un autómata, entonces $L(M)$ es regular.

(2) Si L es regular, entonces existe un AF M tal que $L(M) = L$.

Es decir, que los lenguajes aceptados por loa autómatas finitos son exactamente los lenguajes regulares.

Ejemplo 52. Sea $\Sigma = \{a, b\}$.

(1) Construir M_1 un AFN tal que $L(M) = \{a\}$.

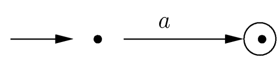
(2) Construir M_2 un AFN tal que acepte sólo al lenguaje vacío.

(3) Construir un M_3 un AFB tal que $L(M_3) = \{\epsilon\}$.

(4) Construir M_4 un AFD tal que $\{a_4\}$.

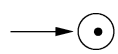
Sol.

(1)

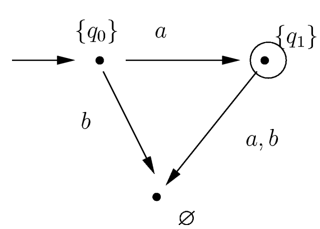


(2)

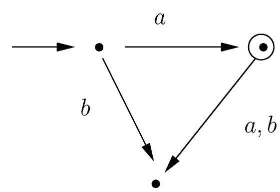
(3)



(4)

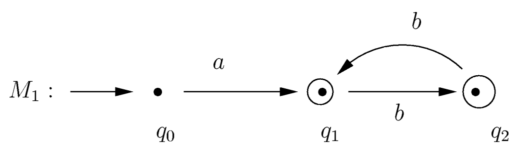


es decir

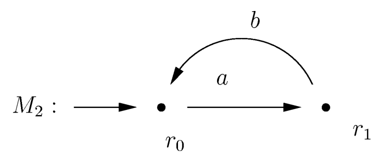


En el siguiente ejemplo se ilustra un procedimiento para construir un autómata que acepte la unión de lenguajes.

Ejemplo 53. Sean

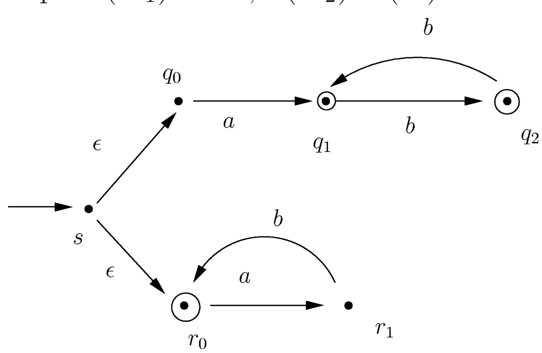


y



Construir M un AFN tal que $L(M) = L(M_1) \cup L(M_2)$.

Sol. Tenemos que $L(M_1) = ab^*$, $L(M_2) = (ab)^*$. El M pedido es



donde claramente $L(M) = L(M_1) \cup L(M_2) = ab^* \cup (ab)^*$.

Teorema 8. *Sean* $M_1 = (Q_1, \Sigma_1, s_1, F_1, \Delta_1)$, $M_2 = (Q_2, \Sigma_2, s_2, F_2, \Delta_2)$ *dos AFN. Entonces existe* M *un AFN tal que*

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

Proof. Se construirá M como un AFN con ϵ-transiciones. Sea

$$M = (Q, \Sigma, s, F, \Delta)$$

donde

$$\begin{aligned} Q &= Q_1 \dot\cup Q_2 \cup \{s\}, \text{ con } s \notin Q_1 \cup Q_2 \\ \Sigma &= \Sigma_\cup \Sigma_1 \\ s& \\ F_0 &= F_1 \dot\cup F_2 \\ \Delta &= \Delta_1 \cup \Delta_2 \cup \{(s, \epsilon, s_1), (s_0, \epsilon, s_2)\} \end{aligned}$$

es decir, Δ se define como: si $\sigma \in \Sigma$,

$$\Delta(q, \sigma) = \begin{cases} \Delta_1(q, \sigma), & \text{si } q \in Q_1 \\ \Delta_2(q, \sigma), & \text{si } q \in Q_2 \end{cases}$$

$$\Delta(s, \sigma) = \varnothing, \quad \Delta(s, \epsilon) = \{s_1, s_2\}.$$

Tenemos que probar que

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

Sea $w \in \Sigma^*$ tal que $w \in L(M)$, entonces $\Delta(s, w) \cap F \neq \varnothing$ lo que implica que

$$\Delta(s, w) \cap F_1 \neq \varnothing \text{ ó } \Delta(s, w) \cap F_2 \neq \varnothing.$$

Si $\Delta(s, w) \cap F_1 \neq \varnothing$ entonces

$$\begin{aligned} \varnothing \neq \Delta(s, w) &\cap F_1 \\ = \Delta(s, \epsilon w) &\cap F_1 \\ = \Delta(\Delta(s, \epsilon), w) &\cap F_1 \\ = \Delta(s_1, w) &\cap F_1 \\ = \Delta_1(s_1, w) &\cap F_1 \end{aligned}$$

lo que implica que $w \in L(M_1)$.

Similarmente, si $\Delta(s, w) \cap F_2 \neq \varnothing$ entonces $w \in L(M_2)$. En cualquier caso:

$$w \in L(M_1) \cup L(M_2).$$

Recíprocamente, $L(M_1) \subseteq L(M)$ pues si $w \in L(M_1)$ entonces

$$\begin{aligned}\varnothing \neq \Delta_1(s_1, w) \cap F_1 \\ &= \Delta(s_1, w) \cap F_1 \\ &= \Delta(\Delta(s, \epsilon), w) \cap F_1 \\ &= \Delta(s, \epsilon w) \cap F_1 \\ &= \Delta(s, w) \cap F_1 \\ &\subseteq \Delta(s, w) \cap F\end{aligned}$$

así, $\Delta(s, w) \cap F \neq \varnothing$, lo que implica $w \in L(M)$.

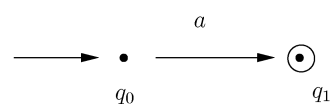
Similarmente $L(M_2) \subseteq L(M)$; y por tanto

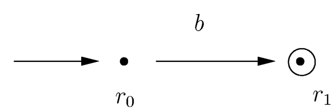
$$L(M_1) \cup L(M_2) \subseteq L(M).$$

□

Una operación que aparece para la construcción de lenguajes regulares es la unión. Para la cual existe un algoritmo correspondiente a autómatas. La siguiente operación que aperece con los lenguajes regulares es la concatenación. También existe un algoritmo correspondiente en autómatas.

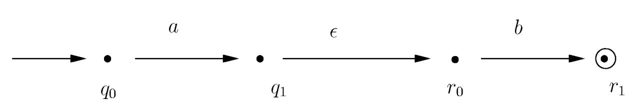
Ejemplo 54. Sean

M_1 :


M_2 :


tenemos que $L(M_1) = \{a\}$ y $L(M_2) = \{b\}$. Encontrar M un AFN tal que $L(M) = L(M_1)L(M_2)$.

Sol.

M :


$$L(M) = \{ab\} = \{a\}\{b\}.$$

Teorema 9. *Si $M_i = (Q_i, \Sigma_i, s_i, F_i, \Delta_i)$, $i = 1, 2$ son dos AFN, entonces existe un M AFN tal que*

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

Proof. Se define $M = (Q, \Sigma, s, F, \Delta)$ donde

$$\begin{aligned} Q &= Q_1 \dot\cup Q_2 \\ \Sigma &= \Sigma_1 \cup \Sigma_2 \\ s &= s_1 \\ F &= F_2 \\ \Delta &= \Delta_1 \cup \Delta_2 \cup (F_1 \times \{\epsilon\} \times \{s_1\}) \end{aligned}$$

es decir, si $\sigma \in \Sigma_1 \cup \Sigma_2$, $q \in Q_1 \cup Q_2$,

$$\Delta(q, \sigma) = \begin{cases} \Delta_1(q, \sigma) & \text{si } q \in Q_1 \text{ y } \sigma \in \Sigma_1 \\ \Delta_2(q, \sigma) & \text{si } q \in Q_2 \text{ y } \sigma \in \Sigma_2 \\ \varnothing & \text{otro caso.} \end{cases}$$

$$\Delta(q, \epsilon) = \begin{cases} \{s_2\} & \text{si } q \in F_1 \\ \varnothing & \text{si } q \notin F_2. \end{cases}$$

Por demostrar que

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

es decir, que para $w \in \Sigma^*$,

$$w \in L(M) \Leftrightarrow w \in L(M_1)L(M_2).$$

($\Leftarrow$) Si $w \in L(M_1)L(M_2)$ entonces $w = xy$ con $x \in L(M_1)$ y $y \in L(M_2)$; en particular $x \in \Sigma_1^*$ y $y \in \Sigma_2^*$. Podemos escribir

$$w = x\epsilon y$$

luego

$$\Delta(s, w) = \Delta(\Delta(\Delta(s, x), \epsilon), y). \tag{4}$$

Como $x \in \Sigma_1^*$, entonces

$$\begin{aligned} \Delta(s, x) &= \Delta(s_1, x) \\ &= \Delta_1(s_1, x) \end{aligned}$$

y como $x \in L(M_1)$ entonces $\Delta(s_1, x) \cap F_1 \neq \varnothing$, por lo que

$$\Delta_1(s_1, x) = \{\ldots, \underbrace{q}_{\in F_1}, \ldots\}$$

y

$$\begin{aligned} \Delta(\Delta(s, x), \epsilon) &= \Delta(\{\ldots, q, \ldots\}, \epsilon) \\ &= \Delta(\underbrace{q}_{\in F_1}, \epsilon) \\ &= \{s_2\} \end{aligned}$$

Usando la ecuación (4),

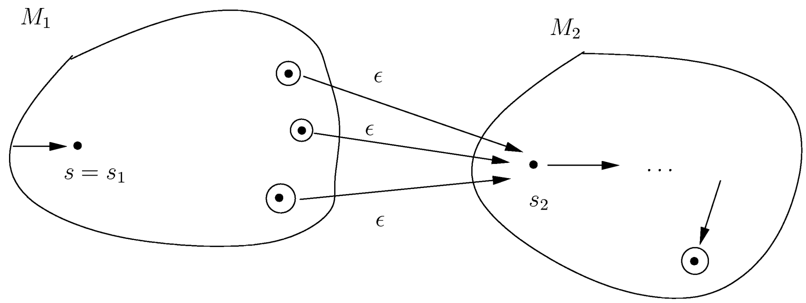


Figure 2. M

$$\Delta(s,w) = \Delta(s_2,y)$$
$$= \Delta(s_2,y)$$

pues $y \in \Sigma_2^*$. Pero $\Delta_2(s_2,y) \cap F_2 \neq \varnothing$, luego

$$\Delta(s,w) \cap F = \Delta(s,w) \cap F_2 \neq \varnothing$$

lo que implica que

$$w \in L(M).$$

($\Rightarrow$) Supongamos que $w \in L(M)$ entonces

$$\underbrace{\Delta(s,w)}_{\Delta(s_1,w)} \cap F \neq \varnothing$$

es decir, las transiciones indicadas por w deben de pasar del estado s_1 en M_1 a un estado de aceptación en M_2: la única forma de pasar de M_1 a M_2 es usando las ϵ-transiciones que ligan a los estados finales de M_1 con el inicial de M_1 (ver figura 2). Por lo que w debe primero de transitar hacia los estados de F_1 y luego hacia F_2. Esto es

$$w = xy$$

con $\Delta_1(s_1,x) \cap F_1 \neq \varnothing$ y $\Delta_2(s_2,y) \cap F_2 \neq \varnothing$. Es decir $x \in L(M_1)$ y $y \in L(M_2)$. Por lo tanto

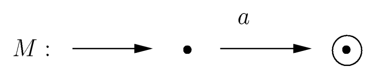
$$w = xy \in L(M_1)L(M_2)$$

$\square$

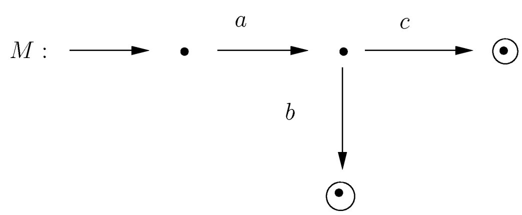
La siguiente operación que se usa para la construcción de los lenguajes regulares es la cerradure de Kleene. También existe una construcción similar para autómatas.

Ejemplo 55. En cada inciso considere $L(M)$ y construya M_1 un AFN tal que $L(M_1) = L(M)^*$.

(1)

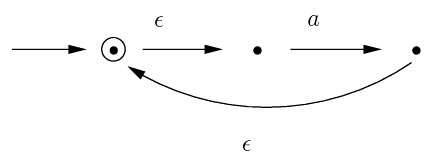


(2)

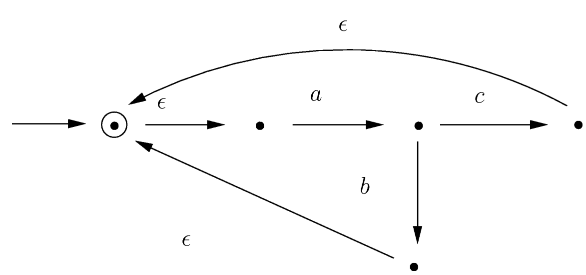


Sol.

(1) Tenemos que $L(M) = \{a\}$. Por lo que tenemos que construir M_1 tal que $L(M_1) = a^*$. Tal M_1 es:



(2) Tenemos que $L(M) = \{ab, ac\}$. Queremos M_1 un AFN tal que $L(M_1) = (ab \cup ac)^*$. Tal es



El ejemplo anterior ilustra el algoritmo subyacente en la demostración del siguiente teorema.

Teorema 10. *Si* $M = (Q, \Sigma, s, F, \Delta)$ *es un AFN, entonces existe* $M_1 = (Q_1, \Sigma_1, s_1, F_1, \Delta_1)$ *tal que* $L(M_1) = L(M)^*$.

Proof. Se le añade un nuevo estado a M:

$Q_1 = Q \cup \{s_1\}$ con $s_1 \notin Q_1$
$\Sigma_1 = \Sigma$
s_1
$F_1 = \{s_1\}$
Δ se le añaden ϵ-transiciones de los estados finales a s_0:

Si $\sigma \in \Sigma_1$, $q \in Q_1$:

$$\Delta_1(q,\sigma) = \begin{cases} \Delta(q,\sigma) & \text{si } q \in Q \\ \varnothing & \text{otro caso.} \end{cases}$$

$$\Delta(q,\epsilon) = \begin{cases} \{s_1\} & \text{si } q \in F \\ \{s\} & \text{si } q = s_1 \\ \varnothing & \text{otro caso.} \end{cases}$$

Tenemos que demostrar que

$$w \in L(M_1) \Leftrightarrow w \in L(M)^*.$$

□

Hemos demostrado:

Teorema 11. *Los lenguajes aceptados por los automatas finitos contienen a*

(1) $\varnothing$, $\{\epsilon\}$, *los lenguajes unitarios* $\{a\}$, $\forall a \in \Sigma$.
(2) *Además tales elnguajes son cerrados con respecto a la unión, concatenación y cerradura de Kleene.*

Corolario 1. *Si r es una expresión regular entonces existe M un autómata finito tal que $r = L(M)$.*

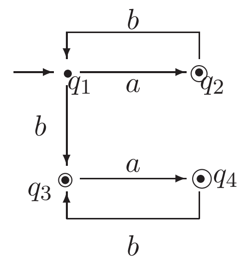
Proof. Las expresiones regulares se contruyen a partir de $\varnothing$, $\{\epsilon\}$ y los lenguajes unitarios $\{\sigma\}$, $\sigma \in \Sigma$ con cerrauras uniones y concatenaciones; y para tales construcciones existen autómatas que las aceptan. □

Tarea 10.

