Expresiones Regulares

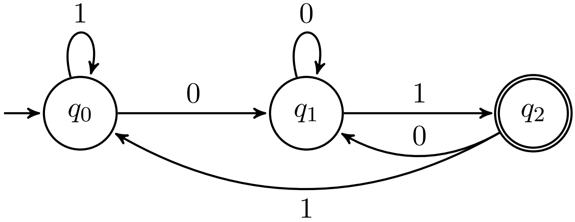
Introducción a la Lógica y la Computación
FAMAF, Universidad Nacional de Córdoba

01/11/17

Temas del Día

Intuiciones I

- Sabemos que los autómatas finitos (DFA, NFA y NFA-ϵ) definen lenguajes.



$$L(M) = \{w \mid w \text{ termina en } 01\}$$

- Notación **machine-like** para especificar lenguajes.
- Convengamos que no es una notación muy cómoda!

Intuiciones II

Las expresiones regulares:

- Nos permiten dar una descripción algebraica de un lenguaje.

$$(\mathbf{0}+\mathbf{1})^*\mathbf{01}$$

- Introducidas en los 50s por Kleene.
- Se han vuelto una herramienta de uso común en muchos sistemas operativos y lenguajes de programación.
- Muchas aplicaciones de procesamiento de texto las utilizan.
    - Analizadores léxicos.
    - Búsqueda de textos.
- Estudiaremos una versión básica de las expresiones regulares.

Temas del Día

1. Expresiones Regulares.
   - Intuiciones
   - Operaciones sobre lenguajes
   - Algebra de Expresiones Regulares
2. Autómatas Finitos y Expresiones Regulares
   - De Expresiones regulares a Autómatas
   - De Autómatas a Expresiones Regulares
3. Lenguajes Regulares

Unión y Concatenación de Lenguajes

Unión

Unión de dos lenguajes L_1 **y** L_2 $(L_1 \cup L_2)$: lenguaje formado por las cadenas que estan en L_1 y las cadenas que están en L_2.

Ejemplo: Sea $L_1 = \{00, 11\}$ y $L_2 = \{100, 111\}$ luego,

$$L_1 \cup L_2 = \{00, 11, 100, 111\}.$$

Concatenación

Concatenación de dos lenguajes L_1 **y** L_2 $(L_1 L_2)$: lenguaje que contiene a todas las palabras que pueden ser formadas concatenando cualquier palabra en L_1 con cualquier palabra en L_2, es decir, $L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$.

Ejemplo: Sea $L_1 = \{00, 11\}$ y $L_2 = \{100, 111\}$, luego,

$$L_1 L_2 = \{00100, 00111, 11100, 11111\}.$$

Clausura de un lenguaje

- Definamos $L^0 = \{\epsilon\}$, $L^1 = L$ y $L^i = LL^{i-1}$, para $i \geq 1$.

Clausura de Kleene

La **clausura de un lenguaje** L (L^*) representa el conjunto de palabras que se pueden formar concatenando un nro. arbitrario de cadenas de L. Formalmente,

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

La **clausura positiva de** L (L^+) se define como

$$L^+ = \bigcup_{i=1}^{\infty} L^i.$$

Clausura de un Lenguaje

Ejemplo: Sea $L = \{0, 11\}$. Calculemos L^*.

$$L^* = \bigcup_{i=0}^{\infty} L^i = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots$$

$$L^0 = \{\epsilon\}$$

$$L^1 = L = \{0, 11\}$$

$$L^2 = LL^1 = \{00, 011, 110, 1111\}$$

$$L^3 = LL^2 = \{000, 0011, 0110, 01111, 1100, 11011, 11110, 111111\}$$

$$\begin{aligned} L^* \quad = \quad & \{\epsilon, 0, 11, 00, 011, 110, 1111, \\ & \qquad 000, 0011, 0110, 01111, 1100, \\ & \qquad\qquad 11011, 11110, 111111, \ldots\} \end{aligned}$$

Temas del Día

1. Expresiones Regulares.
   - Intuiciones
   - Operaciones sobre lenguajes
   - Algebra de Expresiones Regulares
2. Autómatas Finitos y Expresiones Regulares
   - De Expresiones regulares a Autómatas
   - De Autómatas a Expresiones Regulares
3. Lenguajes Regulares

Sintaxis de Expresiones Regulares

Sintaxis

El conjunto de las expresiones regulares sobre el alfabeto Σ, ER_Σ es el menor conjunto tal que:

1. $\emptyset \in ER_\Sigma$.
2. $\epsilon \in ER_\Sigma$.
3. $\mathbf{a} \in ER_\Sigma$, $\forall a \in \Sigma$.

