

1.1. Teoría de la computación

Idea informal. Un *problema de decisión* recibe una instancia, escrita como un string, y responde *sí* o *no*. Como sólo importa en qué instancias la respuesta es *sí*, representamos el problema por el conjunto de sus *instancias positivas*. Ese conjunto de strings es un lenguaje.

1.2. Lenguajes formales

Definición (Alfabetos y strings). Un *alfabeto* Σ es un conjunto finito y no vacío de símbolos. Para cada $k \in \mathbb{N}$, un *string de largo k sobre Σ* es una secuencia

$$w = w_1 w_2 \cdots w_k, \quad w_i \in \Sigma,$$

y Σ^k denota el conjunto de todos esos strings. Existe un único string de largo 0, llamado *string vacío* y denotado por ε ; por lo tanto,

$$\Sigma^0 = \{\varepsilon\}.$$

El conjunto de todos los strings sobre Σ es

$$\Sigma^* = \bigcup_{k \geq 0} \Sigma^k.$$

Si $w \in \Sigma^*$, denotamos su largo por $|w|$.

Definición (Concatenación y potencias de strings). Sean $x = x_1 \cdots x_n$ e $y = y_1 \cdots y_m$ strings sobre un mismo alfabeto Σ . Su *concatenación* es el string

$$x \cdot y = xy = x_1 \cdots x_n y_1 \cdots y_m.$$

Las *potencias* de x se definen por

$$x^0 = \varepsilon \quad \text{y} \quad x^k = x^{k-1}x \quad \text{para todo } k \geq 1.$$

Definición (Lenguajes). Sea Σ un alfabeto. Un *lenguaje sobre Σ* es un subconjunto

$$L \subseteq \Sigma^*.$$

Definición (Operaciones booleanas de lenguajes). Sean $L_1, L_2 \subseteq \Sigma^*$ dos lenguajes sobre el

mismo alfabeto. Definimos

$$\begin{aligned} L_1 \cup L_2 &= \{w \in \Sigma^* : w \in L_1 \text{ o } w \in L_2\}, \\ L_1 \cap L_2 &= \{w \in \Sigma^* : w \in L_1 \text{ y } w \in L_2\}, \\ L_1 \setminus L_2 &= \{w \in \Sigma^* : w \in L_1 \text{ y } w \notin L_2\}, \\ L_1^c &= \Sigma^* \setminus L_1 = \{w \in \Sigma^* : w \notin L_1\}. \end{aligned}$$

En la última igualdad, el complemento siempre se toma con respecto a Σ^* .

Definición (Concatenación de lenguajes). Sean $L_1, L_2 \subseteq \Sigma^*$. La *concatenación* de L_1 y L_2 es el lenguaje

$$L_1 L_2 = \{xy : x \in L_1 \text{ e } y \in L_2\}.$$

Definición (Potencias y estrella de Kleene). Sea $L \subseteq \Sigma^*$ un lenguaje. Sus *potencias* se definen recursivamente por

$$L^0 = \{\varepsilon\} \quad \text{y} \quad L^k = L^{k-1}L \quad \text{para todo } k \geq 1.$$

La *estrella de Kleene* de L es

$$L^* = \bigcup_{k \geq 0} L^k.$$

Ejercicios

- Sea Σ un alfabeto y sean $A, B, C \subseteq \Sigma^*$ lenguajes. En cada inciso, justifique las igualdades a partir de las definiciones.

a) Demuestre que la concatenación de lenguajes es asociativa:

$$(AB)C = A(BC).$$

b) Demuestre que la concatenación distribuye por ambos lados sobre la unión:

$$A(B \cup C) = AB \cup AC \quad \text{y} \quad (A \cup B)C = AC \cup BC.$$

c) Demuestre que $\{\varepsilon\}$ es identidad y que $\emptyset$ es anulador para la concatenación; es decir,

$$\{\varepsilon\}A = A = A\{\varepsilon\} \quad \text{y} \quad \emptyset A = \emptyset = A\emptyset.$$

d) Deduzca que

$$A^* = \{\varepsilon\} \cup AA^* = \{\varepsilon\} \cup A^*A.$$

e) ¿Son las siguientes verdad?

$$AB = BA \quad \text{y} \quad (A \cup B)^* = A^* \cup B^*.$$

2. **Decisión, optimización y búsqueda.** Sea $G = (V, E)$ un grafo simple no dirigido. Un *matching* de G es un conjunto $M \subseteq E$ de aristas que no comparten extremos, y

$$\nu(G) = \max\{|M| : M \text{ es un matching de } G\}$$

denota la cardinalidad de un matching máximo.