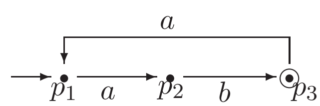
(1) *Obtener un AFN que acepte ϵ.*
(2) *Obtener un AFN que acepte $\{a\}$. Obtener otro AFN que acepte $\{b\}$. Usar las técnicas vistas para unir éstos AFN en uno que acepte el lenguaje $\{a,b\}$.*
(3) *Obtener un AFN que acepte $(a \cup b)^* \cup (aba)^+$.*

(4) *Obtener un AFN que acepte todas las cadenas de la forma bow, bowwowwow, bowwowwowwowwow, ... Conseguir un AFN que acepte todas las cadenas de la forma ohmy, ohmyohmy, ohmyohmyohmy, ... Unir los dos AFN para que se acepte la uninión de los dos lenguajes. Téngase en cuenta que los símbolos de un alfabeto no tiene por qué ser caracteres de longitud uno.*

(5) *Sea M_1 dado por*



y M_2 dado por



Obtener un AFN que acepte $L(M_1)L(M_2)$. Obtener un AFN que acepte $L(M_2)L(M_1)$.

(6) *Sean $M_1 = \big(\{q_1, q_2, q_3\}, a, b, q_1, \{q_1\}, \Delta_1\big)$ y $M_2 = \big(\{p_1, p_2, p_3, p_4\}, \{0, 1\}, p_1, \{p_1, p_2\}, \Delta_2\big)$, donde Δ_1 y Δ_2 viene dados por las tablas siguientes:*

Δ_1	a	b
q_1	$\{q_2, q_3\}$	$\varnothing$
q_2	$\varnothing$	$\{q_1\}$
q_3	$\{q_3\}$	$\{q_3\}$

Δ_2	0	1
p_1	$\{p_2\}$	$\{p_3, p_4\}$
p_2	$\varnothing$	$\{p_3, p_4\}$
p_3	$\{p_2\}$	$\varnothing$
p_4	$\{p_3\}$	$\varnothing$

Obtener un AFN que acepte $L(M_1)L(M_2)$. Obtener un AFN que acepte $L(M_2)L(M_1)L(M_1)$. Obtener finalmente, un AFN que acepte $L(M_1)^2 \cup L(M_1)$.

(7) *Obtener una AFN para $(ab)^*$ a partir de los AFN que aceptan $\{a\}$ y $\{b\}$.*

(8) *Obtener un AFN para $(aa \cup b)^*(bb \cup a)^*$ a partir de los AFN que aceptan $\{a\}$ y $\{b\}$.*

(9) *Obtener un AFN para*

$$((a \cup b)(a \cup b))^* \cup ((a \cup b)(a \cup b)(a \cup b))^*$$

a partir de los AFN para $\{a\}$ y $\{b\}$.

(10) *Si $M = (Q, \Sigma, s, F, \delta)$ es un autómata finito determinista, entonces el complemento de $L(M)$ [es decir $\Sigma^* - L(M)$] es aceptado por el autómata $M' = (Q, \Sigma, s, Q - F, \delta)$. ¿Es M' un AFD o un AFN?*

Obtener un AFD que acepte ab^*ab. Obtener un autómata finito que acepte $\{a,b\}^* - ab^*ab$.

11. Lema de Arden

Definición 35. *Sea $M = (Q, \Sigma, s, F, \Delta)$ un AFN con estado inicial $s = q_0$. Sea $q_i \in Q$. Se define*

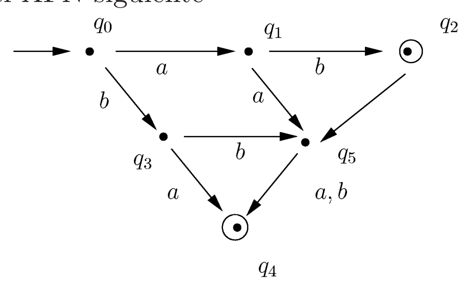
$$A_i = \{w \in \Sigma^* \mid \Delta(q_i, w) \cap F \neq \varnothing\}$$

i.e., A_i es el conjunto de cadenas que desde q_i llegan a un estado de aceptación. El conjunto A_i se llama **cadenas aceptadas por el estado** *q_i.*

Notemos que

- $A_0 = L(M)$
- Si $q_i \in F$, entonces $\varnothing \in A_i$.

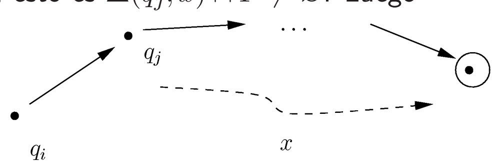
Ejemplo 56. En el AFN siguiente



tenemos que

$$A_5 = \varnothing, \quad A_4 = \{\epsilon\}, \quad A_3 = \{a\}$$
$$A_2 = \{\varnothing\}, \quad A_1 = \{b\}, \quad A_0 = \{ba, ab\}.$$

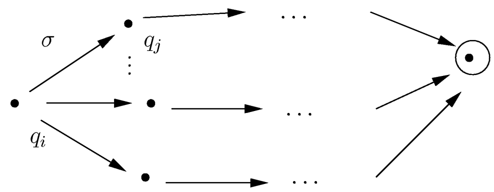
Si $q_j \in \Delta(q_i, \sigma)$ entonces $\sigma A_j \subset A_i$, pues, si $w \in \sigma A_j$ entonces $w = \sigma x$ para algún $x \in A_j$, esto es $\Delta(q_j, x) \cap F \neq \varnothing$. Luego



$$\begin{aligned}\Delta(q_i, w) \cap F &= \Delta(\Delta(q_i, \sigma), x) \cap F \\ &\supseteq \Delta(q_j, x) \cap F \\ &\neq \varnothing\end{aligned}$$

Por lo que

$$A_i = \bigcup_\sigma \{\sigma A_j \mid q_j \in \Delta(q_i, \sigma)\} \tag{5}$$



Ejemplo 57. En el AFN del ejemplo anterior 47, usando la ecuación (5)

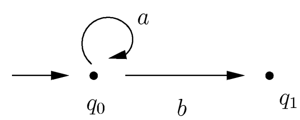
$$A_0 = aA_1 \cup bA_3, \quad A_1 = bA_2 \cup aA_5, \quad A_3 = aA_4 \cup bA_5$$
$$A_2 = aA_5 \cup bA_5 \cup \epsilon, \quad A_4 = \epsilon \cup aA_5 \cup bA_5$$
$$A_5 = \varnothing$$

este es un sistema de cuaciones que se puede resolver por sustitución regresiva:

$$A_4 = \epsilon, \quad A_3 = a, \quad A_2 = \epsilon, \quad A_1 = b$$
$$L(M) = A_0 = ab \cup ba$$

es decir, hemos calculado $L(M) = ab \cup ba$.

A veces, resultan ecuaciones autorecursivas. Por ejemplo, en



resulta, usando de nuevo la ecuación (5),

$$A_0 = aA_0 \cup bA_1, \quad A_1 = \epsilon$$
$$A_0 = aA_0 \cup b$$

para resolver tales ecuaciones autorecursivas se usa el lema de Arden.

Lema 3 (de Arden)**.** *Una ecuación de la forma*

$$X = AX \cup B \tag{6}$$

donde $\epsilon \notin A$ *tiene como solución única a*

$$X = A^*B$$

Proof.

(1) Primero notemos que $X = A^*B$ es solución de (6), pues

$$\begin{aligned} A^*B &= (A^+ \cup \epsilon)B \\ &= A^+B \cup B \\ &= AA^*B \cup B \\ &= A(A^*B) \cup B \end{aligned}$$

(2) Ahora supongamos que hay otra solución a (6) y llamémosla Y. Esto es

$$Y = AY \cup B$$

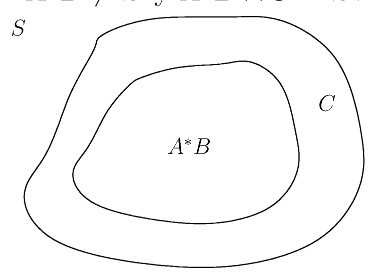
con $X = A^*B \neq Y$. Entonces

$$\begin{aligned} X \cup Y &= AX \cup B \cup AY \cup B \\ &= A(X \cup Y) \cup B \end{aligned}$$

esto es $S = X \cup Y$ es también solución y es más grande que X, pues $X \subset S$. Podemos escribir

$$S = A^*B \cup C$$

donde $C = S - A^*B \neq \varnothing$ y $A^*B \cap C = \varnothing$.



Luego, como S es solución.

$$S = AS \cup B$$

y

$$\begin{aligned} A^* \cup C &= AA^*B \cup AC \cup B \\ &= A^+B \cup AC \cup B \\ &= (A^+ \cup \epsilon)B \cup AC \\ &= A^*B \cup AC \end{aligned}$$

por lo que

$$\begin{aligned} (A^*B \cup C) \cap C &= (A^*B \cup AC) \cap C \\ &= (A^*B \cap C) \cup (AC \cap C) \\ &= \varnothing \cup AC \cup C \end{aligned}$$

y entonces

$$C = AC \cap C \subseteq AC$$

es decir

$$C \subseteq AC$$

Sea $w \in C$ tal que $|w|$ es mínima; entonces $w = aw_0$ con $a \in A$ y $|a| > 0$ pues $\epsilon \notin A$. Luego

$$|w| = |a| + |w_0| > |w_0|$$

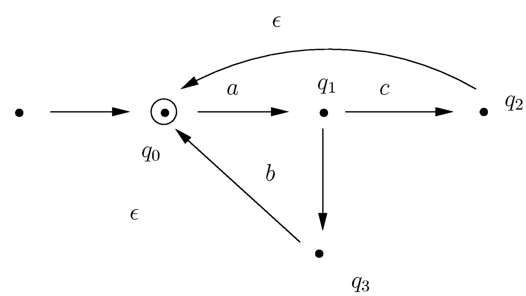
$$|w| > |w_0|$$

lo cual es imposible, pues $|w|$ es mínima.

Por lo tanto $X = A^*B$ es la solución única.

□

Ejemplo 58. Consideremos M como



entonces

$$A_0 = aA_0 \cup bA_1$$
$$A_1 = \epsilon$$
$$A_0 = aA_0 \cup b$$

y como $\epsilon \notin \{a\}$, entonces $A_0 = a^*b$. Es decir

$$L(M) = a^*b.$$

Ejemplo 59. En M el AFN siguiente, calcule $L(M)$.

Proof.

$$A_0 = aA_1 \cup \epsilon$$
$$A_1 = cA_2 \cup bA_3$$
$$A_2 = \epsilon A_0$$
$$A_3 = \epsilon A_0$$

$$\begin{aligned} A_1 &= c\epsilon A_0 \cup b\epsilon A_0 \\ &= cA_0 \cup bA_0 \\ &= (c \cup b)A_0 \end{aligned}$$

y

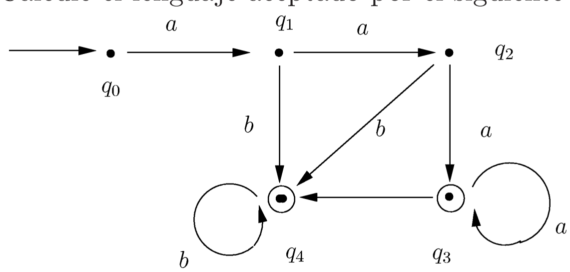
$$A_0 = a(c \cup b)A_0 \cup \epsilon$$

y como $\epsilon \notin L(a(c \cup b))$ entonces podemos usar el lema de Arden y obtener

$$A_0 = (a(c \cup b))^*\epsilon = (ac \cup bc)^*$$

□

Ejemplo 60. Calcule el lenguaje aceptado por el siguiente autómata.



Sol. Tenemos que

$$\begin{aligned} A_0 &= aA_1 \\ A_1 &= aA_2 \cup bA_4 \\ A_2 &= aA_3 \cup bA_4 \\ A_3 &= \epsilon \cup aA_3 \cup bA_4 \\ A_4 &= \epsilon \cup bA_4 = bA_4 \cup \epsilon \end{aligned}$$

entonces, por el lema de Areden, como $\epsilon \notin \{b\}$,

$$A_4 = b^*\epsilon = b^*$$

$$\begin{aligned} A_3 &= aA_3 \cup (\epsilon \cup bb^*) \\ &= aA_3 \cup b^* \end{aligned}$$

lo que implica,

$$A_3 = a^*b^*$$

y

$$\begin{aligned} A_2 &= = aa^*b^* \cup bb^* \\ &= a^+b^* \cup b^+ \end{aligned}$$

$$A_1 = aa^+b^* \cup ab^* \cup bb^*$$

por lo tanto,

$$A_0 = a^2a^+ \cup a^2b^* \cup ab^+$$

□

El lema de Arden garantiza que

Lema 4. *Sea M un AF. Existe r una expresión regular tal que*

$$L(M) = L(r).$$

y en consecuencia,

Teorema 12 (Kleen)**.** *Un lenguaje L es regular si y sólo si L es aceptado por algún autómata.*

Tarea 11.

(1) *Obtener un AFN para $(aa \cup b)^*(bb \cup a)^*$ a partir de los AFN que aceptan $\{a\}$ y $\{b\}$.*

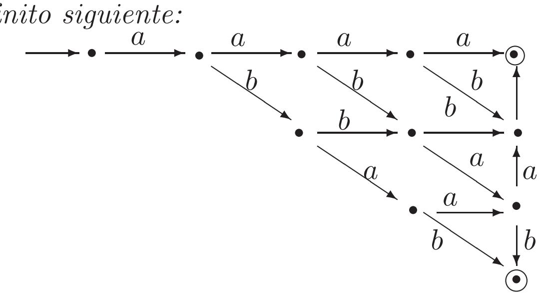
(2) *Obtener un AFN para*

$$((a \cup b)(a \cup b))^* \cup ((a \cup b)(a \cup b)(a \cup b))^*$$

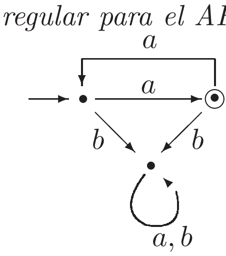
a partir de los AFN para $\{a\}$ y $\{b\}$.

