Propiedades de los lenguajes regulares

Horst H. von Brand
vonbrand@inf.utfsm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

Contenido

Propósito

Nos daremos un tiempo para responder algunas preguntas pendientes (¿Por qué no se incluye intersección, u otras, operaciones al definir lenguajes regulares?) y agregar al arsenal de técnicas para construir lenguajes regulares. A la vez, esto da nuevas herramientas para demostrar que un lenguaje no es regular: suponga que es regular, aplique operaciones que preservan la regularidad para obtener uno que sabe no es regular, contradicción.

Operaciones regulares

El siguiente resultado es trivial:

Teorema

Los lenguajes regulares son cerrados respecto de las operaciones regulares (unión, concatenación, estrella de Kleene) y respecto de las operaciones «extraoficiales» (más, potencias fijas).

Demostración.

Obvio por la definición de expresiones regulares y expresar las operaciones extraoficiales en término de las primeras. □

Reverso

Teorema

Los lenguajes regulares son cerrados respecto de reverso.

Si *damos vuelta* un DFA que reconoce L, obtenemos un NFA que reconoce L^{R}.

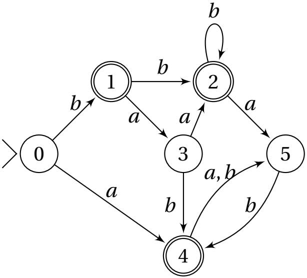
Demostración.

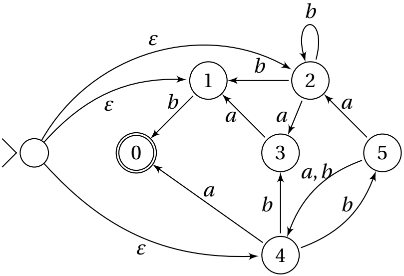
Sea $L = \mathscr{L}(M)$ con el DFA $M = (Q, \Sigma, \delta, q_0, F)$. Sea el NFA $M' = (Q \cup \{q_0'\}, \Sigma.\delta', q_0', \{q_0\})$, donde $q_0' \notin Q$ y definimos:

$$\begin{aligned} \delta'(q_0', \varepsilon) &= F & \delta'(q_0', a) &= \varnothing \quad \text{si } a \in \Sigma \\ \delta'(q, a) &= \{p \in Q \colon \delta(p, a) = q\} & \delta'(q, \varepsilon) &= \varnothing \quad \text{si } q \in Q \end{aligned}$$

Es fácil ver que si $\sigma \in \mathscr{L}(M)$ entonces $\sigma^{\mathrm{R}} \in \mathscr{L}(M')$. □

Un DFA y su reverso





Operaciones entre conjuntos

Complemento

Teorema

Los lenguajes regulares son cerrados respecto complemento.

Demostración.

Sea $L = \mathscr{L}(M)$ para el DFA $M = (Q, \Sigma, \delta, q_0, F)$. El DFA $\overline{M} = (Q, \Sigma, \delta, q_0, \overline{F})$ acepta $\overline{L}$. □

Operaciones entre conjuntos

Intersección

Teorema

Los lenguajes regulares son cerrados respecto intersección.

Demostración.

Sean lenguajes $L_1 = \mathscr{L}(M_1)$ y $L_2 = \mathscr{L}(M_2)$ para DFAs $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ y $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$. El DFA $M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2)$, donde:

$$\delta((q', q''), a) = (\delta_1(q', a), \delta_2(q'', a))$$

acepta $L_1 \cap L_2$. $\square$

Operaciones entre conjuntos

Intersección

Alternativamente:

Demostración.
Sean lenguajes L_1 y L_2 regulares. Entonces es regular $L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$ por las propiedades demostradas antes. $\square$

Operaciones entre conjuntos

Otras operaciones

Otras operaciones entre conjuntos pueden expresarse con las anteriores (unión, complemento, intersección), por lo que podemos demostrar que los lenguajes regulares son cerrados respecto de ellas usando las estrategias anteriores: construir un DFA que simula DFAs para los lenguajes en paralelo y decide según los resultados (esencialmente, aplicando la operación del caso entre los conjuntos de estados finales) o usar equivalencias entre conjuntos y las operaciones básicas.

Operaciones entre conjuntos

Diferencia simétrica
Como un ejemplo, daremos dos demostraciones de:

Teorema

Los lenguajes regulares son cerrados respecto diferencia simétrica.

Operaciones entre conjuntos

Diferencia simétrica — autómatas

Demostración.

Sean $L_1 = \mathscr{L}(M_1)$ y $L_2 = \mathscr{L}(M_2)$ para DFAs $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ y $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$. El DFA $M = (Q_1 \times Q_2, \delta, (q_1, q_2), F)$ donde:

$$\delta((q', q''), a) = (\delta_1(q', a), \delta_2(q'', a)) \qquad \text{todo } q' \in Q_1,\ q'' \in Q_2,\ a \in \Sigma$$

$$F = (F_1 \times Q_2) \bigtriangleup (Q_1 \times F_2)$$

claramente acepta $L_1 \bigtriangleup L_2$. □

Operaciones entre conjuntos

Diferencia simétrica — propiedades

Demostración.

