AUTÓMATAS Y LENGUAJES FORMALES PRELIMINARES MATEMÁTICOS LENGUAJES REGULARES I

Francisco Hernández Quiroz

Departamento de Matemáticas
Facultad de Ciencias, UNAM
E–mail: fhq@ciencias.unam.mx
Página Web: www.matematicas.unam.mx/fhq

Facultad de Ciencias

Inducción en definiciones y demostraciones

En matemáticas es común que un conjunto se defina *inductivamente*, es decir, a partir de un conjunto inicial y de reglas para crear nuevos elementos. Por ejemplo, las listas de naturales:

- [ ] es la lista vacía
- si $[a_1, \ldots, a_n]$ es una lista de $\mathbb{N}$ y $m \in \mathbb{N}$ entonces $[m, a_1, \ldots, a_n]$ es una lista de naturales.

Las definiciones inductivas vienen acompañadas de *demostraciones inductivas*, es decir, de técnicas de demostración que se basan en la estructura inductiva de un conjunto.

Números naturales

Los números naturales se pueden definir inductivamente:

- 0 es un número natural
- si n es un número natural, entonces el sucesor de n también es un número natural

El sucesor de un número k suele denotarse como $s(k)$, $suc(k)$ o simplemente como $k + 1$.

Inducción matemática

Si se quiere demostrar una propiedad P de los números naturales entonces se puede hacer utilizando su estructura inductiva:

1. Se demuestra que P vale para 0.
2. Se formula una hipótesis inductiva: “P vale para cierto número $m \in \mathbb{N}$”.
3. Se demuestra que P vale para $m + 1$.

Ejemplo de inducción matemática

Queremos demostrar que $\sum_{i=0}^{n} i = \frac{n^2+n}{2}$. Entonces

1. Demostramos que $\sum_{i=0}^{0} i = \frac{0^2+0}{2}$ (obvio).
2. Formulamos la hipótesis $\sum_{i=0}^{m} i = \frac{m^2+m}{2}$.
3. Demostramos que $\sum_{i=0}^{m+1} i = \frac{(m+1)^2+m+1}{2}$.

Conjuntos bien ordenados

- Los conjuntos definidos inductivamente se pueden ordenar de acuerdo con el orden estructural

  $a <_S b$ sii b se construye a partir de a.

  En el caso de $\mathbb{N}$, $m <_S n$ sii $n = m + 1$.
- En muchos casos, $<_S$ ordena un conjunto de tal forma que
  - no existen cadenas infinitas descendentes $\ldots <_S a_{n+1} <_S a_n <_S \ldots <_S a_1$
  - para todo elemento, existe un número finito de predecesores en $<_S$.

Inducción bien fundada

Cuando hay un conjunto inductivo C bien ordenado por $<_S$, es posible demostrar una propiedad P de elementos de C inductivamente:

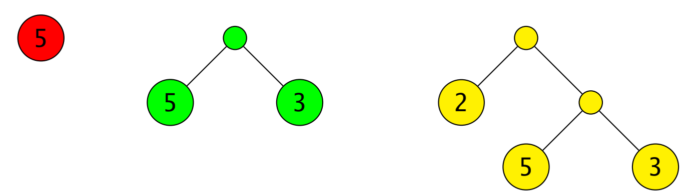
1. Se demuestra que P vale de todos los elementos minimales de C.
2. Se formula una hipótesis inductiva: “Sea $a \in C$. P vale para todos los $b \in C$ tales que $b <_S a$”.
3. Se demuestra que P también vale para a.

Ejemplo de inducción bien fundada I

Definimos los árboles binarios de números naturales:

- Todos los números naturales son árboles binarios (los elementos básicos del conjunto)
- Si a y b son árboles binarios, entonces $(a \bullet b)$ es un árbol binario (caso inductivo)

Los árboles binarios también se pueden representar gráficamente:



Si $a = (b \bullet c)$, entonces $b <_S a$ y $c <_S a$.