(3) *Si $M = (Q, \Sigma, s, F, \delta)$ es un autómata finito determinista, entonces el complemento de $L(M)$ [es decir $\Sigma^* - L(M)$] es aceptado por el autómata $M' = (Q, \Sigma, s, Q - F, \delta)$. ¿Es M' un AFD o un AFN? Obtener un AFD que acepte ab^*ab. Obtener un autómata finito que acepte $\{a, b\}^* - ab^*ab$.*

(4) *Obtener una expresión regular para el lenguaje aceptado por el autómata finito siguiente:*

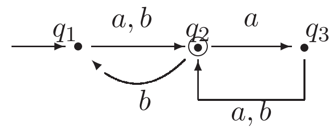


(5) *Obtener una expresión regular para el AFD siguiente:*

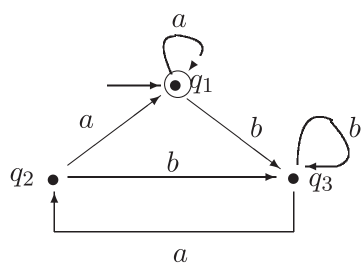


(6) *Obtener una expresión regular para los lenguajes aceptados por cada uno de los autómatas siguientes:*

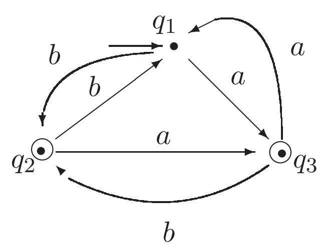
(a)



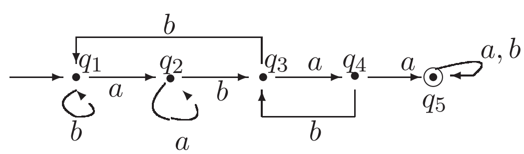
(b)



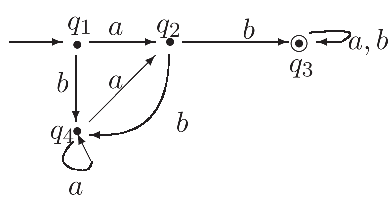
(c)



(d)



(e)



12. Propiedades de los lenguajes regulares

12.1. Lema del bombeo. El lema del bombeo permite mostrar que no todos los lenguajes son regulares.

Lema 5 (del bombeo)**.** *Sea L un lenguaje regular. Entonces existe n constante tal que $\forall w \in L$ con $|w| \geq n$ se obtiene que*

$$w = uvx \text{ con } |v| \geq 1 \text{ y } |uv| \leq n$$

y además

$$\forall i, \quad uv^i x \in L.$$

Proof. Si L es finito entonces existe m la mayor de las longitudes de las palabras de L. Se pone $n = m + 1$ y el lema se cumple por vacuidad (no hay palabras, en L, de longitud $\geq n$).

Si L es infinito, entonces, por el teorema de Kleene, existe M un AFD tal que $L = L(M)$. Sea

$$M = (Q, \Sigma, s, F, \delta).$$

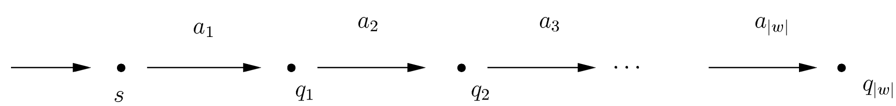
Definimos $n = |Q|$ el número de estados de M. Si $w \in L$ con $|w| \geq n$ entonces

$$w = a_1 \cdots a_{|w|-1} a_{|w|}$$

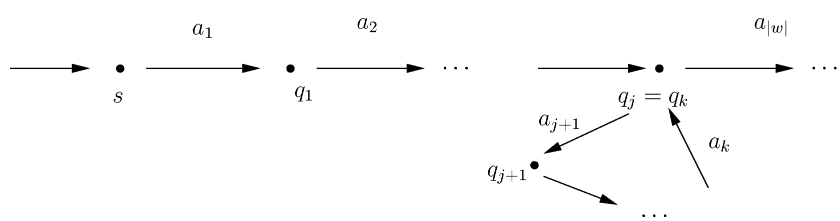
con cada $a_i \in \Sigma$. Definimos además $q_1, q_2, \ldots, q_{|w|}$ como los estados que resultan de las transiciones indicadas por w, esto es:

$$q_1 = \delta(s, a_1), q_2 = \delta(q_1, a_2), \ldots, q_{|w|} = \delta(q_{|w|-1}, a_{|w|}).$$

Puesto que $w \in L(M)$ entonces $q_{|w|} \in F$:



Como sólo hay n estados, los estados $s, q_1, \ldots, q_n, \ldots, q_{|w|}$ no pueden ser todos diferentes, pues en tal caso tendríamos $|w| + 1 > n$: más estados que n!. Por lo que algunos de éstos se repiten: existen j, k tales que $1 \leq j < k \leq n \leq |w|$ tales que $q_j = q_k$. Lo que obliga la aparición de un ciclo:



Definimos

- $u = a_1 \cdots a_j$
- $v = a_{j+1} \cdots a_k$
- $x = a_{k+1} \cdots a_|w|$

Entonces

$$w = uvx$$

Como v corresponde al ciclo, y este existe entonces $|v| > 1$. Además

$$|uv| = k \leq n.$$

Finalmente notemos que u lleva el estado inicial sl estado que se repite q_j, luego x continua con este estado hasta lllevarlo al estdoo de aceptación $q_{|w|}$;

$$\begin{aligned} \delta(s, ux) &= \delta(\delta(s, u), x) \\ &= \delta(q_j, u) \\ &= \delta(q_k, x) \\ &= q_{|w|} \in F \end{aligned}$$

por lo que $ux \in L$. Similarmente, como v^2 lleva el estado q_j al estado

$$q_k$$

entonces uv^2x también se acepta. En general, si $i \geq 0$ entonce

$$\begin{aligned} \delta(s, uv^i x) &= \delta(\delta(\delta(s, u), v^i), x) \\ &= \delta(\delta(q_j, v^i), x) \\ &= \delta(q_k, x)) \\ &= q_{|w|} \in F, \end{aligned}$$

por lo que $uv^ix \in F$. □

Ejemplo 61. Sea $\Sigma = \{0, 1\}$ y L el lenguaje de todas las cadenas con el mismo númoer de 0's que de 1's. Demostrar que L no es regular.

Proof. Por contradicción: si L fuera regular entonces existe n tal que se cumple el lema del bombeo. Esto es las palabras de longitud más grande o igual a n se descomponen en tres partes con ciertas características. En particular $q = 0^n 1^n \in L$ es una palabra de longitud $|w| = 2n > n$, luego tal se puede descomponer en tres partes u, v, x tales que

$$w = 0^n 1^n = uvx$$

con $|uv| \leq n$, $|v| > 1$ y $uv^ix \in L$, $\forall i \geq 0$. Como uv es la parte al principio de 0^n1^n y $|uv| \leq n$ entonces uv está formada sólo por ceros, por lo que x, que ess la parte final de w, debe de tener todos los unos:

$$w = \underbrace{0 \cdots 0}_{uv}\underbrace{0 \cdots 01 \cdots 1}_{x}$$

Pero para $i = 0$, $ux \in L$, por lo que ux tiene n unos, de donde debe de tener n ceros que son los de u. Se deduce entonces que v no contribuye con ningún cero a $w = uvx$. Esto es $v = \epsilon$, lo cual contradice el elma del bombeo. Por lo tanto L no es regular. □

Ejemplo 62. Sea
$$L = \{a^{i^2} \mid i \geq 1\}$$
entonces L no es un lenguaje regular.

Proof. Supongamos que L es regular. Existe una constante n tal que cualquier palabra $w \in L$ con $|w| \geq n$ se descompone según las características del lema del bombeo.

Por definición de L tenemos que $a^{n^2} \in L$. Luego a^{n^2} se puede decomponer como
$$a^{n^2} = uvx$$
con $|uv| \leq n$ y $uv^ix \in L$, $\forall i \geq 0$.. En particular $uv^2x \in L$. Por lo que, para algún k entero $|uv^2x| = k^2$. Luego,
$$\begin{aligned} n^2 &= |a^{n^2}| \\ &= |uvx| \\ &< |uv^2x| \\ &= |u| + |v| + |v| + |x| \\ &= |uvx| + |v| \\ &\leq n^2 + n \end{aligned}$$
pues $n \geq |uv| \geq |v|$, Se sigue que
$$\begin{aligned} n^2 < |uv^2x| &\leq n^2 + n \\ &< n^2 + 2n + 1 \\ &= (n+1)^2 \end{aligned}$$
entonces
$$n^2 < \underbrace{|uv^2x|}_{k^2} < (n+1)^2$$
estp es, $n^2 < k^2 < (n+1)^2$, i.e., $n < k < n+1$ con k entero: un absurdo. □

13. Otra versión del lema del bombeo

Teorema 13. *Sea* $L = \{a^nb^n \mid n \geq 0\} = \{\epsilon, ab, aabb, aaabbb, \ldots\}$. *Entonces* B *no es un lenguaje regular.*

Dem. Por contradicción. Supongamos que B es regular. El teorema de Kleene asegura que existe M un AFD tal que $L = L(M)$. Pongamos
$$M = (Q, \Sigma, s, F, \delta).$$
y sea k el número de elementos de Q. Consideremos $N >> k$ (N *mucho mayor* que k) y la palabra $w = a^Nb^N$. Evidentemente $w \in L$ y w es aceptada

por M. Consideremos los primeros estados que se emplean en la aceptación de w:

$$\begin{aligned}
\delta(s,a) &= r_1 \in Q,\\
\delta(s,a^2) &= r_2 \in Q,\\
&\vdots\\
\delta(s,a^N) &= r_N\\
\delta(s,a^Nb) &= r_{N+1}\\
&\vdots\\
\delta(s,a^Nb^N) &= r_{2N} \in F.
\end{aligned}$$

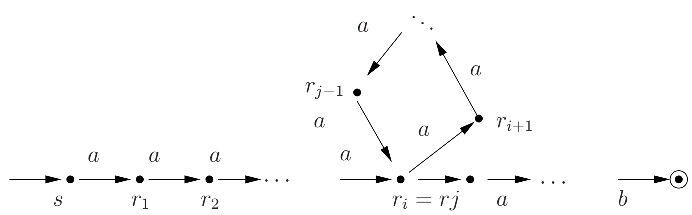
Tenemos que

$$\{r_1, r_2, \ldots, r_N\} \subseteq Q$$

luego como $N >> k$ y el principio de las casillas, se sigue que los estados r_i no pueden ser diferentes entre sí: se deben de repetir. Esto es: existen i, j tales que $1 \leq i < j \leq N$ y

$$r_i = r_j. \tag{7}$$

Esto indica que en el camino de las transiciones indicadas por $w = a^Nb^N$, desde estado inicial, se forma un bucle:



por lo que el camino que usa el bucle es prescindible, i.e., $a^{N-(j-i)}b^N \in L$. En efecto:

$$\begin{aligned}
\delta(s,a^{N-(j-i)}) &= \delta(s,a^{N-j+i})\\
&= \delta(s,a^i a^{N-j})\\
&= \delta(\delta(s,a^i),a^{N-j})\\
&= \delta(r_i,a^{N-j})\\
&= \delta(r_j,a^{N-j})\\
&= \delta(\delta(s,a^j),a^{N-j})\\
&= \delta(s,a^j a^{N-j})\\
&= \delta(s,a^N),
\end{aligned}$$

luego

$$\delta(s, a^{N-(j-i)}b^N) = \delta(\delta(s, a^{N-(j-i)}), b^N) = \delta(s, a^N b^N) \in F$$

por lo que $a^{N-(j-i)}b^N \in L(M) = L$: absurdo. □

De hecho la prueba del teorema anterior se puede usar como un hecho general: *el lema del bombeo.* Veamos otra vez tal prueba en otro ejemplo.

Ejemplo 63. *Sea* $C = \{w \in \{a\}^* \,|\, |w|$ *es una potencia de* $2\}$. *El lenguaje* C *también se puede escribir como*

$$\begin{aligned} C &= \{a^{2^n} \,|\, n \geq 0\} \\ &= \{a, a^2, a^4, a^8, a^{16}, \ldots\} \end{aligned}$$

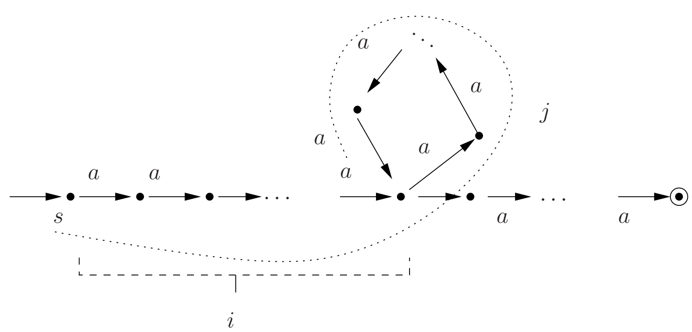
Probar que C *no es un lenguaje regular.*

Dem. Por contradicción. Supongamos que C es regular. Entonces existe M un DFA tal que $C = L(M)$. Sea Q el conjunto de estados de C y k el número de estados en Q. Consideremos $N >> k$ y la palabra $w = a^{2^N} \in C$. Los estados usados en la aceptación de w deben cumplir

$$\{\delta(s, a), \delta(s, a^2), \ldots, \delta(s, a^{2^N})\} \subseteq Q,$$

pero como en Q hay exactamente k estados y $2^N > k$, entonces, el principio del palomar asegura que deben de existir i, j tales que

$$\delta(s, a^i) = \delta(s, a^j), \quad 2^N \geq j > i$$



por lo que, en el camino hacia aceptación, el bucle puede ser despreciado, o bien utilizado varias veces. Si se usa dos veces se obtiene que $a^{2^N+(j-i)} \in$

$L(M)$. En efecto: podemos poner $2^N = j + m$; luego

$$\begin{aligned}
\delta(s, a^{2^N+(j-i)}) &= \delta(s, a^{j+m+(j-i)}) \\
&= \delta(\delta(s, a^j), a^{m+j-i}) \\
&= \delta(\delta(s, a^i), a^{m+j-i}) \\
&= \delta(s, a^{i+m+j-i}) \\
&= \delta(s, a^{j+m}) \\
&= \delta(s, a^{2^N}) \in F.
\end{aligned}$$

esto es, $a^{2^N+(j-i)}$ es aceptada, por lo que $a^{2^N+(j-i)} \in C$. Pero $2^N + j - i$ NO es una potencia de 2, porque

$$2^N < 2^N + (j-i) < \underbrace{2^N + 2^N}_{=2^{N+1}}$$

lo que indica que el número $2^N + (j-i)$ está entre dos potencias consecutivas de 2, de donde $2^N + (j-i)$ no puede ser él mismo una potencia de 2. Por lo tanto $a^{2^N+(j-i)} \notin C$: contradicción. □

Repitiendo el razonamiento de los dos ejemplos anteriores, se puede demostrar el siguiente teorema general.

Teorema 14 (Lema del bombeo)**.** *Sea A un lenguaje regular. La siguiente propiedad la cumple A:*

(P) *Existe $k \geq 0$ tal que para cualesquiera x, y, z cadenas con*

$$xyz \in A \text{ y } |y| \geq k$$

existen cadenas u, v, w tales que

$$y = uvw \text{ con } v \neq \epsilon$$

y para todo $i \geq 0$ se cumple

$$xuv^iwy \in A$$

Sea puede escribir la propiedad (P) en términos de cuantificadores:

(P) $\exists(k \geq 0)$ $\forall(x, y, z$ con $xyz \in A$ y $|y| \geq k)$ $\exists(u, v, w$ con $y = uvw$ y $v \neq \epsilon)$ $\forall(i \geq 0)$ $(xuv^iwy \in A)$

luego la negación de (P) es:

$\neq$(P) $\forall(k \geq 0)$ $\exists(x, y, z$ con $xyz \in A$ y $|y| \geq k)$ $\forall(u, v, w$ con $y = uvw$ y $v \neq \epsilon)$ $\exists(i \geq 0)$ $(xuv^iwy \notin A)$

Lo que da lugar a la forma contrapositiva del lema del bombeo.

Teorema 15 (Lema del bombeo)**.** *Si A es un lenguaje tal que*

(1) $\forall(k \geq 0)$

(2) $\exists(x, y, z$ *con* $xyz \in A$ *y* $|y| \geq k)$

(3) $\forall(u, v, w$ *con* $y = uvw$ *y* $v \neq \epsilon)$

(4) $\exists(i \geq 0)$

(5) $(xuv^iwy \notin A)$

entonces A *no es regular.*

Esta versión contrapositiva puede interpretarse como un juego: Ud versus *un demonio*. Tal juego se juega con un lenguaje A. Ud hace las veces del cuantificador existencial $\exists$, mientras que el demonio hace las veces del cuantificador universal $\forall$. Ud gana el juego si puede demostrar que A No es regular; el demonio gana si logra impedir esto.