Para conjuntos L_1, L_2 podemos expresar:

$$L_1 \triangle L_2 = (L_1 \smallsetminus L_2) \cup (L_2 \smallsetminus L_1)$$

A su vez:

$$L_1 \smallsetminus L_2 = L_1 \cap \overline{L_2}$$

Hemos visto que los lenguajes regulares son cerrados respecto de todas las operaciones involucradas. $\square$

Substitución

Definición

Sea Σ un alfabeto, y para cada $a \in \Sigma$ sea L_a un lenguaje sobre Γ. Para la palabra $\alpha = a_1 a_2 \cdots a_n$ sobre Σ, la *substitución* σ dada por los anteriores se define por:

$$\sigma(\alpha) = L_{a_1} L_{a_2} \cdots L_{a_n}$$

Extendemos de la forma obvia a lenguajes. Para $L \subseteq \Sigma^*$:

$$\sigma(L) = \bigcup_{\alpha \in L} \sigma(\alpha)$$

Substitución

Teorema

Los lenguajes regulares son cerrados respecto de substitución de lenguajes regulares.

Demostración.

Sea $L \subseteq \Sigma^*$, regular, y sean $L_a \subseteq \Gamma^*$, regulares para todo $a \in \Sigma$. Si en una expresión regular para L substituimos a por una expresión regular para L_a, el resultado es una expresión regular para $\sigma(L)$. □

Homomorfismos

Definición

Sean Σ, Γ alfabetos. Un *homomorfismo* es una función $h \colon \Sigma^* \to \Gamma^*$ tal que para todo $\alpha, \beta \in \Sigma^*$ es $h(\alpha \cdot \beta) = h(\alpha) \cdot h(\beta)$.

Note que lo anterior implica:

$$h(\varepsilon) = \varepsilon$$
$$h(a_1 a_2 \cdots a_n) = h(a_1) h(a_2) \cdots h(a_n)$$

Basta definir $h(a)$ para $a \in \Sigma$.

Extendemos de la forma obvia a lenguajes:

$$h(L) = \{h(\alpha) \colon \alpha \in L\}$$

Homomorfismos

Teorema

Los lenguajes regulares son cerrados respecto de homomorfismos.

Demostración.

Si σ es la substitución definida por los lenguajes $L_a = \{h(a)\}$, vemos que:

$$h(L) = \sigma(L)$$

Cada L_a es finito, y por tanto regular. $\square$

Homomorfismo inverso

Un homomorfismo no siempre es una biyección, con lo que en rigor no tiene inverso.

Definición

Sea $L \subseteq \Gamma^*$ y $h \colon \Sigma^* \to \Gamma^*$ un homomorfismo. Definimos el *homomorfismo inverso:*

$$h^{-1}(L) = \{\alpha \in \Sigma^* \colon h(\alpha) \in L\}$$

En rigor, esto no es un inverso. Formalmente se le llama la *preimagen* de L bajo h. Pero el nombre es tradicional.

Homomorfismo inverso

Teorema

Los lenguajes regulares son cerrados respecto de homomorfismos inversos.

Homomorfismo inverso

La idea de la demostración es usar un DFA que reconoce L, aplicándolo a $h(\alpha)$; si acepta, $\alpha \in h^{-1}(L)$. Aplicar h a α es aplicarlo símbolo a símbolo.

Demostración.
Sea $L = \mathscr{L}(M)$, donde $M = (Q, \Gamma, \delta, q_0, F)$ es un DFA, y sea $h \colon \Sigma^* \to \Gamma^*$ un homomorfismo.
El DFA $M' = (Q, \Sigma, \delta', q_0, F)$ acepta $h^{-1}(L)$, donde:

$$\delta'(q, a) = \delta(q, h(a))$$

$\square$

Raíz cuadrada

Definimos la «raíz cuadrada» de un lenguaje:

$$\operatorname{sqrt}(L) = \{\alpha \colon \alpha\alpha \in L\}$$

Raíz cuadrada

Teorema

Los lenguajes regulares son cerrados respecto de sqrt.

Raíz cuadrada

La idea es construir un NFA que adivine el estado a la mitad de la palabra y ejecute dos copias del DFA, una desde el estado inicial y otra desde el estado del medio, verificando que el primero llegue al estado del medio y el segundo a un estado final.

Raíz cuadrada

Demostración.

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un DFA que acepta L. Construimos el NFA $M' = (Q', \Sigma, \delta', q'_0, F')$ como sigue:

- Registramos los estados de dos M procesando como primera y segunda mitades y el estado del medio: $Q' = Q \times Q \times Q$.
- Transiciones son de las copias de M, procesando como primera y segunda mitad, recordando el estado del medio: $\delta'((q', q'', q_m), a) = \{(\delta(q', a), \delta(q'', a), q_m)\}$ para todo $q', q'', q_m \in Q$ y todo $a \in \Sigma$.

Raíz cuadrada

- Agregamos q_0' desde el cual iniciar las copias del DFA: $\delta'(q_0', \varepsilon) = \{(q_0, q_m, q_m) \colon q_m \in Q\}$.
- Estados son finales si la primera copia del DFA llega a q_m y la segunda a un estado final: $F' = \{(q_m, q_f, q_m) \colon q_m \in Q, q_f \in F\}$

$\square$

Resumen

- Demostramos una variedad de operaciones que preservan la regularidad. Demostrar que los lenguajes regulares son cerrados respecto de operaciones adicionales son ejemplos y ejercicios populares en textos de teoría de autómatas.
- Aplicaciones son construcciones simples para ciertos lenguajes regulares, o demostrar (por contradicción) que un lenguaje no es regular. Ejemplos y ejercicios en apuntes del ramo.