- ¿Que problemas modela el problema de matching máximo? ¿Qué aplicaciones podría tener?
- Trate de describir el problema de decisión asociado al problema de optimización de matching máximo. (Respuesta al girar la hoja.)

$$\text{MATCHING} = \langle G, k \rangle : G \text{ es un grafo simple no dirigido, } k \in \mathbb{N} \wedge \exists M \subseteq E \text{ tal que } |M| = k$$

- Demuestre la monotonía de las respuestas respecto de k : si $\langle G, k \rangle$ es una instancia positiva, entonces $\langle G, k' \rangle$ también lo es para todo $0 \leq k' \leq k$. Formule también la afirmación equivalente para una instancia negativa.
- Use esa monotonía y el hecho de que

$$0 \leq \nu(G) \leq \left\lfloor \frac{|V|}{2} \right\rfloor$$

para calcular $\nu(G)$ mediante búsqueda binaria. Pruebe la correctitud del procedimiento y muestre que realiza $O(\log |V|)$ llamadas al algoritmo de decisión.

- Una vez conocido $r = \nu(G)$, queremos recuperar un matching de cardinalidad r . Inicialice $H \leftarrow G$ y $M \leftarrow \emptyset$. Mientras $r > 0$, elija una arista $e = uv$ de H y consulte si

$$\langle H - \{u, v\}, r - 1 \rangle \in \text{MATCHING},$$

donde $H - \{u, v\}$ es el grafo obtenido al eliminar u, v y todas sus aristas incidentes. Si la respuesta es positiva, agregue e a M , reemplace H por $H - \{u, v\}$ y disminuya r en uno. Si la respuesta es negativa, reemplace H por $H - e$ y mantenga r .

Demuestre que el procedimiento es correcto usando el invariante de que M es un matching, ningún extremo de una arista de M pertenece a H , existe en H un matching de tamaño al menos r y

$$|M| + r = \nu(G).$$

Muestre además que la reconstrucción realiza a lo más $|E|$ llamadas al algoritmo de decisión.

- Considere ahora un problema de maximización cuyo valor óptimo es entero y pertenece a un intervalo finito conocido $[a, b]$. Explique cómo recuperar el valor óptimo a partir de preguntas de umbral de la forma “¿existe una solución de valor al menos k ?” y cuántas preguntas son necesarias. ¿Cómo cambia la pregunta de umbral para un problema de minimización?
- Encontrar el valor óptimo no entrega necesariamente una solución que lo alcance. Explique qué propiedad adicional de *autorreducibilidad* usa el procedimiento del inciso c) para reconstruir un objeto óptimo. Ilustre la misma idea para satisfacibilidad: dada una fórmula satisfacible, fije sus variables una por una consultando si la fórmula restringida sigue siendo satisfacible. Justifique por qué el proceso termina con una asignación satisfactoria.

3. **Ecuaciones lineales de lenguajes.** Sea Σ un alfabeto y sean $A, B \subseteq \Sigma^*$ lenguajes fijos. Considere la ecuación en la incógnita $X \subseteq \Sigma^*$

$$X = AX \cup B.$$

- a) Demuestre que A^*B es una solución de la ecuación. Demuestre además que, si X es cualquier solución, entonces

$$A^*B \subseteq X.$$

En otras palabras, A^*B es la menor solución respecto de la inclusión.

- b) Suponga que $\varepsilon \notin A$. Demuestre que la ecuación tiene una única solución. Para ello, pruebe mediante inducción fuerte sobre $|w|$ que toda solución X está contenida en A^*B .
- c) Muestre que la hipótesis $\varepsilon \notin A$ no puede omitirse en general. Use $\Sigma = \{0\}$, $A = \{\varepsilon\}$ y $B = \{0\}$ para exhibir dos soluciones distintas.
- d) El resultado de los incisos anteriores se conoce como el *lema de Arden*. Enúncielo en una sola frase, distinguiendo claramente la existencia de una solución mínima de la condición que garantiza unicidad.
- e) Resuelva por sustitución el siguiente sistema de ecuaciones de lenguajes sobre $\Sigma = \{0, 1\}$:

$$X = \{0\}X \cup \{1\}Y \cup \{\varepsilon\}, \quad Y = \{0\}X \cup \{1\}.$$

Entregue expresiones para X e Y usando únicamente lenguajes finitos, unión, concatenación y estrella de Kleene. Demuestre que la solución es única y describa verbalmente los strings que pertenecen a X .