La descripción del juego es la siguiente:

(1) El demonio escoge $k \geq 0$.

(2) Ud. debe responder con tres cadenas x, y, z tales que $xyz \in A$ y $|y| \geq q$.

(3) El demonio elige otras tres cadenas u, v, w tales que $y = uvw$ y $v \neq \epsilon$.

(4) Ud. elige un $i \geq 0$

(5) Ud. gana si $xuv^iwz \notin A$, el demonio gana en caso contrario.

Ejemplo 64. *Mostrar que*

$$A = \{a^n b^m \mid n \geq m\}$$

no es regular.

Dem. Como un juevo vs demonio.

(1) Sea $k \geq 0$ (la elección del demonio).

(2) Nosostros elegimos $x = a^k, y = b^k, z = \epsilon$ con $xyz = a^k b^k \epsilon = a^k b^k \in A$ y $|y| \geq k$.

(3) El demonio elige palabras u, v, w tales que $b^k = y = uvw$ con $v \neq \epsilon$ De aquí se deduce que $u = b^\ell, v = b^m, w = b^n$ con $m > 0$.

(4) Nosotros tenemos que responder con un número $i \geq 0$ tal que nos haga ganar el juego. El ganar o perder depende de si la contatenación xuv^iwz está o no en el lenguaje A. Calculemos tal concatenación:

$$\begin{aligned} xuv^iwz &= a^k b^\ell b^{im} b^n \\ &= a^k b^\ell b^m b^n b^{im-m} \\ &= a^k b^k b^{im-m} \\ &= a^k b^{k+(i-1)m} \end{aligned}$$

y ésta palabra no pertenecerá a A cuando por ejemplo, $i = 2$.

Respondemos con $i = 2$.

(5) Se calcula xuv^iwz para saber quien ganó el juego:

$$\begin{aligned} xuv^iwz &= xuv^2wz \\ &= a^k b^\ell b^{2m} b^n \epsilon \\ &= a^k b^\ell b^m b^n b^m \\ &= a^k b^k b^m \\ &= a^k b^{k+m} \notin A \end{aligned}$$

Así que nosotros ganamos, esto es, A es no regular. □

Tarea 12.

(1) *Probar que el lenguaje*

$$L = \{a^p \,|\, p \text{ es primo}\}$$

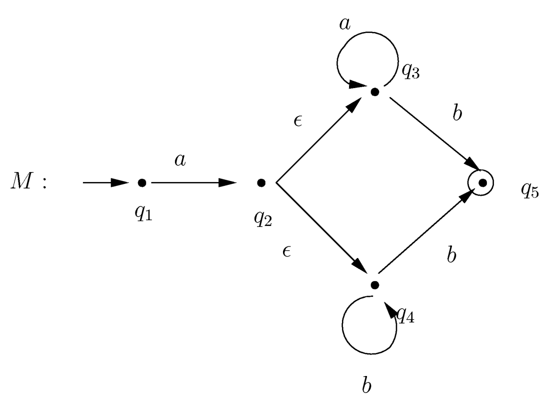
no es regular. item Dewterminar si los siguientes lenguajes son regulares y decir o probar por qué si o por qué no.

(a) $\{a^i b^{2i} \,|\, i \geq 1\}$
(b) $(ab^i \,|\, i \geq i$
(c) $\{a^{2n} \,|\, n \geq 0\}$
(d) $\{a^{2^n} \,|\, n \geq 0\}$

Lenguajes Independientes del Contexto

1. Gramáticas regulares

Un autómata finito puede considerarse como un generador de lenguajes: si M es un AF entonces genera a $L(M)$. Por ejemplo, consideremos



es tal que

$$L(M) = a(a^* \cup b^*)b$$

Las cadenas aceptadas por M se empiezan a producir como:

$$S \rightarrow aE$$

(la flecha se leeerá "produce"), donde S es un símbolo lallamdo *inicial* y E es un símbolo llamado "no terminal". A su vez, el símbolo E tiene dos posibilidades subsecuentes:

$$E \rightarrow A$$
ó
$$E \rightarrow B$$

dependiente de la ϵ-transición hacia q_3 ó q_4, donde A y B son también símbolos no terminales. Si A indica el camino superior, tenemos

$$A \rightarrow aA$$
ó
$$A \rightarrow b$$

donde ahora b es un *símbolo terminal*. Similarmente, para el camino inferior,

$$B \rightarrow bB$$
ó
$$B \rightarrow b$$

En resumen, tenemos

$$S \rightarrow aE$$
$$E \rightarrow A$$
$$E \rightarrow B$$
$$A \rightarrow aA$$
$$A \rightarrow b$$
$$B \rightarrow bB$$
$$B \rightarrow b$$

En forma más compacta, si "—"="$\vee$":

(1) $S \rightarrow aE$
(2) $E \rightarrow A \,|\, B$
(3) $A \rightarrow aA \,|\, b$
(4) $B \rightarrow bB \,|\, b$

las anteriores se consideran **regas de substitución** para la generación de cadenas. Por ejemplo: tomemos S:

$$S$$

la substituimos por aE:

$$aE$$

luego substituimos A por E (también pudimos haber substituido E por B):

$$aA$$

luego substituimos A por aA:

$$aaA$$

y luego A por b:

$$aab$$

Todo este proceso de substituciones se puede abreviar como

$$\begin{aligned} S &\overset{(1)}{\Rightarrow} aE \\ &\overset{(2)}{\Rightarrow} aA \\ &\overset{(3)}{aaA} \\ &\overset{(3)}{aab} \end{aligned}$$

El símbolo "$\Rightarrow$"se lee "deriva".

También la cadena a^3b se puede generar como

$$\begin{aligned} S &\overset{(2)}{\Rightarrow} aE \\ &\overset{(2)}{\Rightarrow} aA \\ &\overset{(3)}{\Rightarrow} aaA \\ &\overset{(3)}{\Rightarrow} aaaA \\ &\overset{(3)}{\Rightarrow} aaab \end{aligned}$$

Definición 36. *Si $w \in \Sigma^*$, se usa $S \overset{*}{\Rightarrow} w$ para indicar que w se generó a partir de S en cero o más etapas.*

Ejemplo 65. $S \overset{*}{\Rightarrow} a^2b$, $S \overset{*}{\Rightarrow} a^3b$.

En las producciones hay símbolos de un alfabeto Σ. Tales se llaman **terminales** para indicar que no son suceptibles a ser substituidos. Los símbolos que sí pueden serlo se llaman **no terminales**. El símbolo inicial S es siempre no terminal.

La generación de cadenas se hace de izuierda a derecha, por lo que en las producciones, los símbolos no terminales deben de aparecer a la derecha.

Definición 37. *En la producción*

$$A \to \gamma$$

A se llama **cabeza de la producción** *y γ se llama* **cuerpo de la producción**.

Definición 38. *Una* **gramática regular** *es una 4-tupla,*

$$G = (\Sigma, N, S, P)$$

donde

- *Σ es un alfabeto*
- *N un conjunto finito de símbolos no terminales*
- *$S \in N$ llamado* **símbolo inicial**
- *P es una colleción de reglas de substitución llamadas* **poducciones** *que son de la forma*
$$A \to \gamma$$
con $A \in N$ y $\gamma \in (\Sigma \cup N)^*$ tal que
(1) *u tiene un no terminal como máximo*
(2) *si γ contiene un no terminal, este está en el exremo derecho.*

Se pude decir que en la producción $A \to \gamma$, $\gamma \in \Sigma^*(N \cup \epsilon)^*$ con ϵ terminal.

Definición 39. *El* **lenguaje generado por** *G se denota por $L(G)$ y este es*

$$L(G) = \{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$$

Ejemplo 66. Sea $G = (\Sigma, N, S, P)$ gramática regular donde

$$\Sigma = \{a, b\}, \quad N = \{S, A\}$$

y las producciones son

$$\begin{aligned} P: \quad & S \to bA \\ & A \to aaA \mid b \mid \epsilon \end{aligned}$$

entonces

$$\begin{aligned} S &\Rightarrow bA \\ &\Rightarrow baa \\ &\Rightarrow baab \end{aligned}$$

por lo que $baab \in L(G)$. También,

$$\begin{aligned} S &\Rightarrow bA \\ &\Rightarrow baaA \\ &\Rightarrow baaaaA \\ &\Rightarrow baaaab \end{aligned}$$

, $b(aa)^2b \in L(G)$. En general,

$$\begin{aligned}
S &\Rightarrow bA \\
&\Rightarrow baaA \\
&\Rightarrow b(aa)^2A \\
&\vdots \\
&\Rightarrow b(aa)^nA \\
&\Rightarrow b(aa)^nb
\end{aligned}$$

por lo que

$$\forall n \geq 1, ba^{2n}b = b(aa)^nb \in L(G).$$

También

$$\begin{aligned}
S &\Rightarrow bA \\
&\Rightarrow baaA \\
&\Rightarrow b(aa)^2A \\
&\vdots \\
&\Rightarrow b(aa)^nA \\
&\Rightarrow b(aa)^n\epsilon = b(aa)^n = ba^{2n} \in L(G).
\end{aligned}$$

Nótese que

$$\begin{aligned}
S &\Rightarrow bA \\
&\Rightarrow b\epsilon = b \in L(G)
\end{aligned}$$

y

$$\begin{aligned}
S &\Rightarrow bA \\
&\Rightarrow bb = b^2 \in L(G)
\end{aligned}$$

por lo que

$$b(a^2)^*(\epsilon \cup b) = b(a^2)^* \cup b(a^2)^*b \subseteq L(G)$$

Recíprocamente $L(G) \subseteq b(a^2)^*(\epsilon \cup b)$, porque si $w \in L(G)$,

$$S \Rightarrow bA \Rightarrow \cdots \Rightarrow w_1 \Rightarrow w$$

onde las primeras derivaciones $S \Rightarrow bA \Rightarrow \cdots \Rightarrow w_1$ ninguna de ellas proviene de aplicar $A \to b$y/ó $A \to \epsilon$. Es decir, éstas derivaciones resultan de aplicar la producción $A \to aaA$. Por lo que

$$w_1 = b(aa)^{k-1}A$$

donde k es el número de deriaciojes aplicadas. Luego,

$$S \Rightarrow bA \Rightarrow \cdots \Rightarrow b(aa)^{k-1}A \Rightarrow w$$

donde la última derivación es de $A \to b$ ó $A\epsilon$. Se deduce que

$$w = b(aa)^{k-1}b \text{ 'o } w = b(aa)^{k-1.}$$

Así $w \in b(a^2)^*b \cup b(a^2)^*$.

Las producciones son pares: en el ejemplo anterior

$$P = \{(S, bA), (A, aaA), (A, b), (A, \epsilon)\}.$$

Únicamente los símbolos no terminales se escriben con mayúsculas, así una gramática regular queda descrita completamente por sus producciones. Por ejemplo

$$S \to aS \mid b$$

describre a la gramática regular que genera al lenguaje a^*b.

2. Gramáticas regulares y lenguajes regulares

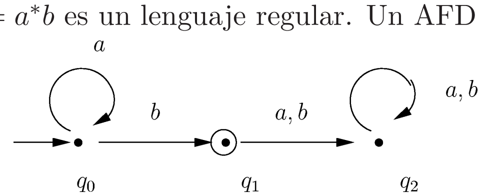
Supongamos que L es un lenguaje regular. Se puede obtener una gramática regular que genera L. Para esto será útil M un AFD tal que $L = L(M)$.

Supongamos que $M = (Q, \Sigma, s, F, \delta)$. Se define $G = (N, \Sigma, S, P)$ mediante

$$N = Q, \quad \Sigma = \Sigma, \quad S = s$$

$$P: \quad \forall q \in Q, \begin{cases} q \to ap & \text{si } \delta(q,a) = p \\ q \to \epsilon & \text{si } q \in F \end{cases}$$

Ejemplo 67. $L = a^*b$ es un lenguaje regular. Un AFD que acepta L es



Una gramática que aceptará a $L = a^*b$ es

$$q_0 \to aq_0 \mid bq_1$$
$$q_1 \to \epsilon \mid aq_2 \mid bq_2$$
$$q_2 \to aq_2 \mid bq_2$$

donde q_0, q_1, q_2 son no terminales. En efecto, tenemos el siguiente teorema general

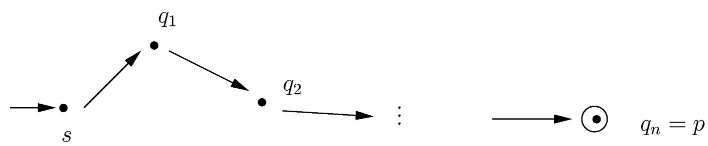
Teorema 16. *Si M es un AFD y G la gramática descrita anteriormente, entonces*

$$L(M) = L(G).$$

Proof. Si $w \in L(M)$ entonces $w = \sigma_1 \cdots \sigma_n$ con cada $\sigma_i \in \Sigma$. Así $\delta(s, \sigma_1 \cdots \sigma_n) = p \in F$. Pongamos $s = q_0$ y

$$\begin{aligned}
\delta(q_0, \sigma_1) &= q_1 \\
\delta(q_1, \sigma_2) &= q_2 \\
&\vdots \\
\delta(q_{n-2}, \sigma_{n-1}) &= q_{n-1} \\
\delta(q_{n-1}, \sigma_n) &= q_n = p
\end{aligned}$$

eso es:



Los anteriores inducen las siguientes producciones en G:

$$\begin{aligned}
0)&\ \ q_0 \to \sigma_1 q_1 \\
1)&\ \ q_1 \to \sigma_2 q_2 \\
&\ \ \vdots \\
n-2)&\ \ q_{n-2} \to \sigma_{n-1} q_{n-1} \\
n-1)&\ \ q_{n-1} \to \sigma_n q_n \\
n)&\ \ p = q_n \to \epsilon
\end{aligned}$$

luego

$$\begin{aligned}
s &\overset{(0)}{\Rightarrow} \sigma_1 q_1 \\
&\overset{(1)}{\Rightarrow} \sigma_1 \sigma_2 q_2 \\
&\overset{(2)}{\Rightarrow} \sigma_1 \sigma_2 \sigma_3 q_3 \\
&\vdots \\
&\overset{(n-1}{\Rightarrow} \sigma_1 \sigma_2 \cdots \sigma_n q_n \\
&\overset{(n)}{\Rightarrow} \sigma_1 \sigma_2 \cdots \sigma_n \epsilon = \sigma_1 \sigma_2 \cdots \sigma_n
\end{aligned}$$

lo que implica que $w = \sigma_1 \sigma_2, \cdots \sigma_n$ es generada por G, i.e., $w \in L(G)$. Hemos demostrado

$$L(M) \subseteq L(G).$$

Recíprocamente, supongamos $w \in L(G)$ entoncea w fué derivada como

$$\begin{aligned} s = q_0 &\Rightarrow \sigma_1 q_1 \\ &\Rightarrow \sigma_1\sigma_2 q_2 \\ &\Rightarrow \sigma_1\sigma_2\sigma_3 q_3 \\ &\vdots \\ &\sigma_1 \cdots \sigma_n q_n \\ &\sigma_1 \cdots \sigma_n \epsilon = \sigma_1 \cdots \sigma_n = w \end{aligned}$$

luego

$$\begin{aligned} \delta(s,w) &= \delta(s, \sigma_1 \cdots \sigma_n) \\ &= \delta(\delta(s,\sigma_1), \sigma_2 \cdots \sigma_n) \\ &= \delta(q_1, \sigma_2 \cdots \sigma_n) && \text{pues } s \to \sigma_1 q_1 \Leftrightarrow \delta(s,\sigma_1) = q_1 \\ &= \delta(\delta(q_1,\sigma_2), \sigma_3 \cdots \sigma_n) && \text{pues } q_1 \to \sigma_2 q_2 \Leftrightarrow \delta(q_1,\sigma_2) = q_2 \\ &= \delta(q_{n-1}, \sigma_n) \\ &= q_n \in F && \text{pues } q_n \to \epsilon \text{sólo es posible con } q_n \in F. \end{aligned}$$

Por tanto $w \in L(M)$. Hemos demostrado

$$L(G) \subseteq L(M).$$

□

El teorema recíproco también es cierto.