Si r y s son expresiones regulares, entonces

4. $\mathbf{r} + \mathbf{s} \in ER_\Sigma$.
5. $\mathbf{rs} \in ER_\Sigma$ (o $\mathbf{r} \cdot \mathbf{s} \in ER_\Sigma$).
6. $\mathbf{r}^* \in ER_\Sigma$.
7. $(\mathbf{r}) \in ER_\Sigma$.

Semántica de expresiones regulares

Lenguaje de una expresión regular

El lenguaje asociado a una expresión regular r, $L(r)$, se define recursivamente como:

1. $L(\emptyset) = \emptyset$.
2. $L(\epsilon) = \{\epsilon\}$.
3. $L(\mathbf{a}) = \{a\}\ \forall \mathbf{a} \in ER_\Sigma$.

Si r y s son expresiones regulares, entonces

4. $L(\mathbf{r} + \mathbf{s}) = L(\mathbf{r}) \cup L(\mathbf{s})$.
5. $L(\mathbf{rs}) = L(\mathbf{r})L(\mathbf{s})$.
6. $L(\mathbf{r}^*) = L(\mathbf{r})^*$.
7. $L((\mathbf{r})) = L(\mathbf{r})$.

Precedencia de Operadores

Precedencia

Sean $*, +, \cdot$ los operadores utilizados en expresiones regulares. Definimos su precedencia de la siguiente forma:

$$* > \cdot > +$$

Ejemplo:

- $((\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}) = \mathbf{01}^* + \mathbf{1}$
- $((\mathbf{01})^*) + \mathbf{1}) \neq \mathbf{01}^* + \mathbf{1}$

Ejemplos

1. $\mathbf{00}$: $L(\mathbf{00}) = L(\mathbf{0})L(\mathbf{0}) = \{00\}$.
2. $r = (\mathbf{0} + \mathbf{1})^*$:

$$L(r) = \{w \mid w \text{ es una cadena de 0s y 1s}\}.$$

3. $r = (\mathbf{0} + \mathbf{1})^*\mathbf{00}(\mathbf{0} + \mathbf{1})^*$:

$L(r) =$ cadenas de 0s y 1s con al menos dos 0s consecutivos.

4. $r = (\mathbf{0} + \mathbf{1})^*\mathbf{011}$:

$L(r) = \{w \mid w$ es una cadena de 0s y 1s que termina en $011\}$.

5. $r = (\mathbf{1} + \mathbf{10})^*$: $L(r) =$ cadenas de 0s y 1s que comienzan con 1 y no tienen dos 0s consecutivos.
6. $r = \mathbf{0}^*\mathbf{1}^*\mathbf{2}^*$: $L(r) =$ cadenas que tienen cierta cantidad de 0s seguido por cierta cantidad de 1s seguido por cierta cantidad de 2s.

Recordar!

Importante

No confundir una expresión regular con una palabra del lenguaje.

- $\mathbf{001} \neq 001$.
- Una expresión regular es un patrón que describe un conjunto de palabras.

Leyes Algebraicas

Leyes Algebraicas

Sean r, s y t expresiones regulares. Se cumple lo siguiente:

- $\mathbf{r}+\mathbf{s}=\mathbf{s}+\mathbf{r}$.
- $(\mathbf{r}+\mathbf{s})+\mathbf{t}=\mathbf{r}+(\mathbf{s}+\mathbf{t})$.
- $(\mathbf{rs})\mathbf{t}=\mathbf{r}(\mathbf{st})$.
- $\emptyset+\mathbf{r}=\mathbf{r}+\emptyset=\mathbf{r}$.
- $\epsilon\mathbf{r}=\mathbf{r}\epsilon=\mathbf{r}$.
- $\emptyset\mathbf{r}=\mathbf{r}\emptyset=\emptyset$.
- $\mathbf{r}(\mathbf{s}+\mathbf{t})=\mathbf{rs}+\mathbf{rt}$.
- $(\mathbf{s}+\mathbf{t})\mathbf{r}=\mathbf{sr}+\mathbf{tr}$.
- $\mathbf{r}+\mathbf{r}=\mathbf{r}$.
- $(\mathbf{r}^*)^*=\mathbf{r}^*$.
- $\emptyset^*=\epsilon$.
- $\epsilon^*=\epsilon$.
- $\mathbf{r}^+=\mathbf{rr}^*=\mathbf{r}^*\mathbf{r}$.
- $\mathbf{r}^*=\mathbf{r}^++\epsilon$.