Ejemplo de inducción bien fundada II

Considérense ahora las siguientes funciones:

$$\begin{array}{rcll} \mathrm{Max}(n) & = & n & \text{si } n \in \mathbb{N} \\ \mathrm{Max}(a \bullet b) & = & \text{máximo}\{\mathrm{Max}(a), \mathrm{Max}(b)\} & \\ \mathrm{Card}(n) & = & 1 & \text{si } n \in \mathbb{N} \\ \mathrm{Card}(a \bullet b) & = & \mathrm{Card}(a) + \mathrm{Card}(b) & \\ \mathrm{Suma}(n) & = & n & \text{si } n \in \mathbb{N} \\ \mathrm{Suma}(a \bullet b) & = & \mathrm{Suma}(a) + \mathrm{Suma}(b) & \end{array}$$

Demostraremos que para cualquier árbol binario de naturales a

$$\mathrm{Suma}(a) \leq \mathrm{Max}(a) \times \mathrm{Card}(a).$$

Ejemplo de inducción bien fundada III

1. Caso básico. Los minimales en este caso son los números naturales. Sea $n \in \mathbb{N}$. Entonces

$$\mathrm{Suma}(n) = n = \mathrm{Max}(n) = \mathrm{Max}(n) \times 1 = \mathrm{Max}(n) \times \mathrm{Card}(n)$$

2. Hipótesis inductiva. Sea $a = (b \bullet c)$. Entonces

$$\mathrm{Suma}(b) \leq \mathrm{Max}(b) \times \mathrm{Card}(b).$$

$$\mathrm{Suma}(c) \leq \mathrm{Max}(c) \times \mathrm{Card}(c).$$

pues $b <_S a$ y $c <_S a$.

3. Debemos demostrar que

$$\mathrm{Suma}(a) \leq \mathrm{Max}(a) \times \mathrm{Card}(a),$$

lo que se sigue de la hipótesis inductiva y las definiciones de Max, Card y Suma.

Relaciones y cerradura transitiva y reflexiva

Definición 1.1

Sea A un conjunto. Una relación R en A es un subconjunto de $A \times A$. La cerradura transitiva y reflexiva de R se define inductivamente:

$$\begin{array}{rcl} R^0 & = & \{(a, a) \mid a \in A\} \\ R^1 & = & R \\ R^{n+1} & = & R^n \cup \{(a, b) \mid \exists c \text{ tal que } (a, c) \text{ y } (c, b) \in R^n\} \\ R^* & = & \bigcup_{n \in \mathbb{N}} R^n \\ R^+ & = & \bigcup_{n \in \mathbb{N}} R^{n+1} \end{array}$$

R^+ es la cerradura transitiva y R^* es la cerradura transitiva y reflexiva de R.

Alfabeto

- Un alfabeto Σ es un conjunto finito de símbolos. Ejemplo: $\{a, b, \ldots z\}$.
- Una cadena (finita) es una sucesión de símbolos de Σ. Ejemplo $acczi$.
- La “cadena” vacía no contiene símbolos. Se suele denotar con ε o λ.
- El conjunto de cadenas del alfabeto Σ se denota con Σ^*.
- Si se excluye a ε, entonces se llama Σ^+.

Concatenación

- Las cadenas se pueden concatenar. Por ejemplo, si $\alpha = aabc$ y $\beta = ccaz$, entonces
$$\alpha \cdot \beta = aabcccaz.$$
- Es muy común omitir el signo de concatenación $\cdot$
- ε es el elemento neutro de la operación de concatenación, pues
$$\alpha\varepsilon = \varepsilon\alpha = \alpha.$$

Lenguajes

Si Σ es un alfabeto, entonces un lenguaje L es un subconjunto de Σ^*.

Nota

- Dado que Σ es finito, Σ^* es un conjunto infinito del mismo tamaño que $\mathbb{N}$.
- El conjunto de subconjuntos de Σ^* se denota como $\mathscr{P}(\Sigma^*)$.
- Por la definición anterior, el conjunto de lenguajes en Σ^* es $\mathscr{P}(\Sigma^*)$.
- $\mathscr{P}(\Sigma^*)$ es infinito, de un tamaño mayor que el de $\mathbb{N}$.

Operaciones en lenguajes

Sean A y B dos lenguajes en Σ. Tenemos las siguientes operaciones:

- Unión $A \cup B$.
- Intersección $A \cap B$.
- Diferencia $A - B = \{\alpha \mid \alpha \in A \text{ y } \alpha \notin B\}$.
- Complemento $\sim A = \Sigma^* - A$.
- Concatenación $AB = \{\alpha\beta \mid \alpha \in A \text{ y } \beta \in B\}$.
- Estrella de Kleene $A^* = \bigcup_{n \in \mathbb{N}} A^n$ donde
$$\begin{aligned} A^0 &= \{\varepsilon\} \\ A^{n+1} &= A^n A \end{aligned}$$

Gramáticas

Definición 1.2

Una gramática G es una cuarteta $\langle \Sigma, \Gamma, S, \to_G \rangle$:

- *Σ es un alfabeto de símbolos terminales*
- *Γ es un alfabeto de símbolos no terminales*
- *$S \in \Gamma$ es el símbolo inicial*
- *$\to_G \subseteq (\Sigma \cup \Gamma)^* \Gamma^+ (\Sigma \cup \Gamma)^* \times (\Sigma \cup \Gamma)^*$ son las reglas de producción*

A partir de G definimos la relación $\Rightarrow_G \subseteq (\Sigma \cup \Gamma)^* \Gamma^+ (\Sigma \cup \Gamma)^* \times (\Sigma \cup \Gamma)^*$.

