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 5

Reducciones y clases R, RE y co-RE

24 de abril de 2026

Importante: En la solución de los problemas no se argumenta porqué las reducciones planteadas son efectivamente computables, pero lo son. En ese sentido, nótese que si es posible definir una máquina de Turing, es posible computar su codificación siguiendo el procedimiento discutido en clases, de modo que la codificación de las MTs construidas e incluso MTs fijas utilizadas en las reducciones (M_{all}, M_{loop}), puede ser calculada sin problema, de modo que las salidas de las reducciones pueden ser computadas, realizando los ajustes necesarios (por ejemplo, shifts).

Problema 1. Sea $k \geq 1$ un número natural fijo. Definimos el siguiente lenguaje sobre $\{0,1\}$:

$$\text{Steps}_{\geq k} = \{\langle M, w\rangle |\ M \text{ es una MT, } w \in \{0,1\}^*, M \text{ se detiene con } w \text{ en } k \text{ o más pasos}\}$$

a) Demuestre que $\text{Steps}_{\geq k}$ está en RE.

Solución: Para esto debemos demostrar que $\text{Steps}_{\geq k}$ es un lenguaje recursivamente enumerable, es decir, que existe una máquina N que acepta $\text{Steps}_{\geq k}$ y no necesariamente se detiene con todo input. Se propone el siguiente pseudocódigo para la MT N, considerando que recibe como entrada un par $\langle M, w\rangle$ donde M es una MT y $w \in \{0,1\}^*$ (esta definición puede expandirse a cualquier entrada en $\{0,1\}^*$ agregando una verificación adicional que rechace las entradas que no son de la forma $\langle M, w\rangle$).

---

1) Ejecutar M sobre w.
2) Si M se detiene en menos de k pasos, rechazar.
3) Si M se detiene posteriormente, aceptar.

Véase que si $\langle M, w\rangle \in \text{Steps}_{\geq k}$, la ejecución de M sobre w es finita y cuenta con k o más pasos, por lo que una vez terminado el paso (1) N acepta el par $\langle M, w\rangle$ en el paso (3). Por otro lado, si $\langle M, w\rangle \notin \text{Steps}_{\geq k}$ debemos considerar 2 casos. En primera instancia, es posible que la ejecución de M sobre w se detenga antes de los k pasos, en este caso la ejecución en (1) termina, y la entrada se rechaza en el paso (2). En el segundo caso M puede no detenerse al ejecutarse sobre w, de este modo, la MT N se queda estancada en la ejecución realizada en el paso (1) y tampoco se detiene, por lo que esta entrada también es rechazada.

b) Demuestre que $\text{Steps}_{\geq k}$ es indecidible.

Solución: Para esto se realiza una reducción desde el lenguaje Halting, que es indecidible, la cual dada una entrada $\langle M, w\rangle$ computa $f(\langle M, w\rangle) = \langle S_M, w\rangle$, donde S_M corresponde a la máquina de Turing que para toda entrada $u \in \{0,1\}^*$, realiza lo siguiente:

1. Avanza k posiciones a la derecha y luego retrocede k posiciones a la izquierda sin modificar el contenido de la cinta.
2. Ejecuta M sobre u.
3. Si M se detiene, acepta.

Esta construcción respeta la pertenencia entre lenguajes, puesto que:

- Si $\langle M, w\rangle \in$ Halting: M se detiene al ejecutarse sobre w en $n \in \mathbb{N}$ pasos. Así, en el paso (1), la máquina S_M realiza un total de $2k$ pasos, independientemente del valor de u, en los que no modifica la entrada. Como S_M realiza la misma cantidad de pasos hacia la derecha que hacia la izquierda, al final del paso (1) el cabezal se encuentra en la posición 1 de la cinta, siendo esta su posición inicial, por lo que la ejecución de M sobre $u = w$ se realiza con normalidad en n pasos y el total de pasos ejecutados por S_M sobre w es de $2k + n$. De modo que, $\langle S_M, w\rangle \in \text{Steps}_{\geq k}$.