Autómatas y Expresiones Regulares

- Los autómatas finitos y las expresiones regulares definen lenguajes.
- Ahora vamos a ver que ambos definen los **mismos** lenguajes.
- Vamos a probar que partiendo de una expresión regular podemos construir un NFA-ϵ que define el mismo lenguaje.
  - Bastante sencillo!
- Vamos a probar que partiendo de un NFA-ϵ podemos construir una expresión regular que define el mismo lenguaje.
  - Mas complicado!.

Temas del Día

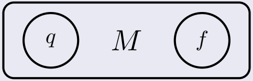
1. Expresiones Regulares.
   - Intuiciones
   - Operaciones sobre lenguajes
   - Algebra de Expresiones Regulares
2. Autómatas Finitos y Expresiones Regulares
   - De Expresiones regulares a Autómatas
   - De Autómatas a Expresiones Regulares
3. Lenguajes Regulares

Un poco de notación

- Queremos ver que a toda expresión regular r le corresponde un autómata NFA-ϵ M tal que $L(r) = L(M)$.

Representación de un autómata

Sea M un autómata M con estado inicial q y estado final f. Lo representaremos de la siguiente forma:



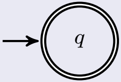
Transformando expresiones regulares en autómatas

Algoritmo

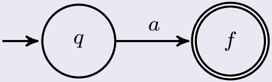
Sean $r, s \in ER_\Sigma$. Definimos inductivamente el autómata M_r correspondiente a una expresión regular r de la siguiente forma:

1. A $\emptyset$ le corresponde el autómata:

2. A ϵ le corresponde el autómata:



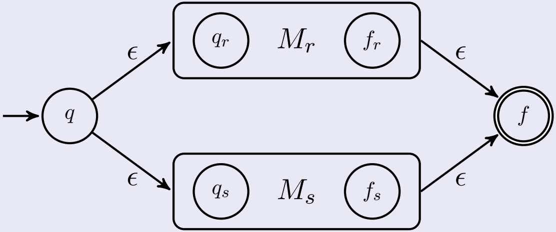
3. Para toda $\mathbf{a} \in ER_\Sigma$, le corresponde el autómata:



Transformando expresiones regulares en autómatas

Algoritmo (cont.)

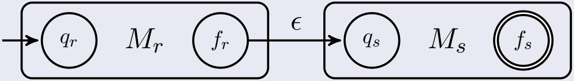
4. A $\mathbf{r} + \mathbf{s}$ le corresponde el autómata:



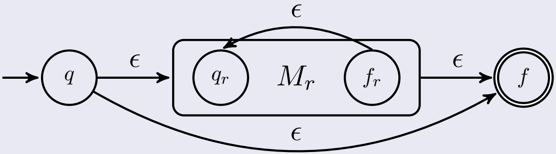
Transformando expresiones regulares en autómatas

Algoritmo (cont.)

5. A $\mathbf{rs}$ le corresponde el autómata:



6. A $\mathbf{r}^*$ le corresponde el autómata:

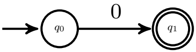


Ejemplo

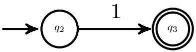
Construir el NFA-ϵ para $r = \mathbf{01}^* + \mathbf{1}$.

$L(r) = \{w|w = 1$ o w empieza con 0 seguida por con cero o mas 1's$\}$

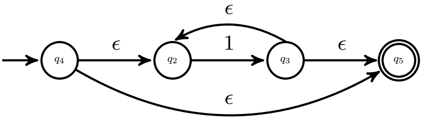
- $\mathbf{0}$:



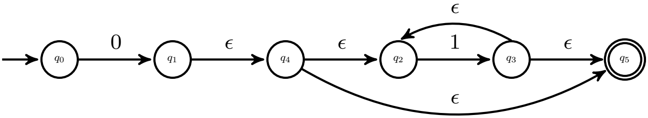
- $\mathbf{1}$:



- $\mathbf{1}^*$:

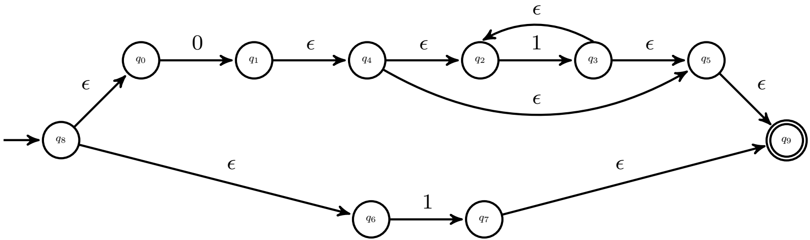


- $\mathbf{01}^*$:



Ejemplo (cont.)

