Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2213 - Lógica para Ciencias de la Computación
Profesor: Miguel Romero
Ayudantes: Caetano Borges, Valentina Juri

Ayudantía 4
Enumeradores, decibilidad y computabilidad

10 de abril de 2026

Problema 1 Demuestre que todo lenguaje finito $L = \{w_1, \ldots, w_k\}$ es decidible.

Solución Consideremos los lenguajes $L_i = \{w_i\}$. Es claro que L_i es decidible, ya que podemos hacer una MT que revisa si la palabra de entrada es w_i y si lo es acepta, si no rechaza, por lo que su ejecución termina para toda palabra de entrada. Además, sabemos que si A y B son lenguajes decibidbles, entonces $A \cup B$ también es decidible. Aplicando inducción con este último hecho, tenemos que $L = \bigcup_{i=i}^{k} L_i$, esto es, la unión finita de lenguajes decidibles, con lo que también es decidible.

Problema 2 Sean Σ_1 y Σ_2 dos alfabetos finitos. Decimos que una función $h : \Sigma_1^* \to \Sigma_2^*$ es computable si existe una MT (modelo determinista con 1 cinta) $M = (Q, \Sigma_1, \Gamma, q_0, q_f, \delta)$, donde $\Sigma_2 \subseteq \Gamma$, tal que para toda entrada $w \in \Sigma_1^*$, la ejecución $\rho = C_0, \ldots, C_m$ de M sobre w es finita y $C_m = \vdash q_f h(w)$. Note que esta es la misma definición de computar funciones de la tarea 1, con la salvedad que la MT tiene una sola cinta y las palabras en el dominio y rango de la función pueden estar sobre alfabetos distintos.

1. Sean Σ_1 y Σ_2 dos alfabetos finitos. Decimos que $h : \Sigma_1^* \to \Sigma_2^*$ es un homomorfismo si:
   - $h(\varepsilon) = \varepsilon$.
   - Para todo par de palabras $u, v \in \Sigma_1^*$, se cumple que $h(uv) = h(u)h(v)$.

   Demuestre que todo homomorfismo es computable.
2. Sean $\Sigma_1, \Sigma_2, \Sigma_3$ alfabetos finitos. Demuestre que si $f : \Sigma_1^* \to \Sigma_2^*$ y $g : \Sigma_2^* \to \Sigma_3^*$ son computables, entonces la composición $g \circ f$ es computable.

Solución

1. Sea h un homomorfismo. Para $w \in \Sigma_1^*, w = a_1 \ldots a_k$ tenemos que $h(w) = h(a_1 \ldots a_k) = h(a_1 \ldots a_{k-1})h(a_k) = h(a_1) \ldots h(a_k)$. Como el alfabeto Σ_1^* es finito, es claro que hay una MT M que computa h para todas las entradas de largo 1 (ya que hay un número finito de ellas, y sus imágenes son siempre finitas). Más formalmente, podemos definir

$$h' \colon \Sigma_1 \to \Sigma_2^* \text{ tal que } h'(a) = h(a)$$

Como el dominio de h' es finito, y toda imagen tiene largo finito, existe una MT M que computa h'. Luego, podemos construir una MT S que ejecute M sobre cada símbolo de la palabra de entrada y luego concatene los resultados. S computa h.

2. La composición $g \circ f$ va de Σ_1^* a Σ_3^*. Como f y g son computables, existen MTs F y G que las computan respectivamente. Podemos construir una MT M que

   a) Ejecuta F sobre la palabra de entrada que está en Σ_1^*. Como F computa f, esta ejecución siempre termina, y además el resultado está en Σ_2^*.

   b) Luego, ejecuta G sobre el resultado obtenido de ejecutar F. Como ese resultado está en Σ_2^*, y G computa g, esta ejecución siempre termina, y el resultado está en Σ_3^*.

   Por construcción, el resultado es $g \circ f$.

Problema 3 Demuestre que L es decidible si y solo si existe una MT enumeradora E que enumera L en orden lexicográfico (i.e. si E imprime w' después de w se tiene que w' es lexicográficamente mayor que w).

Dado un alfabeto finito $A = \{a_1, a_2, \ldots, a_k\}$, donde asumimos un orden $a_1 < a_2 < \ldots < a_k$; $w' = w'_1 \ldots w'_p$ es lexicográficamente mayor que $w = w_1 \ldots w_q$ si $p > q$, o $p = q$ y $w'_i > w_i$ donde $i \in \{1, \ldots, p\}$ es la primera posición donde $w'_i \neq w_i$. Por ejemplo, si $A = \{a, b, c, d\}$, donde $a < b < c < d$, las siguientes palabras están en orden lexicográfico, de menor a mayor: $\epsilon, a, b, d, aa, ab, bc, aba, bca, bda, bdd, cca, daa, aaaa$.