Teorema 17. *Sea G una gramática regular. Entonces existe M un AFN tal que*

$$L(M) = L(G).$$

Proof. Sea $G = (N, \Sigma, S, P)$ una gramática regular. Se define M un AFN tal que

$$M = (Q, \Sigma, s, F, \Delta)$$

con

$$s = S, \quad F = \{f\} \text{ donde } f \notin \Sigma, f \notin N$$

Los estados $Q \supseteq N \cup \{f\}$ y falta añadir estados a Q lo cual haremos en la definición de Δ:

(1) Si $A \to \sigma_1 \cdots \sigma_n B$ es produccción de G con A, B no terminales, entonces se añaden estados $q_1, \ldots, q_{n-1}$ a Q y se define

$$\Delta(A, \sigma_1) = q_1, \Delta(q_1, \sigma_2) = q_2, \ldots, \Delta(q_{n-1}, \sigma_n) = B,$$

esto es, la producción $A \to \sigma_1 \cdots, \sigma_n B$ en G se transforma en

$$A \overset{\sigma_1}{\to} q_1 \overset{\sigma_2}{\to} q_2 \to \cdots \to q_{n-1} \overset{\sigma_n}{\to} B$$

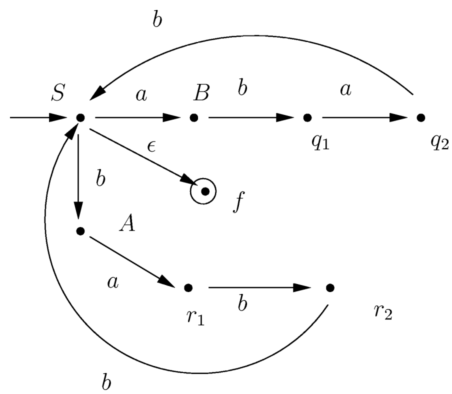
(2) Si $A \to \sigma_1 \cdots \sigma_n$ (cadena de terminales) entonces se añaden los estados $r_1, r_2, \ldots, r_{n-1}$ a Q y se define en el autómata M,

$$A \xrightarrow{\sigma_1} r_1 \xrightarrow{\sigma_2} r_2 \to \cdots \to r_{n-1} \xrightarrow{\sigma_n} f$$

Por ejemplo, si G es la gramática regular

$$S \to aB \mid bA \mid \epsilon$$
$$A \to abaS$$
$$B \to babS$$

da lugar al autómata



Así $L(M) = L(G)$ pues si $w \in L(G)$ entones w debe ser generada por derivaciones:

$$\begin{aligned} S &\Rightarrow \alpha_1 A_1 && \text{con } \alpha_1 \in \Sigma^* \text{ y } A_1 \text{ no terminal} \\ &\Rightarrow \alpha_1\alpha_2 A_2 && \text{con } \alpha_2 \in \Sigma^* \text{ y } A_{\to}\alpha_2 A_2 \text{ en } G \\ &\vdots \\ &\Rightarrow \alpha_1 \cdots \alpha_{n-1} A_{n-1} && \text{con } a_n \in \Sigma^* \text{ y } \alpha_n \in \Sigma^* \end{aligned}$$

que en el AFN, ésta construcción queda como

$$\to s \quad \underbrace{\to \cdots \to}_{\text{letras que forman } \alpha_1} \quad A_1 \quad \underbrace{\to \cdots \to}_{\text{letras que forman } \alpha_2} \quad A_2 \to \cdots \to A_{n-1} \quad \underbrace{\to \cdots \to}_{\text{letras que forman } \alpha_n} \quad f \in F$$

Recíprocamente, si $w \in L(M)$, entonces siguiendo el razonamiento anterior $s \overset{*}{\Rightarrow} w$, por lo que $w \in L(G)$. □

Tenemos

Corolario 2. *L es lenguaje regular $\Leftrightarrow$ L es generado por una gramática regular.*

Tarea 13.

(1) *Supongamos que tenemos las reglas* $S \rightarrow aS|bT$ *y* $T \rightarrow aa$. *Dar una derivación para abaa, aabaa, aaabaa. Probar que se pueden derivar* $a^k ba^2$ *para* $k \geq 1$. *¿Es posible derivar las cadenas baa, b o aa?*

(2) *Obtener una gramática regular que genere, en cada caso, los siguientes lenguajes*
   (a) $a^*b \cup a$
   (b) $a^*b \cup b^*a$
   (c) $(a^*b \cup b^*a)^*$

(3) *La gramática regular dada por*

$$S \rightarrow bA|aB|\epsilon$$
$$A \rightarrow abaS$$
$$B \rightarrow babS$$

genera un lenguaje regular. Obtener una expresión regular para este lenguaje.

(4) *En nuestra definición de gramáticas regulares de dijo que si en el extremo derecho de una producción hay un no terminal, éste debe de estar situado en el extremo derecho. Esto corresponde a la generación de cadenas de izquierda a derecha. Por esta razón, una gramática regular también puede llamarse gramática regular por la derecha. Una gramática regular por la izquierda es aquella cuyas cadenas son generadas por la derecha, es decir sus porducciones son pares de* $N \times (N \cup \Sigma)\Sigma^*$.

(1) *Obtener una gramática regular por la izquierda para el lenguaje* $\{a^n baa \mid n \geq 0\}$.

(2) *Obtener las gramáticas regulares por la derecha y la izquierda para*

$$\{w \in \{a, b, c\}^* \mid w \text{ termina en } b \text{ y toda } c \text{ va seguida de una } a\ \}$$

(3) *Para toda gramática* $G = (N, \sigma, S, P)$ *que sea regular (por la izquierda o por la derecha), se puede definir la inversa de* G *como* $G^I = (N, \Sigma, S, P')$ *donde*

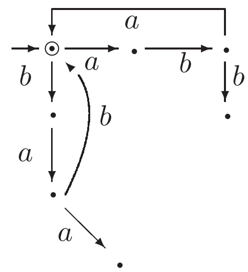
$$P' = \{(A, x^I) \mid (A, x) \in P\}.$$

Por lo tanto, si $A \rightarrow aB$ *es una producción de* G, *entonces,* $A \rightarrow Ba$ *es una producción de* G^I.

Supongamos que G *es una gramática regular por la derecha.*
   (a) *Probar que* G^I *es una gramática regular por la izquierda.*
   (b) *Probar que* $w \in L(G)$ *si y sólo si* $w^I \in L(G')$ *por inducción sobre el número de producciones usadas para obtener* w.

Se puede deducir de la parte (c) que la clase de los lenguajes generados por gramáticas regulares por la izquierda es la misma que la clase de lenguajes generados por gramáticas gegulares por la derecha. Por eso habitualmente, el término gramática regular se aplica para referirse a cualquier gramática ya sea regular por la izquierda o regular por la derecha.

(4) *Construir la gramática regular para el lenguaje aceptado por el autómata finito siguiente:*



(5) *Construir un autómata finito para la gramática regular*

$$S \rightarrow abA|B|baB|\epsilon$$
$$A \rightarrow bS|b$$
$$B \rightarrow aS$$

(6) *Obtener una gramática regular para el lenguaje*

$$L = \{w \in \{a,b\}^* \,|\, w \text{ no contiene la subcadena } aa\}$$

(7) *Obtener una gramática regular para* $L = \{a^n b^n \,|\, n \geq 0\}$.

3. Gramáticas independientes del contexto

Recordemos que en una gramática regular G, las producciones son parejas:

$$P: \quad \begin{cases} A \rightarrow wB \\ \text{ó} \\ A \rightarrow w \end{cases}$$

con A, B no terminales y $w \in \Sigma^*$ cadena de terminales. Éstas corresponden a las parejas:

$$(A, wB) \text{ ó } (A, w)$$

así

$$P \subseteq N \times \Sigma^*(N \cup \{\epsilon\})$$

esto es, en las producciones, los terminales, si aparecen, tienen que estar en el extremo derecho del cuerpo de la producción.

Si se permiten que los terminales aparezcan, más de una vez, y en cualquier lado, i.e.,

$$P \subseteq N \times (N \cup \Sigma)^*$$

entonces se obtienen *gramáticas independientes del contexto.*

Ejemplo 68. La gramática

$$\begin{aligned} S &\to aB \,|\, bA \\ A &\to a \,|\, aS \,|\, bAA \\ B &\to b \,|\, bS \,|\, aBB \end{aligned}$$

es una gramática independiente del contexto que no es regular.

Definición 40. *Una* **gramática independiente del contexto** *(GIC) es una 4-tupla*

$$G = (N, |sigma, S, P)$$

donde

- N *es colección de símbolos no terminales*
- Σ *es un alfabeto*
- $S \in N$ *símbolo inicial*
- $P \subseteq N \times (N \cup \Sigma)^*$ *conjunto de producciones.*

El lenguaje que genera G*,* $L(G)$ *se llama* **lenguaje independiente del contexto***:*

$$L(G) = \{w \in \Sigma^* \,|\, S \overset{*}{\Rightarrow} w\}.$$

Ejemplo 69. Consideremos como G la grámatica independiente del contexto (que no es regular) siguiente:

$$S \to aSb \,|\, \epsilon.$$

entonces

$$S \Rightarrow \epsilon$$

i.e., $\epsilon \in L(G)$. Además

$$\begin{aligned} S &\Rightarrow aSb \\ &\Rightarrow a\epsilon b = ab \end{aligned}$$

i.e., $ab \in L(G)$. También

$$\begin{aligned} S &\Rightarrow aSb \\ &\Rightarrow aaSbb \\ &\Rightarrow aa\epsilon bb = a^2b^2 \end{aligned}$$

i.e., $a^2b^2 \in L(G)$. En general

$$\{a^n b^b \,|\, n \geq 0\} \subseteq L(G).$$

Recíprocamente, si $w \in L(G)$, entonces

$$\begin{aligned} S &\Rightarrow aSb \\ &\Rightarrow aaSbb \\ &\vdots \\ &\Rightarrow a^nSb^n \\ &\Rightarrow a^n\epsilon b^n = w \end{aligned}$$

i.e., $w = a^nb^n$ para algún n. Por lo tanto

$$L(G) = \{a^nb^n \mid n \geq 0\}$$

es un lenguaje independiente del contexto que no es regular, como puede notarse de la siguiente propiedad.

Propiedad 7. *El lenguaje $L = \{a^mb^m \mid m \geq 0\}$ no es regular.*

Proof. Supongamos que L es regular. Sea n la cosntante del lema del bombeo para L y consideremos la palabra $w = a^nb^n$ que evidentemente pertenece a L. Tenemos que $|w| = 2n \geq n$. Entonces podemos escribir

$$w = uvx$$

con $|uv| \leq n$, $|v| \geq 1$ y $\forall i \geq 0$, $uv^ix \in L$.

Como $|uv| \leq n$ entonces uv está formado sólo por a's:

$$w = \underbrace{a\cdots a}_{uv}\underbrace{\cdots ab\cdots b}_{x}$$

así $u = a^r$, $v = a^s$ para algunos $o \leq r \leq n$, $1 \leq s \leq n$. Lo que implica

$$x = a^{n-r-s}b^n$$

luego, bombeamos con $i = 2$:

$$\begin{aligned} L \ni uv^2x &= a^ra^2sa^{n-r-s}b^n \\ &= a^{n+s}b^n \end{aligned}$$

con $s \geq 1$. Lo cual es imposible.

Por tanto L no es regular. $\square$

El nombre *independiente del contexto* es para diferenciarlas de las gramáticas que son sensibles al contexto; en éstas las producciones son de la forma

$$\alpha_1A\alpha_2 \rightarrow \alpha_1B\alpha_2$$

i.e, dependen del contexto.

Tarea 14.

(1) *Dada la gramática independiente del contexto*

$$S \to AA$$
$$A \to AAA|a|bA|Ab$$

(a) *Obtener una derivación para la cadena* b^2aba^2ba

(2) *Dada la gramática independiente del contexto*

$$S \to AA$$
$$A \to AAA|a|bA|Ab$$

(a) *Obtener una derivación para la cadena* b^2aba^2ba
(b) *Probar cómo puede obtenerse una derivación para* $b^{m_1}ab^{m_2}a\ldots b^{m_{2n}}ab^{m_{2n+1}}$, *para todo* $n > 0$ *y* $m_1, m_2, \ldots, m_{2n+1} \geq 0$.

(3) *La gramática* G *independiente del contexto dada por*

$$S \to aSb|aSa|bSa|bSb|\epsilon$$

no es una gramática regular, aunque $L(G)$ *es un lenguaje regular! Obtener una gramática regular* G' *tal que* $L(G') = L(G)$.

(4) *Obtener una gramática independiente del contexto para cada uno de los siguientes lenguajes independientes del contexto:*
(a) $\{a^m b^n \mid m \geq n\}$
(b) $\{a^m b^n \mid n \leq m \leq 2n\}$

4. Árboles de derivación ó análisis

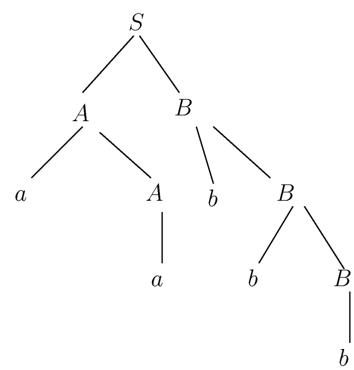
Una forma de visualizar las derivaciones de la gramáticas independientes del contexto es con los *árboles de derivación* ó *árboles de análisis*. Este cosniste de un nodo raíz que es el símbolo inicial. Luego el nodo raíz tiene nodos hijos, uno por cada símbolo que aparezca en el cuerpo de la producción usada para reeemplazar el símbolo inicial. Cada nodo no terminal tendrá hijos a su vez dados por la producción usada al substituir. Por ejemplo, consideremos la gramática G:

(1) $S \to AB$
(2) $A \to aA \mid a$
(3) $B \to bB \mid b$

entonces $aabbb \in L(G)$ pues puede ser generada por

$$\begin{aligned} S &\overset{(1)}{\Rightarrow} AB \\ &\overset{(3)}{\Rightarrow} AbB \\ &\overset{(3)}{\Rightarrow} AbbB \\ &\overset{(3)}{\Rightarrow} Abbb \\ &\overset{(2)}{} aAbbb \\ &\overset{(2)}{} aaabbb \end{aligned}$$

que le corresponde árbol de derivación

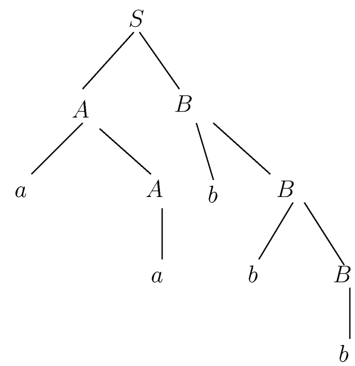


Si leemos las hojas de izquierda a derecha obtenemos $aabbb$.

