Parte III: Lenguajes y Autómatas

1 de noviembre de 2019

Repaso

Alfabeto Conjunto finito no vacío, habitualmente usamos Σ para referirnos a alfabetos.

Cadena (fijado el alfabeto) Secuencia finita de símbolos de Σ.

Lenguaje un subconjunto de Σ^*.

Lenguaje Regular puede ser reconocido por un AFD, o un AFN o un AFN-ϵ.

Hablemos de lenguajes

Puesto que un lenguaje es un subconjunto de Σ^* podemos hablar de:

vacío	$\emptyset$
total	Σ^*
complemento	$\bar{L} = \Sigma^* \setminus L$
intersección	$L \cap L' = \{\alpha \,\vert\, \alpha \in L \text{ y } \alpha \in L'\}$
unión	$L \cup L' = \{\alpha \,\vert\, \alpha \in L \text{ o } \alpha \in L'\}$
concatenación	$LL' = \{\alpha\beta \,\vert\, \alpha \in L \text{ y } \beta \in L'\}$
potencias	$L^n = \begin{cases} \{\epsilon\} & \text{si } n = 0 \\ LL^k & \text{si } n = k+1 \end{cases}$
clausura	$L^* = \bigcup_{n \in \mathbb{N}} L^n$

Expresiones regulares

Las expresiones regulares son una forma más algebraica de definir lenguajes regulares. Fijado un alfabeto Σ las definimos inductivamente:

vacío	$\emptyset \in ER_{\Sigma}$.
épsilon	$\epsilon \in ER_{\Sigma}$.
símbolo	Si $x \in \Sigma$, entonces $\mathbf{x} \in ER_{\Sigma}$.
unión	Si $A, B \in ER_{\Sigma}$, entonces $A + B \in ER_{\Sigma}$.
concatenación	Si $A, B \in ER_{\Sigma}$, entonces $AB \in ER_{\Sigma}$.
clausura	Si $A \in ER_{\Sigma}$, entonces $A^* \in ER_{\Sigma}$.

Lenguaje de una expresión regular

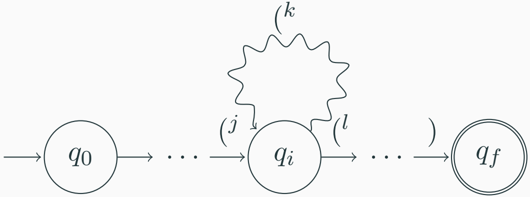
Como ER_Σ fue definido inductivamente, podemos definir sus lenguajes por recursión:

vacío	$L(\emptyset) = \emptyset.$
épsilon	$L(\epsilon) = \{\epsilon\}.$
símbolo	$L(\mathbf{x}) = \{x\}.$
unión	$L(A + B) = L(A) \cup L(B).$
concatenación	$L(AB) = L(A) \cup L(B).$
clausura	$L(A^*) = (L(A))^*.$

$L_{()} = \{\alpha \in \{(,)\}^* \mid$ **existe** $n\,, \alpha = (^n)^n\}$ **no es regular.**

Por qué hay lenguajes no regulares

- Supongamos que $L_{()}$ es regular, entonces existe un AFD A que lo reconoce.
- A tiene una cantidad finita de estados, digamos m.
- Consideremos la palabra $(^{m+1})^{m+1}$: al leer el prefijo de 0s, visitamos uno de los estados más de una vez, digamos q_i.
- Es decir hay un ciclo



- El autómata no puede distinguir entre $(^{j+k+l}$ y $(^{j+2k+l}$.
- Por lo tanto, debe aceptar $(^{j+2k+l})^{j+k+l}$

Lema de bombeo (Pumping Lemma)

Teorema

Si L *es un lenguaje regular, entonces* existe *una constante* $n \in \mathbb{N}$ *tal que* para toda *cadena* $\alpha \in L$ *con* $|\alpha| \geqslant n$ existen *palabras* β, γ, δ *tales que* $\alpha = \beta\gamma\delta$

- $\gamma \neq \epsilon$
- $|\beta\gamma| \leqslant n$
- para toda $k \geqslant 0$, *la cadena* $\beta\gamma^k\delta \in L$.

Uso del pumping lemma

Los cuantificadores son importantes

- Asumimos que L es regular.
- Como es regular, existe n (no lo puedo elegir).
- Para toda palabra $\alpha \in L$ (la elijo en función de L y n),
- existen β_1, γ, β_2 tales que $\alpha = \beta_1 \gamma \beta_2$ y $|\beta_1 \gamma| \leqslant n$, $|\gamma| > 0$ (no puedo elegir a β_1, γ, β_2).
- para todo $k \geqslant 0$ (puedo elegir k en función de todo lo anterior), se da que $\beta_1 \gamma^k \beta_2 \in L$.

Resumen

Elijo una palabra que sé que pertenece L (usar n). Considero *TODAS* las formas de partir la palabra de acuerdo a las hipótesis, trabajo para concluir que hay una palabra que está en L pero que, por la definición del mismo, NO puede estar.

Uso del pumping lemma (sin contradicción)

El pumping lemma es útil para demostrar que ciertos lenguajes no son regulares, pero también lo podemos usar positivamente sin llegar a ninguna contradicción.

Por ejemplo, al aplicar el pumping lemma sobre:

- $L_1 = \emptyset$
- $L_2 = \{00, 11\}$
- $L_3 = \{01^l 0^k 1\}$

no podemos llegar a contradicción alguna porque:

- en L_1 no hay palabra para elegir;
- en L_2 no hay palabra para elegir si $n \neq 2$
- en L_2 el adversario podrá "partir" adecuadamente cualquier palabra α que elijamos.