  Nótese que se podría haber utilizado una cota más ajustada haciendo que en el paso (1) S_M se moviera $\frac{k}{2}$ posiciones a la derecha e izquierda (k pasos totales), en caso de ser par o $\frac{k+1}{2}$ posiciones de ser impar ($k + 1$ pasos totales). Sin embargo, dado que solo se pide que la cantidad de pasos realizados por S_M sobre w sea mayor o igual a k, se prefiere hacer k pasos por simplicidad.
- Si $\langle M, w\rangle \notin$ Halting: M no se detiene al ejecutarse sobre w, por lo que al ejecutar M sobre $u = w$ en el paso (2) S_M tampoco se detiene y entra en un loop. De este modo, $\langle S_M, w\rangle \notin \text{Steps}_{\geq k}$

Como la reducción planteada es computable y respeta la pertenencia entre lenguajes, concluimos que Halting se reduce a $\text{Steps}_{\geq k}$. Finalmente, dado que Halting es un lenguaje indecidible queda demostrado, por propiedades de las reducciones, que $\text{Steps}_{\geq k}$ también es indecidible.

De lo anterior podemos concluir que $\text{Steps}_{\geq k} \notin$ co-RE, dado que un lenguaje L y su complemento $\overline{L}$ son recursivamente enumerables sí solo sí L es decidible. Así, como $\text{Steps}_{\geq k} \notin$ R, pero $\text{Steps}_{\geq k} \in$ RE, necesariamente debe cumplirse que $\overline{\text{Steps}_{\geq k}} \notin$ RE, es decir que $\text{Steps}_{\geq k} \notin$ co-RE. Véase que intuitivamente esto tiene sentido, puesto que $\overline{\text{Steps}_{\geq k}}$ contiene todos los pares $\langle M, w\rangle$ con M una máquina de Turing y $w \in \{0,1\}^*$, tales que al ejecutarse M sobre w, M se detiene en menos de k pasos o no se detiene. Este lenguaje es similar a $\overline{\text{Halting}}$, que contiene a todos los pares $\langle M, w\rangle$ con M MT y $w \in \{0,1\}^*$ tales que M no se detiene con w y también corresponde a un lenguaje no recursivamente enumerable.

Problema 2. Definimos el siguiente lenguaje sobre $\{0, 1\}$:

$$\text{Write} = \{\langle M\rangle |\ M \text{ es una MT, en la ejecución de } M \text{ con } \epsilon \text{ en algún momento se escribe 0 en la posición 1}\}$$

a) Demuestre que Write está en RE.

Solución: Para esto debemos demostrar que Write es un lenguaje recursivamente enumerable, es decir, que existe una máquina N que acepta Write y no necesariamente se detiene con todo input. Se propone el siguiente pseudocódigo para la MT N, considerando que recibe como entrada $\langle M\rangle$ con M una MT:

1) Ejecutar M sobre ε.
2) Si M escribe 0 en la posición 1, **aceptar**.
3) Si M se detiene, **rechazar**.

Véase que si $\langle M\rangle \in$ Write, en algún momento de la ejecución de M sobre ε M escribe un 0 en la posición 1, así la entrada es aceptada en el paso (2). En caso contrario, M es rechazada al detenerse sin realizar esta escritura en el paso (3) o N entra en un loop en el paso (1) si la ejecución de M sobre w es infinita.

b) Demuestre que Write es indecidible.

Solución: Para esto se realiza una reducción desde el lenguaje Halting, que es indecidible. La reducción dada una entrada $\langle M, w\rangle$ computa $f(\langle M, w\rangle) = \langle W_{M,w}\rangle$, donde $W_{M,w}$ corresponde a la máquina de Turing que para toda entrada $u \in \{0,1\}^*$, realiza lo siguiente:

---

1. Borrar u de la cinta.
2. Escribir $\vdash w_1 \cdots w_n$ entre posiciones 2 y $n+2$ de la cinta, de modo que el contenido de la cinta sea: $\vdash \mathrm{B} \vdash w_1 \cdots w_n$
3. Llevar la cabeza lectora a la posición 3.
4. Ejecutar M sobre w.
5. Si M se detiene, escribir un 0 en la posición 1 y aceptar.

Esta construcción es correcta, puesto que:

- Si $\langle M, w\rangle \in$ Halting: M se detiene al ejecutarse sobre w, por lo que una vez terminado el paso (4), $W_{M,w}$ escribe un 0 en la posición 1, y $\langle W_{M,w}\rangle \in$ Write.
- Si $\langle M, w\rangle \notin$ Halting: M no se detiene al ejecutarse sobre w, por lo que $W_{M,w}$ entra en un loop en el paso (4), de modo que no logra llegar al paso (5) que modifica la posición 1 de la cinta. Ahora bien, como la ejecución de M sobre w se realiza desde la posición 2 en adelante, sabemos que esta no altera la posición 1 de la cinta, puesto que para efectos de la ejecución la posición 1 de la cinta es la posición 3 (la posición 2 tiene el símbolo de inicio). De este modo, nuna se escribe un 0 en la posición 1 y $W_{M,w}$ y $\langle W_{M,w}\rangle \notin$ Write.