- $\mathbf{01^* + 1}$:



Temas del Día

1. Expresiones Regulares.
   - Intuiciones
   - Operaciones sobre lenguajes
   - Algebra de Expresiones Regulares
2. Autómatas Finitos y Expresiones Regulares
   - De Expresiones regulares a Autómatas
   - De Autómatas a Expresiones Regulares
3. Lenguajes Regulares

Intuiciones

- Vamos a construir expresiones regulares que describan "caminos" dentro del autómata.
- Nos interesan los caminos que van desde el estado inicial a los estados finales.
- Camino = Palabra.
- En cada paso de la construcción vamos a considerar autómatas mas pequenõs.

Un poco de notación

Dados $M = (Q, \Sigma, \delta, q_0, F)$, $R \subseteq Q$ y $q_n, q_m \in R$, definimos

$$M_{nm}(R) = (R, \Sigma, \delta_{\restriction R}, q_n, \{q_m\}).$$

Algoritmo

Definición

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un autómata. Sea $Q = \{q_0, \ldots, q_r\}$ y $\Sigma = \{a_1, \ldots, a_u\}$. Dados $0 \leq n, m \leq r$, $R \subseteq Q$, definimos recursivamente las siguientes expresiones regulares.

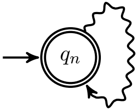
$$L_{nm}(R) = \begin{cases} \emptyset & \text{si } q_n \text{ o } q_m \notin R. \\ I_n(R)^* & \text{si } n = m. \\ I_n(R)^* F_{nm}(R) & \text{si } n \neq m. \end{cases}$$

$$I_n(R) = \sum_{\substack{q_n \xrightarrow{a} q_t \\ q_s \xrightarrow{b} q_n}} \mathbf{a} L_{ts}(R \backslash \{q_n\}) \mathbf{b} + \sum_{q_n \xrightarrow{c} q_n} \mathbf{c}.$$

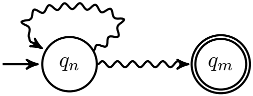
$$F_{nm}(R) = \sum_{q_n \xrightarrow{a} q_t} \mathbf{a} L_{tm}(R \backslash \{q_n\}).$$

Entendiendo el algoritmo - $L_{nm}(R)$

- $L_{nm}(R)$ es el lenguaje de las cadenas que llevan del estado q_n al estado q_m en el autómata $M_{nm}(R)$.
- $L_{nm}(R)$ depende de n, m y R:
- Si $q_n, q_m \notin R$: $L_{nm}(R)$ es el lenguaje vacío.
- Si $q_n, q_m \in R$ y $q_n = q_m$: $L_{nm}(R)$ es el lenguaje de las cadenas que salen de q_n y vuelven a q_n, sin pasar por q_n.



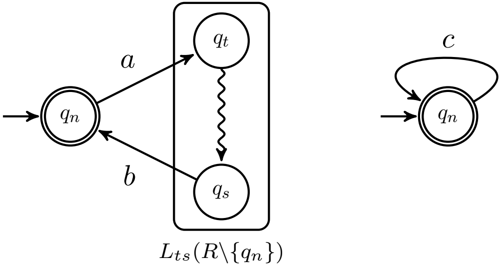
- Si $q_n, q_m \in R$ y $q_n \neq q_m$: $L_{nm}(R)$ es el lenguaje que resulta de concatenar el lenguaje de las cadenas que salen de q_n y vuelven a q_n, sin pasar por q_n, con el lenguaje de las cadenas que van de q_n a q_m.



Entendiendo el algoritmo - $I_n(R)$

$$I_n(R) = \sum_{q_n \xrightarrow{a} q_t, q_s \xrightarrow{b} q_n} \mathbf{a} L_{ts}(R \backslash \{q_n\}) \mathbf{b} + \sum_{q_n \xrightarrow{c} q_n} \mathbf{c}.$$

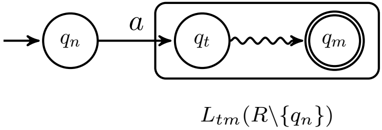
- $I_n(R)$ es el lenguaje de las cadenas que salen del estado q_n y vuelven a q_n, sin pasar por q_n, en el autómata $M_{nn}(R)$.
- $I_n(R)$ es la unión de 2 lenguajes.
- Las cadenas que **salen** de q_n y **regresan** a q_n.
- Las cadenas de longitud 1 (loops) que salen y entran a q_n.