Nótese que se puede derivar a^2b^3 de muchas formas:

$$\begin{aligned} S &\overset{(1)}{\Rightarrow} AB \\ &\overset{(2)}{\Rightarrow} aAB \\ &\overset{(2)}{\Rightarrow} aaB \\ &\overset{(3)}{\Rightarrow} aabB \\ &\overset{(3)}{\Rightarrow} aabbB \\ &\Rightarrow aabbb \end{aligned}$$

cuyo árbol de derivación resulta:



que es el mismo que antes. Otra posible derivación de a^2b^3 es

$$\begin{aligned} S &\Rightarrow AB \\ &\Rightarrow aAB \\ &\Rightarrow aAbB \\ &\Rightarrow aAbbB \\ &\Rightarrow aAbbb \\ &\Rightarrow aabbb \end{aligned}$$

cuyo árbol de derivación es de nuevo, el mismo que antes. Resulta que todos los árboles de derivación de a^2b^3 tienen el mismo árbol. *Este no es siempre el caso*. Por ejemplo, en la gramática,

$$S \to SbS \mid ScS \mid a$$

se puede derivar la cadena $abaca$ de dos formas:

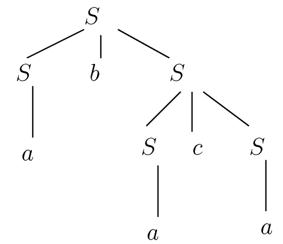
(1)

$$\begin{aligned} S &\Rightarrow SbS \\ &\Rightarrow SbScS \\ &\Rightarrow SbSca \\ &\Rightarrow Sbaca \\ &\Rightarrow abaca \end{aligned}$$

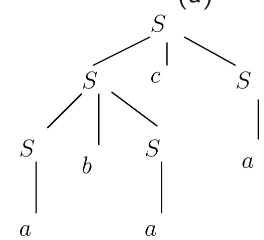
(2)

$$\begin{aligned} S &\Rightarrow ScS \\ &\Rightarrow SbScS \\ &\Rightarrow abScS \\ &\Rightarrow abacS \\ &\Rightarrow abaca \end{aligned}$$

cuyos árboles de derivación son:



(a)



(b)

que son diferentes.

Las cadenas derivadas corresponden a las hojas de los árboles de derivación y se llaman **producto** del árbol de derivación .

Definición 41. *Una gramática G se llama*

(1) **ambigua** *si $\exists w \in L(G)$ tal que w tiene al menos dos árboles de derivación diferentes.*

(2) **no ambigua** *si $\forall w \in L(G)$, w tiene todos sus árboles de derivación iguales.*

La ambiguedad ocurre en los lenguajes "naturales". Por ejemplo

Pedro vió a un hombre con un telescopio

significa que ¿ Pedro usó un telescopio para ver a un hombre? ó que ¿ Pedro vió a un hombre que tenía un telescopio?.

También las ambiguedades aparecen en las expresiones algebraicas:

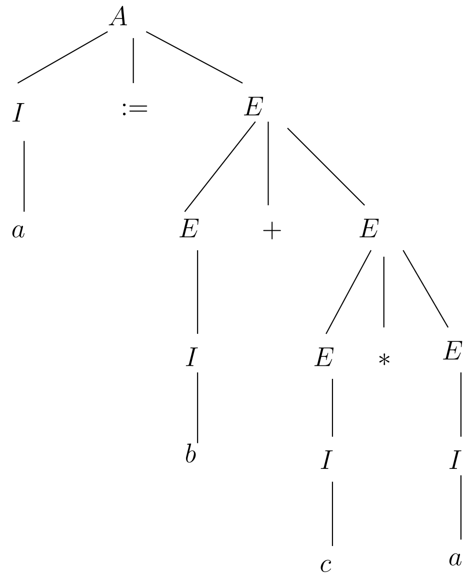
Ejemplo 70. Consideremos la gramática

$$A \to I := E$$
$$I \to a \,|b\,|\, c$$
$$E \to E + E\,|E * E\,|\,(E)\,|\,I$$

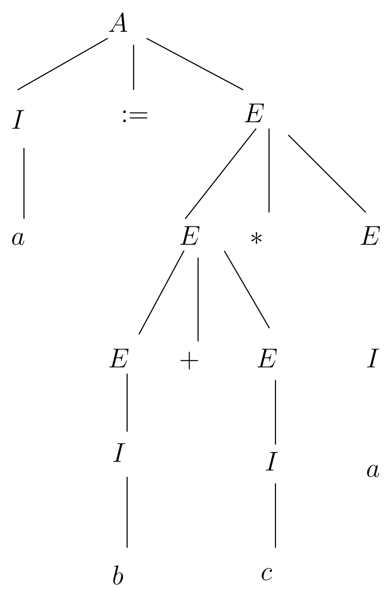
(las mayúsculas son no terminales; los demás símbolos, ":=", "*", "(","(" son terminales). La cadena

$$a := b + c * a$$

se puede derivar de formas diferentes:



ó



En algunos casos , se puede encontrar una gramática que produzca el mismo lenguaje, pero que no sea ambigua. Si esto no es posible, entonces el lenguaje se llama *inherentemente ambiguo*.

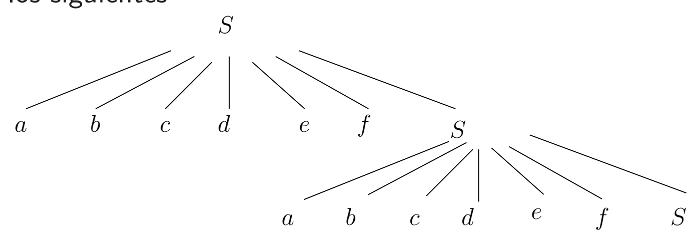
Tarea 15. *Demostrar que la gramática es ambigua*

$$S \to bA|aB$$
$$A \to a|aS|bAA$$
$$B \to b|bS|aBB$$

5. Simplificación de las GIC

Las gramáticas pueden tener árboles de derivación innecesariamente complicados:

(1) $S \to abcdefgS \,|\, abcdefg$ que le corresponde árboles de derivación como los siguientes



(2)

$$S \to A$$
$$A \to B$$
$$B \to C$$
$$C \to D$$
$$D \to a \,|\, A$$

Lo importante de una gramática es el lenguaje que genera. Se pretende tener gramáticas razonables que generen los mismos lenguajes: lo primero que se hace es eliminar las producciones y símbolos inútiles.

Definición 42. *Sea* $G = (N, \Sigma, S, P)$ *una GIC y* $X \in N \cup \Sigma$. *Se dice que*

(1) X *es* **útil** *si existe una derivación de la forma*

$$S \stackrel{*}{\Rightarrow} \alpha X \beta \stackrel{*}{\Rightarrow} w$$

con $w \in \Sigma^*$.

(2) X *es* **inútil** *si no es útil.*

(3) X *es* **generador** *si* $X \stackrel{*}{\Rightarrow} w$ *para algún* $w \in \Sigma^*$.

(4) X *es* **alcanzable** *si existe una derivación de la forma*

$$S \stackrel{*}{\Rightarrow} \alpha X \beta.$$

Ejemplo 71.

$$G: \qquad \begin{aligned} S &\to Aa \mid B \mid D \\ B &\to b \\ A &\to aA \mid bA \mid B \\ C &\to abd \end{aligned}$$

Obérvese que C nunca forma parte de una derivación de una palabra generada:

$$S \Rightarrow \cdots \Rightarrow \alpha C \beta \Rightarrow \cdots \Rightarrow w \in \Sigma^*$$

es imposible. Así C y la producción $C \to aba$ son inútiles.

Tampoco D es generador y por tanto nunca deriva (ó forma parte de una derivación productiva). En este sentido D es un símbolo inútil.

El siguiente algoritmo elimina los símbolos no terminales que no son generadores.

Ejemplo 72. Le aplicaremos el algoritmo descrito a

$$G: \qquad \begin{aligned} S &\to Aa \mid B \mid D \\ B &\to b \\ A &\to aA \mid bA \mid B \\ C &\to abd \end{aligned}$$

Comenzamos con $N' = \{B, C\}$ y

$$P': \qquad \begin{aligned} B &\to b \\ C &\to abd \end{aligned}$$

Algorithm 1: 3.5.1

Data: $G = (N, \Sigma, S, P)$ una GIC
Result: $G' = (N', \Sigma, S, P')$ una GIC tal que $L(G') = L(G)$ y

$$\forall A \in N', \quad A \overset{*}{\Rightarrow} w$$

con $w \in \Sigma^*$ (no necesariamente $A = S$).

```
begin
    Inicializar N' con los no terminales A para los cuales A → w es una
    producción de G, con w ∈ Σ*;
    Inicializar P' con las producciones A → w de G tales que w ∈ Σ*;
    repeat
        Añadir a N' todos los terminales A para los cuales A → w es una
        producción de G con w ∈ (N' ∪ Σ)*;
    until Hasta que no se puedan añadir más terminales a N';
end
```

1er pasada del ciclo:

$$N' = \{B, C\} \cup \{A\} \qquad P' : \qquad \begin{aligned} B &\to b \\ C &\to abd \\ A &\to B \end{aligned}$$

2da pasada del ciclo:

$$N' = \{B, C, A\} \cup \{S\} \qquad P' : \qquad \begin{aligned} B &\to b \\ C &\to abd \\ A &\to B \\ S &\to Aa \,|\, B \\ A &\to aA \,|\, bA \end{aligned}$$

3er pasada del ciclo: fin. queda

$$\begin{aligned} G : S &\to Aa \,|\, B \\ A &\to aA \,|\, bA \,|\, B \\ B &\to b \\ C &\to abd \end{aligned}$$

Se permiten producciones del tipo $A \to \epsilon$. Por ejemplo.

$$G: \qquad \begin{aligned} S &\to aA \mid \epsilon \\ A &\to aA \mid bB \mid \epsilon \\ B &\to bB \end{aligned}$$

Las inicializaciones del algoritmo anterior quedan:

$$N' = \{S, A\}, \qquad P': \qquad \begin{aligned} S &\to \epsilon \\ A &\to \epsilon \end{aligned}$$

1er pasada del ciclo:

$$N' = \{S, A\} \qquad P': \qquad \begin{aligned} S &\to \epsilon \mid aA \\ A &\to aA \mid bB \mid \epsilon \\ B &\to b \end{aligned}$$

2da pasada del ciclo:

$$N' = \{S, A\} \qquad P': \qquad \begin{aligned} S &\to \epsilon \\ A &\to \epsilon \\ S &\to aA \\ A &\to aA \end{aligned}$$

3er pasada: fin.

Salida:

$$G': \qquad \begin{aligned} S &\to aA \mid \epsilon \\ A &\to aA \mid \epsilon \end{aligned}$$

En el ejemplo 72, el algoritmo 3.5.1 no eliminó la última producción $C \to abc$ que no es productiva pues C no es $alcanzable$. El siguiente algoritmo elimina los $inalcanzables$

Ejemplo 73.

$$G: \qquad \begin{aligned} S &\to Aa \mid B \\ B &\to b \\ A &\to aA \mid bA \mid B \\ C &\to abd \end{aligned}$$

y le aplicamos el algoritmo 3.5.2:

(2) inicialización: $N' = \{S\}$, $P' = \varnothing$, $\Sigma' = \varnothing$.

Algorithm 2: 3.5.2

Data: $G = (N, \Sigma, S, P)$ una GIC.
Result: $G' = (N', \Sigma', S, P')$ una GIC tal que $L(G') = L(G)$ y todos los símbolos de N' son alcanzables.

1 **begin**
2   Inicializar $N' = \{S\}$, $P' = \varnothing$, $\Sigma' = \varnothing$;
3   **repeat**
4   **foreach** $A \in N'$ **do**
5   **foreach** $A \to w$ *en* P **do**
6   Introducir $A \to w$ en P';
7   Para todo B no terminal que forma w introducir B en N';
8   Para todo terminal σ que forma w introducir σ en Σ'
9   **end**
10   **end**
11   **until** *Hasta que no se puedan añadir más producciones a* P';
12 **end**

(3) **primer pasada**: tenemos $S \in N'$ ($A = S$ en el algoritmo) y

$$S \to Aa$$
$$S \to B$$

en P;

(6) $P' : S \to Aa \mid B$
(7) $N' = \{S, A, B\}$
(8) $\Sigma' = \{a\}$

(3) **2da pasada**: $S \in N'$ ya;

(4) $A \in N'$: (6)

$$P' : \quad \begin{aligned} &S \to Aa \mid B \\ &A \to aA \mid bA \mid B \end{aligned}$$

(7) $N' = \{S, A, B\}$
(8) $\Sigma' = \{a\} \cup \{b\} = \{a, b\}$

(4) $B \in N'$: (6)

$$P' : \quad \begin{aligned} &S \to Aa \mid B \\ &A \to aB \mid bA \mid B \\ &B \to b \end{aligned}$$

(7) $N' = \{S, A, B\}$
(8) $\Sigma' = \{a, b\}$

(3) **3er pasada**: no se pueden añadir más producciones: fin.

Salida:

$$G' : \quad \begin{aligned} S &\to Aa \mid B \\ A &\to aA \mid bA \mid B \\ B &\to b \end{aligned}$$

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

Nótese que se han eliminado C y d.

Finalmente, para quitar los símbolos inútiles e improductivos se aplican los algoritmos 3.5.1 y a la gramática de ésta salida se le aplica el algoritmo 3.5.2 y **en este orden**. Los algoritmos 3.5.1 y 3.5.2 no conmutan, i.e., el orden de su aplicación cambia las salidas. Por ejemplo, para la gramática

$$G : \quad \begin{aligned} S &\to AB \mid a \\ A &\to a \end{aligned}$$

puede notarse que B no es necesario. Le aplicamos a G el algoritmo 3.5.1 y a la gramática que resulte le aplicamos el 3.5.2 parta quitar los improductivos.

$$G \xrightarrow{3.5.1} \begin{cases} S \to a \\ A \to a \end{cases} \xrightarrow{3.5.2} \begin{cases} S \to a \end{cases}$$

siendo ésta última la gramática esperada. Pero si aplicamos los algoritmos en el otro orden

$$G \xrightarrow{3.5.2} \begin{cases} S \to AB \mid a \\ A \to a \end{cases} \xrightarrow{3.5.1} \begin{cases} S \to a \\ A \to a \end{cases}$$

que no es la respuesta esperada.

Tarea 16.

(1) *Aplicar el algoritmo 3.5.1 a las siguientes gramáticas:*

(a)

$$\begin{aligned} S &\to aAb|cEB|CE \\ A &\to dBE|eeC \\ B &\to ff|D \\ C &\to gFB|ae \\ D &\to h \end{aligned}$$

(b)

$$
\begin{aligned}
S &\to aB \\
A &\to bcCCC|dA \\
B &\to e \\
C &\to fA \\
D &\to Dgh
\end{aligned}
$$

(c)

$$
\begin{aligned}
S &\to a|aA|B|C \\
A &\to aB|\epsilon \\
B &\to Aa \\
C &\to bCD \\
D &\to ccc
\end{aligned}
$$

(2) *Aplicar el algoritmo 3.5.2 a las siguientes gramáticas independiente del contexto*

(a)

$$
\begin{aligned}
S &\to aAb \\
A &\to ccC \\
B &\to dd|D \\
C &\to ae \\
D &\to f \\
U &\to gW \\
W &\to h
\end{aligned}
$$

(b)

$$
\begin{aligned}
S &\to a|aA|B \\
A &\to aB|\epsilon \\
B &\to Aa \\
D &\to ddd
\end{aligned}
$$

(3) *Eliminar los símbolos inútiles de la siguiente gramática por medio de los algoritmos 3.5.1 y 3.5.2:*

$$
\begin{aligned}
S &\rightarrow A|AA|AAA \\
A &\rightarrow ABa|ACa|a \\
B &\rightarrow ABa|Ab|a \\
C &\rightarrow Cab|CC \\
D &\rightarrow CD|Cd|CEa \\
E &\rightarrow b
\end{aligned}
$$

6. Eliminación de las producciones ϵ

Definición 43. *Una producción del tipo $A \rightarrow \epsilon$ se llama* **producción** ϵ*.*

Sea G una GIC. Si $\epsilon \notin L(G)$ entonces se pueden eliminar todas las producciones ϵ. Si $\epsilon \in L(G)$ también se pueden eliminar todas la producciones ϵ exepto una.