Como la reducción planteada es computable y respeta la pertenencia entre lenguajes, concluimos que Halting se reduce a Write. Finalmente, dado que Halting es un lenguaje indecidible queda demostrado, por propiedades de las reducciones, que Write también es indecidible.

De lo anterior podemos concluir que Write $\notin$ co-RE, dado que un lenguaje L y su complemento $\overline{L}$ son recursivamente enumerables sí solo sí L es decidible. Así, como Write $\notin$ R, pero Write $\in$ RE, necesariamente debe cumplirse que $\overline{\text{Write}} \notin$ RE, es decir que Write $\notin$ co-RE.

Problema 3. Definimos el siguiente lenguaje sobre $\{0,1\}$:

$$\text{EQUIV} = \{\langle M_1, M_2\rangle |\ M_1 \text{ y } M_2 \text{ son MTs y } L(M_1) = L(M_2)\}$$

a) Demuestre que EQUIV no pertenece a RE.

Solución: Para esto se realiza una reducción desde el lenguaje All, un lenguaje que no es recursivamente enumerable. Esta reducción dada una entrada $\langle M\rangle$ computa $f(\langle M\rangle) = \langle M, M_{all}\rangle$, donde M_{all} corresponde a una máquina de Turing arbitraria que acepta todas las palabras posibles en $\{0,1\}^*$.

Esta construcción respeta la pertenencia entre lenguajes, puesto que:

- Si $\langle M\rangle \in$ All: El lenguaje aceptado por M es $L(M) = \{0,1\}^*$, el cual, por definición, corresponde al mismo lenguaje aceptado por M_{all}. Así, $L(M) = L(M_{all})$ y $\langle M, M_{all}\rangle \in$ EQUIV
- Si $\langle M\rangle \notin$ All: Existe al menos una palabra $u \in \{0,1\}^*$ que es rechazada por M, es decir, $u \notin L(M)$. Sin embargo, por la definición de M_{all} sabemos que esta MT acepta todas las palabras en $\{0,1\}^*$, por lo que $u \in L(M_{all})$, de este modo los lenguajes aceptados por las máquinas no son iguales y $\langle M, M_{all}\rangle \notin$ EQUIV.

Como la reducción planteada es computable y respeta la pertenencia entre lenguajes, concluimos que All se reduce a EQUIV. Finalmente, dado que All no pertenece a la clase RE concluimos por propiedades de las reducciones que EQUIV tampoco pertenece a RE.

b) Demuestre que $\overline{\text{EQUIV}}$ no pertenece a RE.

Solución: Nótese que, por propiedades de reducciones tenemos que si $K \leq_m L$, entonces $\overline{K} \leq_m \overline{L}$, puesto que al respetar una reducción la pertenencia entre lenguajes, también respeta la pertenencia entre sus complementos (recordar que una reducción f cumple que $w \in K \leftrightarrow f(w) \in L$, es decir, que $w \in K \rightarrow f(w) \in L$ y $w \notin K \rightarrow f(w) \notin L$), por lo que la misma reducción puede ser utilizada entre los complementos de K y L.
En base a esto, de la reducción de All $\leq_m$ EQUIV construida en el apartado anterior se puede concluir que $\overline{\text{All}} \leq_m \overline{\text{EQUIV}}$, y como $\overline{\text{All}}$ también es un lenguaje que no pertenece a la clase RE, podemos concluir por propiedades de reducciones que $\overline{\text{EQUIV}}$ no es un lenguaje recursivamente enumerable.
Así, podemos notar que los lenguajes EQUIV y $\overline{\text{EQUIV}}$ no pertenecen a la clase RE ni co-RE.

Problema 4. Definimos el siguiente lenguaje sobre $\{0, 1\}$:

$$\text{D} = \{\langle M_1, w_1, M_2, w_2\rangle |\ M_1 \text{ se detiene con } w_1,\ M_2 \text{ no se detiene con } w_2\}$$

a) Demuestre que D no pertenece a RE.

