Lenguajes regulares

Hay algoritmos para decidir si:

Sea L un lenguaje regular.

- $L \neq \emptyset$
  - Basta ver si existe cadena de longitud $< n$ *(n del lema de pumping)*.
  - Basta buscar los no terminales activos de la gramática que genera el lenguaje regular, luego ver si S pertenece al conjunto de los no terminales activos $(\in S)$.
- L es infinito
  - Basta ver si existe alguna cadena con longitud entre n y $2n-1$ *(n del lema de pumping)*.
- $L = \emptyset$
  - Basta ver si **no existe ninguna** cadena de longitud $< n$ *(n del lema de pumping)*.
  - Basta buscar los no terminales activos de la gramática que genera el lenguaje regular, luego ver si S **no** pertenece al conjunto de los no terminales activos $(\notin S)$.
- L es finito
  - Basta ver si **no existe ninguna** cadena con longitud entre n y $2n-1$ *(n del lema de pumping)*.

En la teórica vimos métodos efectivos para pasar de:

- **Autómata finito** a **gramática regular**.
- **Gramática regular** a **autómata finito**.
- **Autómata finito** a **expresión regular**.
- **Expresión regular** a **autómata finito**.

Por lo que, las 3 representaciones de lenguajes regulares son equivalentes.

Están cerrados por:

- Unión
- Intersección
- Complemento
- Concatenación
  Esto lo hace un álgebra de Boole.

- Reversa
- Diferencia $\rightarrow L_1 - L_2 = L_1 \cap \overline{L_2}$
- Diferencia simétrica $\rightarrow L_1 \oplus L_2 = (L_1 - L_2) \cup (L_2 - L_1)$
- Clausura de Kleene

Lenguajes libres de contexto (no determinísticos)

Hay algoritmos para decidir si:

Sea L un lenguaje libre de contexto.

- $L \neq \emptyset$
- L es infinito
- $L = \emptyset$
- L es finito
  Son los mismos que para lenguajes regulares, solo que el n del lema de pumping es el n lema de pumping para lenguajes libres de contexto.

En la teórica vimos métodos efectivos para pasar de:

- **Autómata finito** a **gramática regular**.
- **Gramática regular** a **autómata finito**.

NO hay algoritmos para decidir si:

Sean L, L_1, L_2 lenguajes libres de contexto

- L es regular
- Una gramática G que genera L es ambigua
- $L = \Sigma^*$
- $L_1 \subseteq L_2$ es libre de contexto
- $L_1 \cap L_2$ es libre de contexto

Están cerrados por:

- Unión
- Intersección con un lenguaje regular
- Concatenación
- Reversa
- Clausura de Kleene

NO están cerrados por:

- Intersección
- Complemento
- Diferencia

Lenguajes libres de contexto (determinísticos)

Hay algoritmos para decidir si:

Sean L, L_1, L_2 lenguajes libres de contexto, y sea R un lenguaje regular.

- $L \neq \emptyset$
- L es infinito
- $L = \emptyset$
- L es finito

  Idem no determinísticos.
- L es cofinito
- $L = \Sigma^*$
- $L_1 = L_2$
- L es regular
- $L = R$
- $L \subseteq R$

NO hay algoritmos para decidir si:

Sean L_1, L_2 lenguajes libres de contexto.

- $L_1 \subseteq L_2$
- $L_1 \cap L_2 = \emptyset$

Están cerrados por:

- Complemento
- Intersección con un lenguaje regular
- LR

  Concatenación de un lenguaje L libre de contexto con un lenguaje regular R.
- $pre(L)$

  Prefijos de un lenguaje L libre de contexto.
- $mín(L)$
- $máx(L)$

NO están cerrados por:

- Unión
- Intersección
- Concatenación
- Reversa
- Clausura de Kleene