$\alpha \Rightarrow_G \beta$ sii existen $\gamma, \kappa, \theta, \lambda$ tales que

- $\alpha = \gamma\kappa\theta$
- $\beta = \gamma\lambda\theta$
- $\kappa \rightarrow_G \lambda$

Lenguaje generado por una gramática

Definición 1.3

Sea G una gramática. El lenguaje generado por G es

$$L_G = \{\alpha \in \Sigma^* \mid S \Rightarrow_G^* \alpha\}.$$

Ejemplo. Sea G la siguiente gramática

- $\Sigma = \{a, b\}$
- $\Gamma = \{S\}$
- $\{S \rightarrow_G aSb, S \rightarrow_G \varepsilon\}$

El lenguaje generado por la gramática anterior son todas las cadenas formadas por cierto número de a seguidas del mismo número de b

Jerarquía de Chomsky

Se pueden clasificar las gramáticas y los lenguajes que generan de acuerdo con ciertas restricciones:

- *Tipo 3*: Si todas las reglas de producción tienen la forma $A \rightarrow \alpha B$ o $A \rightarrow \alpha$, donde $A, B \in \Gamma$ y $\alpha \in \Sigma^*$
- *Tipo 2*: Si todas las reglas tienen la forma $A \rightarrow \alpha$, donde $A \in \Gamma$ y $\alpha \in (\Sigma \cup \Gamma)^*$
- *Tipo 1*: Si no hay una regla de producción con la forma $\alpha \rightarrow \varepsilon$
- *Tipo 0*: Sin restricciones.

Esta clasificación se conoce como la *jerarquía de Chomsky*

Gramáticas y el curso

Estos tipos también se conocen con otros nombres:

- Tipo 3: lenguajes regulares
- Tipo 2: lenguajes independientes del contexto (o libres del contexto)
- Tipo 1: lenguajes dependientes del contexto (o sensibles del contexto)
- Tipo 0: lenguajes recursivamente enumerables

Estudiaremos los tipos 3, 2 y 0 en las primeras tres partes del curso.

Autómatas finitos

Definición 1.4

Un autómata finito determinista (DFA) *está formado por*

- *Un alfabeto de entrada* Σ
- *Un conjunto finito de estados* Q
- *Un estado inicial* $s \in Q$ *y un conjunto de estados finales* $F \subseteq Q$
- *Una función de transición* $\delta : Q \times \Sigma \to Q$

Ejemplo 1

El autómata formado por $\Sigma = \{a, b, c, d\}$, $Q = \{s, 1, 2, 3, f\}$ y δ descrita por la siguiente tabla:

Q	a	b	c	d
s	—	1	—	—
1	2	—	—	—
2	3	—	2	2
3	—	f	—	—
f	—	—	—	—

Y que acepta el lenguaje formado por cadenas que empiezan con ba, terminan con ab y enmedio tienen un número indeterminado de c y d.

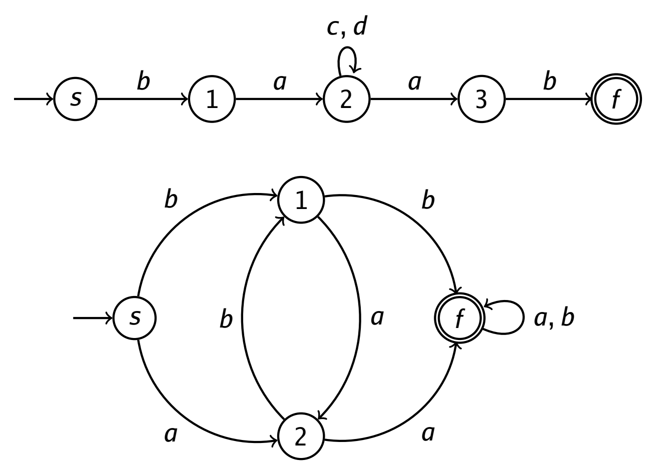
Ejemplo 2

El autómata formado por $\Sigma = \{a, b\}$, $Q = \{s, 1, 2, f\}$ y δ descrita por la siguiente tabla:

Q	a	b
s	2	1
1	2	f
2	f	1
f	f	f

Y que acepta un lenguaje cuya descripción es demasiado larga como para ser comprensible.