Solución: Para esto se realiza una reducción desde $\overline{\text{Halting}}$, un lenguaje que no es recursivamente enumerable. La reducción calcula, dada una entrada $\langle M, w\rangle$ $f(\langle M, w\rangle) = \langle M_{all}, w, M, w\rangle$, donde M_{all} es una máquina de Turing que acepta todas las palabras en $\{0, 1\}^*$.
Nótese que la pertenencia del resultado de la reducción ($\langle M_{all}, w, M, w\rangle$) a D, depende únicamente del comportamiento de la ejecución de M sobre w, puesto que M_{all} acepta y en particular se detiene, con todas las palabras posibles en $\{0, 1\}^*$, entre las que se encuentra w. Así, M_1 siempre se detiene con w_1, por lo que $\langle M_{all}, w, M, w\rangle$ pertenecerá o no a D, dependiendo solo del comportamiento de $\langle M, w\rangle$. Demostremos así que la reducción es correcta:

- Si $\langle M, w\rangle \in \overline{\text{Halting}}$: M no se detiene con w, por lo que en $\langle M_{all}, w, M, w\rangle$ se tiene que M_2 no se detiene con w_2 y como M_1 siempre se detiene con w_1, $\langle M_{all}, w, M, w\rangle \in D$
- Si $\langle M, w\rangle \notin \overline{\text{Halting}}$: M se detiene con w, por lo que en $\langle M_{all}, w, M, w\rangle$ se tiene que M_2 se detiene con w_2 y $\langle M_{all}, w, M, w\rangle \notin D$

Como la reducción planteada es correcta y computable, concluimos que $\overline{\text{Halting}}$ se reduce a D. Dado que $\overline{\text{Halting}}$ es un lenguaje que no pertenece a la clase RE y por propiedades de reducciones concluimos que D tampoco pertenece a RE.

b) Demuestre que $\overline{\text{D}}$ no pertenece a RE.

Solución: Nótese que en este caso, no nos sirve utilizar la propiedad utilizada en 3b (una reducción entre dos lenguajes, también es una reducción entre sus complementos), puesto que, con esta concluimos que Halting $\leq_m \overline{D}$, sin embargo, como Halting es un lenguaje indecidible, pero recursivamente enumerable, esta reducción solo nos permite demostrar que $\overline{D}$ es indecidible y no que está fuera de la clase RE.
Para demostrar lo pedido se realiza nuevamente una reducción desde $\overline{\text{Halting}}$, la reducción calcula, dada una entrada $\langle M, w\rangle$ $f(\langle M, w\rangle) = \langle M, w, M_{loop}, w\rangle$ siendo M_{loop} una máquina que entra en loop infinito con toda entrada en $\{0, 1\}^*$

Nótese que la pertenencia del resultado de la reducción ($\langle M, w, M_{loop}, w\rangle$) a $\overline{D}$, depende únicamente del comportamiento de la ejecución de M sobre w, puesto que M_{loop} no se detiene con ninguna entrada, en particular, no se detiene con w. Así, M_2 nunca se detiene con w_2, por lo que $\langle M, w, M_{loop}, w\rangle$ pertenecerá o no a $\overline{D}$, dependiendo solo del comportamiento de $\langle M, w\rangle$. Demostremos así que la reducción es correcta:

- Si $\langle M, w\rangle \in \overline{\text{Halting}}$: M no se detiene con w, por lo que en $\langle M, w, M_{loop}, w\rangle$ se tiene que M_1 no se detiene con w_1 por lo que $\langle M, w, M_{loop}, w\rangle \notin D$, es decir, $\langle M, w, M_{loop}, w\rangle \in \overline{D}$
- Si $\langle M, w\rangle \notin \overline{\text{Halting}}$: M se detiene con w, por lo que en $\langle M, w, M_{loop}, w\rangle$ se tiene que M_1 se detiene con w_1 y $\langle M, w, M_{loop}, w\rangle \in D$, es decir, $\langle M, w, M_{loop}, w\rangle \notin \overline{D}$

Como la reducción planteada es correcta y computable, concluimos que $\overline{\text{Halting}}$ se reduce a D. Dado que $\overline{\text{Halting}}$ es un lenguaje que no pertenece a la clase RE y por propiedades de reducciones concluimos que $\overline{D}$ tampoco pertenece a RE.

Así, podemos notar que los lenguajes D y $\overline{D}$ no pertenecen a la clase RE ni co-RE.