Definición 44. *Si A es no terminal, A se dice* **anulable** *si*

$$A \stackrel{*}{\Rightarrow} \epsilon$$

El siguiente algoritmo calcula η el conjunto de todos los anulables y está basado en que si $A \rightarrow X_1 \cdots X_n$ y cada $X_i \stackrel{*}{\Rightarrow} \epsilon$ entonces A es anulable.

Algorithm 3: 3.5.3

Data: $G = (N, \Sigma, S, P)$ una GIC
Result: η el conjunto de todos los anulables

begin
  Inicio: $\eta = \{A \in N \mid A \rightarrow \epsilon\}$;
  repeat
    Si $B \rightarrow w$ en P y $w \in \eta^*$, añadir B a η
  until *hasta que no se puedan añadir más terminales a* η;
end

Para obtener una G' GIC sin producciones ϵ se crea un conjunto de producciones P' como: si $B \rightarrow X_1 \cdots X_n$ en P esta se cambia por producciones $B \rightarrow Y_1 \cdots Y_n$ donde

$$Y_i = \begin{cases} X_i & \text{si } X_i \text{ no es anulable} \\ Y_i = X_i \text{ó } Y_i = \epsilon & \text{si } X_i \text{ es anulable} \end{cases}$$

pero no $(\forall i, Y_i = \epsilon)$. Es decir, se añaden producciones donde los anulables X_i aparecen ó no, pero sin incluir a las producciones ϵ.

Ejemplo 74. Sea

$$G: \qquad \begin{aligned} S &\to aA \\ A &\to aA \mid \epsilon \end{aligned}$$

Queremos encontrar G' una GIC tal que $L(G') = L(G)$ pero G' sin producciones ϵ.

Primero calculamos el conjunto de anulables:

inicio: $\eta = \{A\}$

ciclo: fin

Cambiamos ahora las producciones (donde A esté presente ó ausente)

$$G' \qquad \begin{aligned} S &\to a \mid aA \\ A &\to a \mid aA \end{aligned}$$

Comoe $\epsilon \notin L(G)$, entonces $L(G') = L(G)$.

Ejemplo 75.

$$G: \qquad \begin{aligned} S &\to AB \\ A &\to aAA \mid \epsilon \\ B &\to bBB \mid \epsilon \end{aligned}$$

Encontraremos una gramática G' sin producciones ϵ tal que $L(G') = L(G) - \{\epsilon\}$ (nótese que $\epsilon \in L(G)$).

Primero calculamos los anulables:

Inicio: $\eta = \{A, B\}$

ciclo: como $S \to AB$ con $AB \in \eta^*$ entonces $S \in \eta$. Fin.

Resulta $\eta = \{A, B, S\}$ (todos los símbolos no terminales son anulables). Ahora reeemplazamos las producciones de G por ausencia o no de los anulables:

de $S \to AB$ se obtienen

$$S \to A \mid B \mid AB$$

de $A \to aAA$ se obtienen

$$A \to aAA \mid aA \mid a$$

y de $B \to bBB$ se obtienen

$$B \to bBB \mid bB \mid b$$

Resulta entonces

$$G': \quad \begin{aligned} S &\to A \mid B \mid AB \\ A &\to aAA \mid aA \mid a \\ B &\to bBB \mid bB \mid b \end{aligned}$$

tal que $L(G') = L(G) - \{\epsilon\}$.

Tarea 17.

(1) *Obtener la colección de no terminales anulables que pertenecen a la siguiente gramática:*

$$\begin{aligned} S &\to aA|bA|a \\ A &\to aA|bAb|\epsilon \end{aligned}$$

(2) *Obtener, para la siguiente gramática, el número de no terminales anulables:*

$$\begin{aligned} S &\to ABaC \\ A &\to AB \\ B &\to b|\epsilon \\ C &\to D|\epsilon \\ D &\to d \end{aligned}$$

(3) *Eliminar las producciones ϵ de las gramáticas:*

(a)

$$\begin{aligned} S &\to aA|bA|a \\ A &\to aA|bAb|\epsilon \end{aligned}$$

(b)

$$\begin{aligned} S &\to AB \\ A &\to aA|abB|aCa \\ B &\to bA|BB|\epsilon \\ C &\to \epsilon \\ D &\to dB|BCB \end{aligned}$$

(c)

$$\begin{aligned} S &\to a|aA|B \\ A &\to aB|\epsilon \\ B &\to Aa \end{aligned}$$

(4) *El lenguaje asociado con la siguiente GIC contiene ϵ. Eliminar las producciones ϵ exepto $S \to \epsilon$.*

$$\begin{aligned} S &\to AB|aB|\epsilon \\ A &\to BBB|aB|a|\epsilon \\ B &\to a|aA|\epsilon \end{aligned}$$

7. Eliminación de producciones unitarias

Definición 45. *Una* **producción unitaria** *es una del tipo $A \to B$ con A, B no terminales.*

A veces son útiles las producciones unitarias, pero otras veces son indesables pues complican los árboles de derivación. Nos proponemos eliminar tales producciones unitarias.

Definición 46.

(1) *$A \Rightarrow A$ usando cero producciones.*

(2) *Si A es no terminal se define*

$$U(A) = \{B \in N \mid A \overset{*}{\Rightarrow} B \text{ usando solamente producciones unitarias ó cero producciones}\}$$

Nótese que $\forall A$,

$$A \in U(A).$$

El siguiente algoritmo elimina las producciones unitarias:

Ejemplo 76. Sea G la gramática:

$$\begin{aligned} S &\to A \mid Aa \\ A &\to B \\ B &\to C \mid b \\ C &\to D \mid ab \\ D &\to b \end{aligned}$$

A esta le aplicaremos el algoritmo de eliminación de producciones unitarias (EPU):

(2) $P' = \varnothing$;

Algorithm 4: EPU

```
   Data: G = (N, Σ, S, P) una GIC
   Result: G' = (N, Σ, S, P') una GIC sin producciones unitarias tal que
           L(G') = L(G)
1  begin
2      Inicializar P' = ∅;
3      foreach A ∈ N do
4          obtener U(A)
5      end
6      foreach A ∈ N do
7          foreach B ∈ U(A) do
8              para cada producción no unitaria B → w de P añadir A → w
               en P'
9          end
10     end
11 end
```

(3-4)

$$U(S) = \{S, A, B, C, D\}$$
$$U(A) = \{A, B, C, D\}$$
$$U(B) = \{B, C, D\}$$
$$U(C) = \{C, D\}$$
$$U(D) = \{D\}$$

(6) $S \in N$

(7) $S \in U(S)$

(8) P': $S \rightarrow Aa$

(7) $A \in U(S)$

(8) No se añade nada a P', pues no hay producciones no unitarias en P con cabeza A: $P' : S \rightarrow Aa$

(7) $B \in U(S)$

(8) Como $S \rightarrow b$ en P y no es unitaria, se añade a P':

$$P' : \qquad S \rightarrow Aa \,|\, b$$

(7) $C \in U(S)$

(8) $C \rightarrow ab$ en P no unitaria:

$$P' : \qquad S \rightarrow Aa \,|\, b \,|\, ab$$

(7) $D \in U(S)$

(8) $D \to b$ en P no unitaria:

P' : $S \to Aa \,|\, b \,|\, ab$

(6) $A \in N$

(7) $A \in U(A)$

(8) No se añade nada a P' pues no hay producciones no unitarias con cabeza A en P.

(7) $B \in U(A)$

(8) $B \to b$ en P no unitaria:

P' : $S \to Aa \,|\, b \,|\, ab$

$A \to b$

(7) $C \in U(A)$

(8) $C \to ab$ en P no unitaria:

P' : $S \to Aa \,|\, b \,|\, ab$

$A \to b \,|\, ab$

(7) $D \in U(A)$

(8) $D \to b$ en P no unitaria, hay que añadir $A \to b$ a P', pero ésta ya existe en P':

P' : $S \to Aa \,|\, b \,|\, ab$

$A \to b \,|\, ab$

(6) $B \in N$

(7) $B \in U(B)$

(8) $B \to b$ no unitaria en P

P' : $S \to Aa \,|\, b \,|\, ab$

$A \to b \,|\, ab$

$B \to b$

(7) $C \in U(B)$

(8) $C \to ab$ no unitaria en P

P' : $S \to Aa \,|\, b \,|\, ab$

$A \to b \,|\, ab$

$B \to b \,|\, ab$

(7) $D \in U(B)$

(8) $D \to b$ no unitaria en P, hay que añadir $B \to b$ a P', pero ésta ya está en P':

$$P': \quad \begin{aligned} S &\to Aa \,|\, b \,|\, ab \\ A &\to b \,|\, ab \\ B &\to b \,|\, ab \end{aligned}$$

(6) $C \in N$

(7) $C \in U(C)$

(8) $C \to ab$ no unitaria en P

$$P': \quad \begin{aligned} S &\to Aa \,|\, b \,|\, ab \\ A &\to b \,|\, ab \\ B &\to b \,|\, ab \\ C &\to ab \end{aligned}$$

(7) $D \in U(C)$

(8) $D \to b$ no unitaria en P

$$P': \quad \begin{aligned} S &\to Aa \,|\, b \,|\, ab \\ A &\to b \,|\, ab \\ B &\to b \,|\, ab \\ C &\to ab \,|\, b \end{aligned}$$

(6) $D \in N$

(7) $D \in U(D)$

(8) $D \to b$ no unitaria en P

$$P': \quad \begin{aligned} S &\to Aa \,|\, b \,|\, ab \\ A &\to b \,|\, ab \\ B &\to b \,|\, ab \\ C &\to ab, \,|\, b \\ D &\to b \end{aligned}$$

Salida:

$$G': \quad \begin{aligned} S &\to Aa \,|\, b \,|\, ab \\ A &\to b \,|\, ab \\ B &\to b \,|\, ab \\ C &\to ab \,|\, b \\ D &\to b \end{aligned}$$

Ejemplo 77. Sea

$$G: \quad \begin{aligned} I &\to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1 \\ F &\to I\,|\,(E) \\ T &\to F\,|\,T*F \\ E &\to T\,|\,E+T \end{aligned}$$

Tal G tiene producciones unitarias; eliminémoslas:

(1) $P' = \varnothing$

(2) $U(I) = \{I\}$, $U(F) = \{F, I\}$, $U(T) = \{T, F, I\}$, $U(E) = \{E, T, F, I\}$

(3) $I \in N$:
   (a) $I \in U(I)$; $I \Rightarrow I$; $I \to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1$, añadir $I \to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1$

(4) $F \in N$:
   (a) $F \in U(F)$; $F \Rightarrow F$, $F \to (E)$, añadir $F \to (E)$
   (b) $I \in U(F)$; $F \Rightarrow I$; $I \to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1$; añadir $F \to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1$

(5) $T \in N$:
   (a) $T \in U(T)$; $T \Rightarrow T$, $T \to T*F$, añadir $T \to T*F$
   (b) $F \in U(T)$; $T \Rightarrow F$, $F \to (E)$, añadir $F \to (E)$
   (c) $I \in U(T)$; $T \Rightarrow I$, $I \to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1$, añadir $T \to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1$

(6) $E \in N$;:
   (a) $E \in U(E)$; $E \Rightarrow E$, $E \to E+T$, añadir $E \to E+T$
   (b) $T \in U(E)$; $E \Rightarrow T$, $T \to T*F$, añadir $E \to T*F$
   (c) $F \in U(E)$; $E \Rightarrow F$, $F \to (E)$, añadir $E \to (E)$
   (d) $I \in U(E)$; $E \Rightarrow I$, $I \to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1$, añadir $E \to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1$

Salida:

$$G': \quad \begin{aligned} I &\to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1 \\ F &\to a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1\,|\,(E) \\ T &\to T*F\,|\,a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1\,|\,(E) \\ E &\to \,|\,E+T\,|\,T*F\,|\,a\,|\,b\,|\,Ia\,|\,Ib\,|\,I0\,|\,I1 \end{aligned}$$

Tarea 18.

(1) *Diseñar un algoritmo para construir* $U(A)$*, siendo* A *un no terminal de una GIC.*

(2) *Eliminar todas las producciones unitarias de las siguientes GIC:*

(a)

$$
\begin{aligned}
S &\to CBa|D \\
A &\to bbC \\
B &\to Sc|ddd \\
C &\to eA|f|C \\
D &\to E|SABC \\
E &\to gh
\end{aligned}
$$

(b)

$$
\begin{aligned}
S &\to Aa|Ba|B \\
A &\to Aa|\epsilon \\
B &\to aA|BB|\epsilon
\end{aligned}
$$

Obsérvese que $\epsilon \in L(G)$.

8. Formas normales

Definición 47.

(1) *Una GIC está en* **forma normal de Chomsky** *(FNC) si todas sus producciones son de la forma $A \to BC$ ó $A \to a$ donde $A, B, C \in N$ y $a \in \Sigma$.*

(2) *Una GIC está en* **forma normal de Greibach** *(FNG) si todas sus producciones son de la forma*

$$A \to aB_1 \dots B_k$$

para algún $k \geq 0$, donde $A, B_1, \dots, B_k \in N$ y $a \in \Sigma$ (si $k = 0$ se obtiene la producción $A \to a$).

Ejemplo 78. Sea $\Sigma = \{[,]\}$, entonces la gramática

$$
\begin{aligned}
S &\to AB \,|\, AC \,|S \\
C &\to SB \\
A &\to [\\
B &\to]
\end{aligned}
$$

es una gramática en FNC.

Se puede probar que ésta gramática es, en cierto sentido (teorema de Chomsky-Schuatzenberger), el prototipo de las gramáticas independientes de contexto.

Ejemplo 79. Sea, de nuevo $\Sigma = \{[,]\}$ y gramática

$$S \to [B \,|\, [SB \,|\, [BS \,|\, [SBS$$
$$\to]$$

que es una gramática en FNG.

Las gramáticas de los dos ejemplos inmediatos anteriores generan el lenguaje de paréntesis balanceados.

Nótese que ninguna gramática en FNC, ni en FNG puede generar la palabra vacía.

Se quiere probar que:

Teorema 18. *Sea G una GIC. Entonces*

(1) *Existe G' una GIC en FNC tal que $L(G') = L(G) - \{\epsilon\}$.*
(2) *Existe G' una GIC en FNG tal que $L(G') = L(G) - \{\epsilon\}$.*

8.1. Forma normal de Chomsky. Dada G una GIC, su forma normal de Chomsky puede obtenerse de la siguiente forma:

La salida del algoritmo FNC da una gramática tal que todas sus producciones son de la forma $A \to BC$ ó $A \to \sigma$ con σ no terminal.

Ejemplo 80. *Sea*

$$\begin{aligned} G: \qquad & S \to bA \,|\, aB \\ & A \to bAA \,|\, aS \,|\, a \\ & B \to aBB \,|\, bS \,|\, b \end{aligned}$$

Poner ésta gramática en FNC.

Sol.