Representación gráfica



Lenguajes regulares

Podemos ampliar la función δ a una función $\delta^* : Q \times \Sigma^* \to Q$ de la siguiente forma. Sean $q \in Q$, $a \in \Sigma$ y $\alpha \in \Sigma^*$:

$$\begin{aligned} \delta^*(q, \varepsilon) &= q \\ \delta^*(q, a\alpha) &= \delta^*(\delta(q, a), \alpha) \end{aligned}$$

O alternativamente

$$\begin{aligned} \delta^*(q, \varepsilon) &= q \\ \delta^*(q, \alpha a) &= \delta(\delta^*(q, \alpha), a) \end{aligned}$$

Nota: el alumno puede demostrar que ambas definiciones son equivalentes.

Ahora podemos definir el lenguaje aceptado por el autómata $A = (Q, \Sigma, s, F, \delta)$:

$$L(A) = \{\alpha \mid \delta^*(s, \alpha) \in F\}.$$

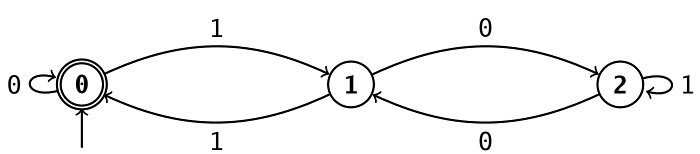
Definición 1.5

Un lenguaje $L \subseteq \Sigma^*$ es regular sii exite un autómata finito A tal que $L = L(A)$.

Un ejemplo aritmético

El siguiente autómata acepta el lenguaje

$$\{\alpha \in \{0, 1\}^* \mid \alpha \text{ es un múltiplo de 3 en binario}\}$$



núm. representado por el prefijo leído	estado del autómata
0 **mód** 3	**0**
1 **mód** 3	**1**
2 **mód** 3	**2**

Sea $\#\alpha$ el número denotado por la cadena binaria α. Entonces

$$\delta^*(\mathbf{0}, \alpha) = \#\alpha \ \mathbf{mód}\ 3$$

Demostración. Primero

$$\begin{aligned} \#(\alpha d) &= 2(\#\alpha) + d \qquad d \in \{0, 1\} \\ \delta(q, d) &= (2q + d)\ \mathbf{mód}\ 3 \end{aligned}$$

Ahora, por inducción en α. Caso básico

$$\begin{aligned} \delta^*(\mathbf{0}, \varepsilon) &= \mathbf{0} && \text{definición de } \delta^* \\ &= \#\varepsilon && \text{pues } \#\varepsilon = 0 \\ &= \#\varepsilon\ \mathbf{mód}\ 3 \end{aligned}$$

Hipótesis inductiva: $\delta^*(\mathbf{0}, \alpha) = \#\alpha \textbf{ mód } 3$. Por demostrar

$$\begin{aligned}
\delta^*(\mathbf{0}, \alpha d) &= \delta(\delta^*(\mathbf{0}, \alpha), d) \\
&= \delta(\#\alpha \textbf{ mód } 3, d) \\
&= (2(\#\alpha \textbf{ mód } 3) + d) \textbf{ mód } 3 \\
&= (2(\#\alpha) + d) \textbf{ mód } 3 \\
&= \#\alpha d \textbf{ mód } 3
\end{aligned}$$

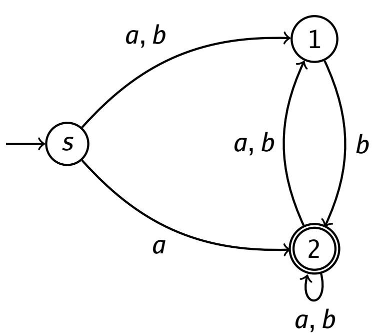
Autómatas no deterministas

- Un autómata finito no determinista (NFA) puede *elegir* uno de varios estados posibles cuando lee un carácter.
- Además, tiene un conjunto de estados iniciales S.
- La función de transición de un NFA es $\Delta : Q \times \Sigma \to \mathscr{P}(Q)$.
- La función anterior se extiende a una función $\Delta^* : \mathscr{P}(Q) \times \Sigma^* \to \mathscr{P}(Q)$ de esta forma:

$$\begin{aligned}
\Delta^*(A, \varepsilon) &= A \\
\Delta^*(A, \alpha a) &= \bigcup_{q \in \Delta^*(A, \alpha)} \Delta(q, a)
\end{aligned}$$

- El lenguaje aceptado es $\{\alpha \mid \Delta^*(S, \alpha) \cap F \neq \varnothing\}$

Ejemplo



Equivalencia con los DFA

Sea $N = (Q, \Sigma, \Delta, S, F)$ un NFA y sea $D = (Q_D, \Sigma, \delta_D, s_D, F_D)$ el siguiente autómata DFA

$$\begin{aligned}
Q_D &= \mathscr{P}(Q) \\
\delta_D(A, a) &= \Delta^*(A, a) \\
s_D &= S \\
F_D &= \{A \subseteq Q \mid A \cap F \neq \varnothing\}
\end{aligned}$$

Ambos autómatas aceptan el mismo lenguaje.