Solución Sea $L \subseteq \Sigma^*$. Existen dos posibilidades: que L sea finito o infinito. Si L es finito, es trivialmente decidible (problema 1), y además es claro que existe un enumerador que lo enumera en orden lexicográfico. Para el caso de L infinito demostramos las dos direcciones:

- $\Rightarrow$: Como L es decidible, existe una MT M tal que $L = L(M)$ y que siempre se detiene. Construimos un enumerador E que itera sobre todas las palabras de Σ^* en orden lexicográfico, y ejecuta M sobre cada una; si M acepta, entonces E enumera la palabra de entrada respectiva. Si M rechaza, entonces E no la enumera. Como M siempre se detiene, la ejecución sobre cada palabra siempre termina, y además para toda palabra existe un número finito de palabras lexicográficamente menores a ella, con lo que siempre habrá un número finito de pasos antes de que se realice la ejecución de cualquier palabra. Esto implica que M será ejecutada sobre toda palabra $w \in \Sigma^*$. Con ello, E enumera todas las palabras aceptadas por M, esto es, todas las palabras pertenecientes a L, y lo hace en orden lexicográfico.

- $\Leftarrow$: Sea E un enumerador que enumera L en orden lexicográfico. Podemos construir una MT M de tres cintas que simule E en la segunda y tercera cinta, usando la segunda como cinta de trabajo y la tercera como cinta de enumeración. Cada vez que E enumere una palabra en la cinta de enumeración, M compara la palabra de entrada w que está en la cinta 1 con la enumerada. Si son iguales, acepta. Si son distintas, M revisa si la

palabra enumerada es menor o igual al de w. En tal caso, continúa con la siguiente iteración. Si es lexicográficamente mayor a w, M rechaza.

Como L es infinito, y como para toda palabra w hay un número finito de palabras lexicográficamente menores a w, siempre habrá una palabra en L que sea lexicográficamente mayor que w, por lo que si la palabra de entrada no está en L tenemos la certeza de que eventualmente M terminará rechazando. Por otra parte, si $w \in L$ entonces E la enumera eventualmente, con lo que M acepta. Con ello, $L(M) = L$ y M siempre termina, con lo que M decide L.

Problema 4 Demuestre que todo lenguaje infinito recursivamente enumerable L tiene un subconjunto $L' \subseteq L$ tal que L' es infinito y decidible.

Solución Como L es recursivamente enumerable, existe un enumerador E que enumera a L. Luego, podemos crear un enumerador E' que hace lo siguiente:

1. Ejecuta E
2. Cuando E imprime una palabra, checkea si esta es lexicográficamente mayor a la anterior palabra impresa. Si lo es, E' la imprime.

Claramente el lenguaje que se va escribiendo es un subconjunto de L y está en orden lexicográfico. Además, por un argumento similar al del ejercicio anterior, como L es infinito y, para toda palabra w existe un número finito de palabras lexicográficamente menores a w, necesariamente siempre habrá una palabra lexicográficamente mayor a w, con lo que el lenguaje L' enumerado por E' es infinito y por el ejercicio anterior es decidible.

Problema 5 Sea un lenguaje L sobre alfabeto Σ. Definimos el lenguaje prefijo de L como:

$$\text{prefix}(L) = \{w \in \Sigma^* \mid \text{ existe } z \in \Sigma^* \text{ tal que } wz \in L\}$$

Demuestre que si L es recursivamente enumerable, entonces prefix(L) también lo es.

Solución Como L es recursivamente enumerable, existe un enumerador E que lo enumera. Podemos construir una MT M que ejecuta E y, para cada palabra z imprimida, revisa si w es prefijo de v. Si lo es, M acepta.

Sea $w \in \text{prefix}(L)$. Tenemos que necesariamente existe $z \in L$ tal que w es prefijo de z. Como $L = L(E)$, E eventualmente imprime z. Cuando lo haga, M verá que w es prefijo y aceptará w. Con ello, prefix$(L) \subseteq L(M)$. La dirección opuesta sigue un argumento prácticamente idéntico. Concluímos que M reconoce prefix(L), con lo que prefix(L) es recursivamente enumerable.