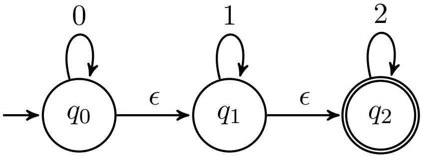
Entendiendo el algoritmo - $F_{nm}(R)$

$$F_{nm}(R) = \sum_{q_n \xrightarrow{a} q_t} \mathbf{a} L_{tm}(R \backslash \{q_n\}).$$

- $F_{nm}(R)$ es lenguaje de las cadenas que **salen** del estado q_n y llevan al estado q_m.



Ejemplo



$$L_{02}(Q) = I_0(Q)^* F_{02}(Q) = \mathbf{0}^*(\mathbf{0}+\epsilon)\mathbf{1}^*(\mathbf{1}+\epsilon)\mathbf{2}^* = \mathbf{0}^*\mathbf{1}^*\mathbf{2}^*$$
$$I_0(Q) = \mathbf{0}$$
$$F_{02}(Q) = (\mathbf{0}+\epsilon)L_{12}(Q\backslash\{q_0\}) = (\mathbf{0}+\epsilon)\mathbf{1}^*(\mathbf{1}+\epsilon)\mathbf{2}^*$$
$$L_{12}(Q\backslash\{q_0\}) = I_1(Q\backslash\{q_0\})^* F_{12}(Q\backslash\{q_0\}) = \mathbf{1}^*(\mathbf{1}+\epsilon)\mathbf{2}^*$$
$$I_1(Q\backslash\{q_0\}) = \mathbf{1}$$
$$F_{12}(Q\backslash\{q_0\}) = (\mathbf{1}+\epsilon)L_{22}(Q\backslash\{q_0,q_1\}) = (\mathbf{1}+\epsilon)\mathbf{2}^*$$
$$L_{22}(Q\backslash\{q_0,q_1\}) = I_2(Q\backslash\{q_0,q_1\})^* = \mathbf{2}^*$$
$$I_2(Q\backslash\{q_0,q_1\}) = \mathbf{2}$$

Equivalencia entre expresiones regulares y autómatas

Teorema

Sea r una expresión regular y $L(r)$ el lenguaje asociado. Entonces existe un NFA-ϵ M_r tal que $L(M_r) = L(r)$.

Prueba: Por inducción en la estructura de r.

Teorema de Kleene

Sea M un NFA con ϵ-transiciones y $L(M)$ el lenguaje asociado a M. Entonces existe una expresión regular r tal que $L(M_r) = L(r)$.

Prueba: Hay que probar 2 lemas intermedios.

Lenguajes Reconocidos

- Los DFA, NFA, NFA-ϵ y las expresiones regulares son equivalentes.
  - Reconocen los mismos lenguajes: **lenguajes regulares**.
- ¿Todos los lenguajes pueden ser descriptos de esta forma?.
- NO. Existen lenguajes que no pueden ser descriptos por autómatas finitos y/o expresiones regulares.

Ejemplo

Considere el lenguaje $L_{01} = \{0^n 1^n \mid n \geq 1\}$. ¿Es regular?

Prueba: Si L_{01} es regular existe un DFA M con k estados tq $L(M) = L_{01}$. Sabemos que si $|\alpha| = k$ entonces hay $k+1$ prefijos. Supongamos que M recibe k 0's.Por cada prefijo, M esta en un estado, como hay $k+1$ prefijos y k estados, por el principio del pidgeonhole, existen $i, j \leq k+1$ y $i \neq j$ tq $q_0 \xrightarrow{0^i} p$ y $q_0 \xrightarrow{0^j} p$. Si luego de recibir el prefijo 0^i (ó 0^j) M empieza a recibir 1's, luego de i 1's M debería aceptar la cadena ($p \xrightarrow{1^i} q_f$, $q_f \in F$) solo si antes se recibieron i 0's (prefijo 0^i) y no aceptar la cadena si se recibieron j 0's. Sin embargo como estamos en el estado p, no podemos distinguir si hemos recibido el prefijo 0^i ó 0^j.Por lo tanto, el autómata podria aceptar la cadena $0^j 1^i$ con $i \neq j$. Luego L_{01} no es un lenguaje regular.