Demostración. Los tres lemas siguientes

$$\begin{array}{rcl} \Delta^*(A, \alpha\beta) & = & \Delta^*(\Delta^*(A, \alpha), \beta) \\ \Delta^*(\bigcup_i A_i, \alpha) & = & \bigcup_i \Delta^*(A_i, \alpha) \\ \delta_D^*(A, \alpha) & = & \Delta^*(A, \alpha) \end{array}$$

se demuestran por inducción sobre β, α y α, en ese orden. Entonces, $L(N) = L(D)$:

$$\begin{array}{rll} \alpha \in L(D) & \text{sii} & \delta^*(s_D, \alpha) \in F_D \\ & \text{sii} & \Delta^*(S_N, \alpha) \cap F_N \neq \varnothing \\ & \text{sii} & \alpha \in L(N) \end{array}$$

Transiciones ε

Definición 1.6

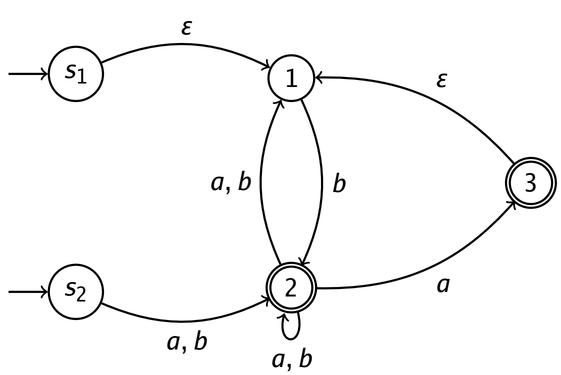
Un autómata finito no determinista con transiciones–ε es una quinteta $N = (Q, \Sigma, \Delta, S, F)$, igual a un NFA *salvo que*

$$\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathscr{P}(Q).$$

Ahora es posible transitar de un estado a otro sin *consumir* símbolos. Δ^* se define así

$$\begin{array}{rcl} \Delta^*(A, \varepsilon) & = & A \cup \bigcup_{q \in A} \Delta(q, \varepsilon) \\ \Delta^*(A, \alpha a) & = & \bigcup_{q \in \Delta^*(A,\alpha)} \Delta(q, a) \quad \cup \bigcup_{\substack{r \in \Delta(q,\varepsilon) \\ q \in \Delta^*(A,\alpha)}} \Delta(r, a) \end{array}$$

Ejemplo



NFA–ε es equivalente a NFA

Sean $N = (Q, \Sigma, \Delta, S, F) \in$ NFA–ε y $q \in Q$. Definimos la cerradura–ε de q

$$\begin{array}{rcl} \Delta_\varepsilon^0(q) & = & \{q\} \\ \Delta_\varepsilon^{n+1}(q) & = & \bigcup_{r \in \Delta_\varepsilon^n(q)} \Delta(r, \varepsilon) \\ \Delta_\varepsilon^*(q) & = & \bigcup_{n \in \mathbb{N}} \Delta_\varepsilon^n(q) \end{array}$$

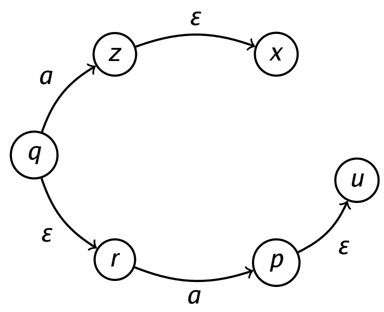
Sea $N' = (Q, \Sigma, \Delta', S, F')$, con

$$\begin{array}{rcl} \Delta'(q, a) & = & \Delta(q, a) \cup \bigcup_{r \in \Delta(q,a)} \Delta_\varepsilon^*(r) \cup \bigcup_{r \in \Delta_\varepsilon^*(q)} \Delta(r, a) \cup \bigcup_{p \in \cup_{r \in \Delta_\varepsilon^*(q)} \Delta(r,a)} \Delta_\varepsilon^*(p) \\ F' & = & \{q \mid \Delta_\varepsilon^*(q) \cap F \neq \varnothing\} \end{array}$$

Claramente, $L(N) = L(N')$

¿Por qué Δ' se definió así?