(1) Primero quitamos los símbolos inútiles: obtenemos la misma G, pues no tiene símbolos inútiles.
(2) Quitamos las producciones ϵ: queda, de nuevo, la misma G, pues G no tiene producciones ϵ.
(3) Quitamos las producciones unitarias: queda la misma G pues G no tiene producciones unitarias.
(4) Ponemos producciones con cuerpos formados de sólo cadenas de variables ó sólo un terminal:

$$\begin{array}{ll} S \to C_aA \,|\, C_aB & A \to C_bAA \,|\, C_aS \,| b \\ C_b \to b & B \to C_aBB \,|\, C_bS \,|\, b \\ C_a \to a & \end{array}$$

Algorithm 5: FNC

Data: G una GIC
Result: G' una GIC en FNC tal que $L(G') = L(G) - \{\epsilon\}$
begin
  $G_1 \leftarrow$ Algoritmo 3.5.1(G) ;
  $G_2 \leftarrow$Algorithmo 3.5.2(G_1);
  $G_3 \leftarrow$ Eliminar producciones ϵ de G_2;
  $G_4 \leftarrow$ Eliminar producciones unitarias de G_3;
  Poner $P_5 = \varnothing$;
  foreach $A \rightarrow w$ *producción en* G_4 **do**
    if $|w| = 1$ **then**
      Añadir $A \rightarrow w$ a P_5
    end
    if $|w| > 1$ **then**
      $w = X_1 \dots X_n$ con cada X_i terminal ó no;
      if X_i *terminal* **then**
        Añadir un nuevo símbolo no terminal C_{X_i} y producción $C_{X_i} \rightarrow X_i$ a P_5
      end
    end
  end
  Las producciones de P_5 son de la forma $A \rightarrow B_1 \dots B_n$ ó $A \rightarrow \sigma$ con $B_1, \dots, B_n$ terminales y σ terminal;
  foreach $A \rightarrow B_1 \dots B_n$ *con* $n \geq 2$ **do**
    reemplazar por

$$\begin{aligned} A &\rightarrow B_1 D_1 \\ D_1 &\rightarrow B_2 D_2 \\ &\vdots \\ D_{n-1} &\rightarrow B_n D_n \end{aligned}$$

  end
  Salida: la gramática G' con producciones P_5
end

(5) Ponemos producciones binarias para las cadenas de no terminales: la producción $A \rightarrow C_b AA$ se reemplaza por

$$\begin{aligned} A &\rightarrow C_b D_1 \\ D_1 &\rightarrow AA \end{aligned}$$

(nótese que

$$\begin{aligned} A &\Rightarrow C_b D_1 \\ &\Rightarrow C_b AA \end{aligned}$$

) y $B \to C_a BB$ se reemplaza por

$$\begin{aligned} B &\to C_a D_2 \\ D_2 &\to BB \end{aligned}$$

(nótese que

$$\begin{aligned} A &\Rightarrow C_a D_1 \\ &C_a BB \end{aligned}$$

). La FNC pedida es

$$\begin{aligned} S &\to C_b A \,|\, C_a B \\ A &\to C_b D_1 \,|\, C_a S \,|\, a \\ B &\to C_a D_2 \,|\, C_b S \,|\, b \\ D_2 &\to BB \\ D_1 &\to AA \\ C_b &\to b \\ C_a &\to a \end{aligned}$$

□

Recordemos que los algoritmos 3.5.1 + algoritmo 3.5.2 quitan los inútiles.

Ejemplo 81. Sea G la GIC:

$$\begin{aligned} S &\to AACD \\ A &\to aAb \,|\, \epsilon \\ C &\to aC \,|\, a \\ D &\to aDa \,|\, bDb \,|\, \epsilon. \end{aligned}$$

Calcular la FNC de G.

Tarea 19.

(1) *Convertir las siguientes GIC a forma normal de Chomsky:*
(a)

$$\begin{aligned} S &\to AB|CA \\ A &\to a \\ B &\to BC|AB \\ C &\to aB|b \end{aligned}$$

(b)

$$
\begin{aligned}
S &\to aAb|cHB|CH\\
A &\to dBH|eeC\\
B &\to ff|D\\
C &\to gFB|ah\\
D &\to i\\
E &\to jF\\
F &\to dcGGG|cF\\
G &\to kF\\
H &\to Hlm
\end{aligned}
$$

9. Autómatas de Pila

Las grámaticas independientes del contexto corresponden exactamente a cierta clase de auómatas que tienen memoria (pila): *autómatas finitos de pila*.

Definición 48. *Un* **autómata finito de pila no determinista** *(APND) es una 7-upla*

$$M = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$$

donde

- Q *es un conjunto finito de estados;*
- Σ *es un conjunto finito de símbolos de un alfabeto de entrada;*
- Γ *es un conjunto finito de símbolos de un alfabeto de pila;*
- $\Delta \subseteq \big(Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma\big) \times \big(Q \times \Gamma^*\big)$ *relación de transición;*
- $s \in Q$ *estado inicial;*
- $\perp \in \Gamma$ *símbolo inicial de pila;*
- $F \subseteq Q$ *estados finales o de aceptación.*

Estos autómatas se interpretan como mecanismos que tienen un estado actual y además una memoria y su función es leer una cinta que contiene símbolos. Más especificamente: nótese que los elementos de la relación de transición Δ son de la forma

$$\big((p, a, A), (q, B_1 \dots B_k)\big) \in \Delta$$

que se interpreta como que la máquina está en el estado p leyendo el símbolo a sobre una cinta de entrada y A está encima de la pila (memoria):

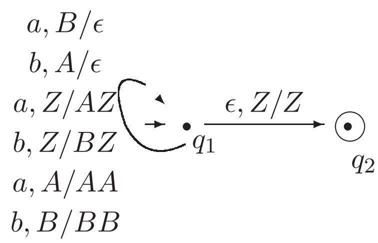
Tarea 20.

(1) *Probar que cada uno de los siguientes lenguajes no son lenguajes independientes del contexto.*

(a) $\{a^i b^i c^i \mid i \geq 1\}$
(b) $\{a^i b^i c^j \mid j \geq i\}$
(c) $\{a^i b^j c^j \mid i \leq j \leq k\}$
(d) $\{a^i \mid i \text{ es primo}\}$
(e) $\{w \in \{a, b, c\}^* \mid n \leq m \leq 2n\}$
(f) $\{ww \mid w \in \{a, b\}^*\}$

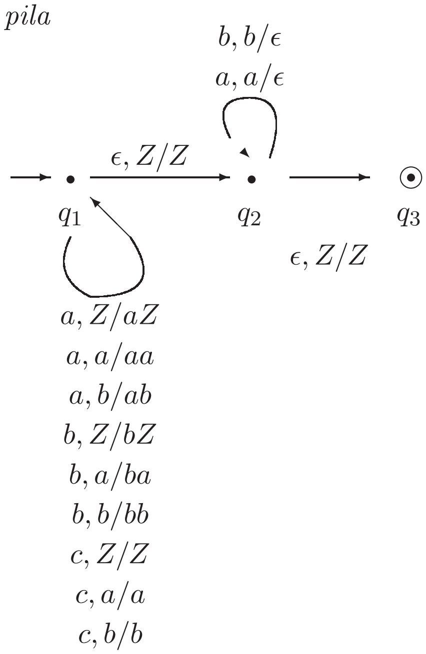
(2) *Obtener un autómata a pila que acepte* $\{a^n b^n \mid n \geq 0\}$.

(3) *Describir los cambios de configuraciones sobre las cadenas abaababb y abaa realizados por el autómata a pila siguiente (con símbolo inicial de pila* Z*):*



¿son aceptadas?.

(4) *Los mismo que el anterior para las cadenas* c, $abcba$, $abcab$ *y* $babbcbbab$ *y el autómata a pila*



Máquinas de Turing

1. Definición y termininología

Definición 49. *Una* **máquina de Turing** *es una 7-tupla:*

$$M = (Q, \Sigma, \Gamma, s, B, F, \delta)$$

donde

- Q *es un conjunto finito de estados.*
- Σ *es un alfabeto de entrada.*
- Γ *es un alfabeto llamado* **de entrada**.
- $s \in Q$ *estado inicial.*
- Γ *es alfabeto llamado* **de cinta**, *con* $\Gamma \supseteq \Sigma$.
- $B \in \Gamma$ *pero* $B \notin \Sigma$ *llamado* **símbolo blanco**.
- $F \subseteq Q$ *estados finales.* $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ *es una función parcial llamada función de transición*

Aclaramos que, en general, una función parcial $g : A \to B$ es una relación de A en B tal que $\forall a \in A, \quad g(a)$ es a lo más un elemento.

Ejemplo 82. Sea $f : \mathbb{R} \to \mathbb{R}$, $f(x) = 1/x$ es función parcial. Nótese que $f(0)$ no está definida.

Ejemplo 83. *Sea* $g : (-1, 1)$, $g(x) = \log(x)$ *también es una función parcial pues* g *no está definida en* $(-1, 0]$.

Si M es una MT con funciónde transición δ, entonces

$$\delta(q, \gamma) = (p, \tau, X)$$

donde

(1) q es **estado actual**, γ es **símbolo de cinta leído**

(2) p es **estado siguiente**, τ es **símbolo escrito en cinta**.

(3) X es un **movimiento de escritura/lectura de cabeza** que puede ser L (izquierda) ó R (derecha).

Toda ésta teminología se debe a que las MT se interpretan como mecanismos que constan de una cinta infinita hacia la izquierda y hacia la derecha, separada en celdas que contienen a los símbolos de cinta:

$\cdots$	B	B	B	a	b	b	c	B	B	$\cdots$

Tal cinta está casi llena de símbolos blancos. Sólo una porción finita entre símbolos blancos contiene la información que interesa.

Las MT cambian sus configuraciones celda por celda, senalando la celda a modificar con una *cabeza de cinta*. Por ejemplo;

$$\delta(q_1, a) = (q_5, b, R)$$

indica que si

a	b	b

estado q_0

entonces

b	b	b

estado q_1

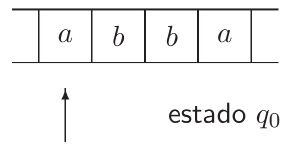
Ejemplo 84. Sea M la MT definida por

- $Q = \{q_0, q_1\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, b, B\}$
- $F = \{q_1\}$
- $s = q_0$

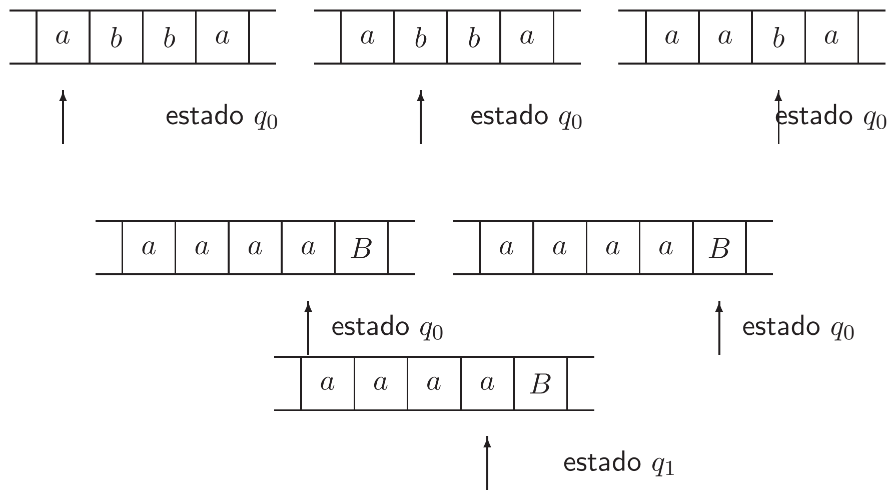
y transiciones dadas por

δ	a	b	B
q_0	(q_0, a, R)	(q_0, a, R)	(q_1, R, L)
q_1			

Inciamos con configuración



y entonces tenemos los siguientes cambios de configuración



Nótese que se ha insertado B que se sobreentiende siempre aparece a la izquierda y a la derecha de la palabra escrita en cinta.

Tales cambios de confuguración se pueden denotar más brevemente si se usa la notación torniquete; hay al menos dos variantes

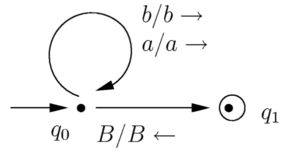
(1)

$$\begin{aligned}(q_0, \underline{a}bba) &\vdash (q_0, a\underline{b}ba)\\ &\vdash (q_0, aa\underline{b}a)\\ &\vdash (q_0, aaa\underline{a})\\ &\vdash (q_0, aaaa\underline{B})\\ &\vdash (q_1, aaa\underline{a})\end{aligned}$$

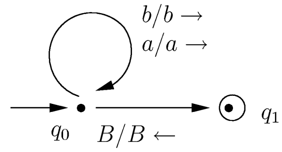
(2)

$$\begin{aligned}q_0abba &\vdash aq_1bba\\ &\vdash aaq_1ba\\ &\vdash aaaq_1a\\ &\vdash aaaaq_1B\\ &\vdash aaaq_2a\end{aligned}$$

Las MT tambíen tienen diagramas de transición similares a los autómatas de pila. Por ejemplo, el diagrama de transición de la MT del ejemplo inmediato anterior es



Ejemplo 85. Sea M la máquina de Turing definida por el diagrama

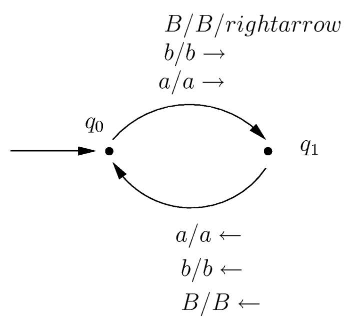


entonces

$$\begin{aligned} aabq_1abb &\vdash aaq_1babb \\ &\vdash aq_1ababb \\ &\vdash q_1aababb \\ &\vdash q_1Baababb \\ &\vdash Bq_2aababb = q_2aababb \\ &\vdash q_3Baababb \end{aligned}$$

y entonces la máquina se detiene.

Ejemplo 86. Consideremos la máquina de Turing



aquí el conjunto de estado finales es $F = \varnothing$ y el alfabeto de entrada es $\Sigma = \{a, b\}$.

Para cualquier $w \in \Sigma^*$ tenemos que

$$\begin{aligned} q_1abw &\vdash aq_2w \\ &\vdash q_1aw \\ &\vdash aq_2bw \\ &\vdash q_1abw \\ &\vdash \dots \end{aligned}$$

y la MT nunca para (bucle infinito). Se pone

$$q_1abw \overset{*}{\vdash} \infty$$

para indicar que comenzando en la configuración q_1abw la máquina nunca para.

2. Aceptación

Definición 50. *Sea* $M = (Q, \Sigma, \Gamma, s, B, F, \delta)$ *una MT. El* **lenguaje aceptado** *por* M *es*

$$L(M) = \{w \in \Sigma^* \mid sw \vdash w_1pw_2,\ \textit{con}\ w_1, w_2 \in \Gamma^*, p \in F\}$$

Tarea 21.

(1) *Construir una máquina de Turing que analice una cadena de* $\{a, b\}^+$ *desplazándose por la cinta de izquierda a derecha y que reeemplace cada b's por c. La máquina de Turing debería comenzar con la cabeza sobre el primer símbolo (el que está más a la izquierda) de la cadena y terminar su cabeza sobre el blanco final (el blanco que sigue a la a o a la c que esté más a la derecha en la cadena transformada).*

(2) *Construir una máquina de Turing que enumere todos los enteros binarios, en orden numérico sobre su cinta cuando comience con* $(q_1, 0B)$. *Es decir, la máquina de Turing debe ejecutarse de esta forma:*

$$(q_1, \underline{0}B) \overset{*}{\vdash} (q_1, \underline{1}B) \overset{*}{\vdash} (q_1, 1\underline{0}B) \overset{*}{\vdash} (q_1, 1\underline{1}B) \overset{*}{\vdash} \cdots$$