1. $z \in \Delta(q, a)$
2. $x \in \bigcup_{r \in \Delta(q,a)} \Delta^*_\varepsilon(r)$
3. $p \in \bigcup_{r \in \Delta^*_\varepsilon(q)} \Delta(r, a)$
4. $u \in \bigcup_{p \in \bigcup_{r \in \Delta^*_\varepsilon(q)} \Delta(r,a)} \Delta^*_\varepsilon(p)$



Varios estados finales e iniciales

La introducción de transiciones–ε permite transformar autómatas con varios estados iniciales o finales en autómatas con un solo estado inicial o final. Esta técnica será útil más adelante.

Propiedades de cerradura

Los conjuntos regulares son cerrados bajo las operaciones de:

- Unión
- Intersección
- Complemento
- Concatenación
- Estrella de Kleene

Nota: en las demostraciones siguientes se supondrá que los autómatas son completos, i.e., que la función δ no es parcial.

Demostración de cerradura bajo ∪

Sean $D_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ y $D_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ dos DFA. Construiremos un $D \in$ DFA tal que

$$L(D) = L(D_1) \cup L(D_2).$$

Definimos $D = (Q, \Sigma, \delta, s, F)$ donde

$$\begin{aligned} Q &= Q_1 \times Q_2 \\ s &= (s_1, s_2) \\ F &= (Q_1 \times F_2) \cup (F_1 \times Q_2) \\ \delta((q_1, q_2), a) &= (\delta_1(q_1, a), \delta_2(q_2, a)) \end{aligned}$$

Se puede demostrar por inducción en α que

$$\alpha \in L(D) \quad \text{sii} \quad \alpha \in L(D_1) \text{ o bien } \alpha \in L(D_2).$$

Cerradura bajo ∩ y complemento

Intersección
Se construye un autómata como en el caso de ∪, salvo que

$$F = F_1 \times F_2.$$

Complemento
Sea $A = (Q, \Sigma, \delta, s, F)$ un DFA. Sea $\bar{A} = (Q, \Sigma, \delta, s, Q - F)$. Claramente

$$L(\bar{A}) = \Sigma^* - L(A).$$

Cerradura bajo concatenación y estrella de Kleene

Sean $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ y $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ dos DFA. El autómata $C = (Q_C, \Sigma, \Delta_C, S_C, F_C)$ se construye de la siguiente forma:

$$\begin{array}{rcll} Q_C & = & Q_A \cup Q_B & \\ S_C & = & \{s_A\} & \\ F_C & = & F_B & \\ \Delta(q,a) & = & \{\delta_A(q,a)\} & \text{si } q \in Q_A \\ \Delta(q,a) & = & \{\delta_B(q,a)\} & \text{si } q \in Q_B \\ \Delta(q,\varepsilon) & = & \{s_B\} & \text{si } q \in F_A \end{array}$$

C es un NFA–ε tal que $L(C) = L(A)L(B)$. Para construir A' tal que $L(A') = L(A)^*$, se añaden transiciones–ε de los estados en F_A a un nuevo estado inicial (que también será de aceptación) y una transición–ε a s_A.

Definiciones alternativas de lenguajes regulares

Existen otras formas de definir los lenguajes regulares:

- Expresiones regulares
- Patrones
- Gramáticas lineales

Expresiones regulares

Las *expresiones regulares* son una forma muy compacta de definir lenguajes. El lenguaje generado por la expresión α es $L(\alpha)$. He aquí una definición inductiva:

- Si $a \in \Sigma$ entonces $\boldsymbol{a}$ es una expresión regular y $L(\boldsymbol{a}) = \{a\}$.
- $\boldsymbol{\emptyset}$ y $\boldsymbol{\varepsilon}$ son expresiones regulares con $L(\boldsymbol{\emptyset}) = \emptyset$ y $L(\boldsymbol{\varepsilon}) = \{\varepsilon\}$
- Las expresiones regulares α y β generan las expresiones y lenguajes:

$$\begin{array}{lrcl} \alpha + \beta & L(\alpha + \beta) & = & L(\alpha) \cup L(\beta) \\ \alpha \cdot \beta & L(\alpha \cdot \beta) & = & L(\alpha)L(\beta) \\ \alpha^* & L(\alpha^*) & = & L(\alpha)^* \end{array}$$

Ejemplos

Los números naturales en notación decimal:

$$0 + ((1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9) \cdot (0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9)^*)$$

Las fórmulas atómicas del cálculo proposicional:

$$(p + q + r) + ((p + q + r) \cdot N)$$

donde N se refiere a la expresión del ejemplo anterior.
Las cadenas del alfabeto $\{a, b, c\}$ con al menos una aparición de cada una de las tres letras:

$$(AaAbAcA) + (AaAcAbA) + (AbAaAcA) + (AcAaAbA) + (AcAbAaA) + (AbAcAaA)$$

donde A es la expresión $(a + b + c)^*$.

Patrones

Los *patrones* son otra forma muy común. También se definen inductivamente:

- $\boldsymbol{a}$, $\boldsymbol{\emptyset}$ y $\boldsymbol{\varepsilon}$ son iguales que en las expresiones regulares
- #, con $L(\#) = \Sigma$
- @, con $L(@) = \Sigma^*$
- Los patrones α y β generan los patrones y lenguajes:

$$\begin{array}{lrcl}
\alpha + \beta & L(\alpha + \beta) & = & L(\alpha) \cup L(\beta) \\
\alpha \cap \beta & L(\alpha \cap \beta) & = & L(\alpha) \cap L(\beta) \\
\alpha\beta & L(\alpha\beta) & = & L(\alpha)L(\beta) \\
\sim\alpha & L(\sim\alpha) & = & \Sigma^* - L(\alpha) \\
\alpha^* & L(\alpha^*) & = & L(\alpha)^* \\
\alpha^+ & L(\alpha^+) & = & L(\alpha)^+
\end{array}$$

Ejemplos

Las cadenas que contienen apariciones de a, b y c, en ese orden, pero no necesariamente consecutivas:

$$@a@b@c@$$

Las cadenas del alfabeto $\{a, b, c\}$ salvo las que son repeticiones consecutivas de la cadena abc:

$$\sim((abc)^+).$$

Cualquier símbolo del alfabeto seguido de cualquier cadena menos las repeticiones consecutivas de ab o cd:

$$\# \sim((ab)^+ + (cd)^+).$$

Gramáticas lineales

Recordatorio. Una gramática es $G = \langle \Sigma, \Gamma, S, \to \rangle$.

Sean $A, B \in \Gamma$ y $\alpha \in \Sigma^*$. Si todas las reglas de G tienen una de las siguientes dos formas

$$A \to \alpha B \qquad \text{o bien} \qquad A \to \alpha$$

se trata de una *gramática lineal por la derecha*.

Si todas las reglas de G son de alguna de las dos formas

$$A \to B\alpha \qquad \text{o bien} \qquad A \to \alpha$$

es una *gramática lineal por la izquierda*.

Ejemplos

- La gramática $G_N = \langle\{0,\ldots 9\}, \{S, N\}, S, \rightarrow\rangle$ con reglas de producción

$$\begin{aligned} S &\rightarrow 0 \mid 1N \mid 2N \mid \ldots \mid 9N \\ N &\rightarrow 0N \mid 1N \mid \ldots 9N \mid \varepsilon, \end{aligned}$$

  genera los números naturales en notación decimal.
- La gramática $G_P = \langle\{p, q, r, 0, \ldots 9\}, \{S', S, N\}, S', \rightarrow\rangle$ con las siguientes reglas adicionales a las del ejemplo anterior:

$$S' \rightarrow p \mid q \mid r \mid pS \mid qS \mid rS$$

  genera las fórmulas atómicas del cálculo de proposiciones.

Equivalencia entre patrones y expresiones regulares I

Para todo patrón α, existe una expresión regular β tal que $L(\alpha) = L(\beta)$.

Demostración. Por inducción en α.

Primero los casos básicos: a, ε, $\emptyset$, # y @. Los tres primeros tienen expresiones regulares equivalentes obvias.
En cuanto # y @, si $\Sigma = \{c_1, \ldots, c_m\}$, entonces

$$L(\#) = L(c_1 + \cdots + c_m) \quad \text{y} \quad L(@) = L((c_1 + \cdots + c_m)^*)$$

Hipótesis inductiva. Supongamos que dados los patrones α y β, existen expresiones regulares γ y η tales que

$$L(\alpha) = L(\gamma) \qquad L(\beta) = L(\eta)$$

Equivalencia entre patrones y expresiones regulares II

Procedemos a probar los casos inductivos con los operadores +, concatenación, *, $^+$ $\cap$ y $\sim$.
Los tres primeros son obvios, pues también se encuentran presentes en las expresiones regulares.
En cuanto a $^+$, basta observar que α^+ es equivalente a $\alpha\alpha^*$.
La prueba del patrón $\sim\alpha$ se deja pendiente.
Finalmente, el patrón $\alpha \cap \beta$ es equivalente al patrón $\sim(\sim\alpha + \sim\beta)$, el cual queda cubierto por los casos anteriores.

Nota. La afirmación inversa es trivialmente cierta.

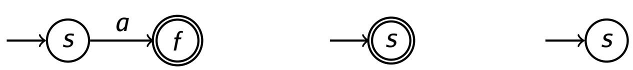
Equivalencia entre expresiones regulares y autómatas

Para toda expresión regular α, existe un autómata finito A tal que

$$L(\alpha) = L(A).$$

La demostración se hará por inducción en α.

Casos básicos. Para las expresiones $\boldsymbol{a}$, $\boldsymbol{\varepsilon}$ y $\boldsymbol{\emptyset}$, se tienen los siguientes autómatas:



Casos inductivos. Ya se estudió cómo construir autómatas que acepten la unión, la concatenación y la estrella de Kleene de lenguajes regulares.

Equivalencia entre autómatas y expresiones regulares I

Sea $N = (Q, \Sigma, \Delta, S, F) \in$ NFA y sean $X \subseteq Q$ y $p, q \in Q$. Definiremos una expresión regular α_{pq}^X tal que

$$L(\alpha_{pq}^X) = \{\beta \mid q \in \Delta^*(\{p\}, \beta) \text{ y el camino pasa sólo por estados en } X\}.$$

Por inducción en X. Sean $a_1, \ldots a_k$ todos los símbolos de Σ tales que $q \in \Delta(p, a_i)$.
Si $p \neq q$

$$\alpha_{pq}^{\emptyset} = \begin{cases} a_1 + \cdots + a_k & 1 \leq k \\ \emptyset & k = 0 \end{cases}$$

Si $p = q$

$$\alpha_{pq}^{\emptyset} = \begin{cases} a_1 + \cdots + a_k + \varepsilon & 1 \leq k \\ \varepsilon & k = 0 \end{cases}$$

Equivalencia entre autómatas y expresiones regulares II

Si $X \neq \emptyset$, sea $r \in X$

$$\alpha_{pq}^X = \alpha_{pq}^{X-\{r\}} + \alpha_{pr}^{X-\{r\}}(\alpha_{rr}^{X-\{r\}})^* \alpha_{rq}^{X-\{r\}}.$$

Finalmente, si $s_1, \ldots, s_n \in S$ y $f_1, \ldots, f_m \in F$, entonces

$$L(N) = L(\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{s_i f_j}^Q)$$

Equivalencia entre gramáticas lineales y autómatas I

Sea $G = \langle \Sigma, \Gamma, S, \rightarrow \rangle$ una gramática lineal por la derecha. Sea $N \in$ NFA$-\varepsilon$ el siguiente autómata:

$$\begin{array}{lcll} Q & = & \{[\alpha] \mid \exists V \in \Gamma. V \rightarrow \beta\alpha\} \cup \{[S]\} & \\ \Delta([V], \varepsilon) & = & \{[\alpha] \mid V \rightarrow \alpha\} & \text{si } V \in \Gamma \\ \Delta([a\alpha], a) & = & \{[\alpha]\} & \text{si } a \in \Sigma \wedge \alpha \in \Sigma^* \cup \Sigma^*\Gamma \\ \{[S]\} & = & \text{estado inicial} & \\ \{[\varepsilon]\} & = & \text{estado final.} & \end{array}$$

Entonces

$$[\alpha] \in \Delta^*([S], \gamma) \text{ sii } S \Rightarrow^* \eta V \Rightarrow \eta\theta\alpha,$$

si (a) $V \rightarrow \theta\alpha$ y $\gamma = \eta\theta$ o (b) $\alpha = S$ y $\gamma = \varepsilon$.
Demostración. Inducción en $\Rightarrow^*$. A partir de este resultado, es claro que

$$L(N) = L(G).$$

Equivalencia entre gramáticas lineales y autómatas II

A la inversa, sea $A = (Q, \Sigma, \delta, s, F) \in$ DFA. Sea $G = \langle \Sigma, Q, s, \rightarrow \rangle$ una gramática con producciones de la forma

$$\begin{array}{rcl} q \rightarrow ar & \text{sii} & \delta(q, a) = r \\ q \rightarrow a & \text{sii} & \delta(q, a) \in F \end{array}$$

Entonces

$$\delta^*(q, \alpha) = r \quad \text{sii} \quad q \Rightarrow_G^* \alpha r$$

Ejemplos

Sea $G = \langle\{0, 1\}, \{S, N\}, S\rangle$ una gramática con las siguientes reglas de producción:

$$S \to 0 \mid 1N \qquad N \to 0N \mid 1N \mid \varepsilon$$

El autómata respectivo es