Práctica 5

1. Sean L_1, L_2, L_3, L_4 cuatro lenguajes definidos sobre $\{0,1\}^*$

$$L_1 = \{0^n 1 \mid n \geq 0\}$$

$$L_2 = \{1^n 0 \mid n \geq 0\}$$

$$L_3 = \{\lambda\}$$

$$L_4 = \{1^n 0 \mid n > 0\}$$

a. Demuestre que existe una reducción $(L_1 \, \alpha \, L_2)$

1. **Análisis de los lenguajes**:
   i. Las primeras cadenas de L_1 son: $1, 01, 001, 0001, 00001, \ldots$
   ii. Las primeras cadenas de L_2 son: $0, 10, 110, 1110, 11110, \ldots$
   iii. λ no pertenece a ninguno de los dos lenguajes porque por más que $n = 0$, siempre se tiene o un 1 o un 0 en la cadena.
   iv. Se puede ver claramente que L_2 simplemente invierte todos los bits de L_1.
2. **Definición de la reducción**:
   i. $L_1 \, \alpha \, L_2$ si existe una función total computable $f : \Sigma^* \to \Sigma^*$ tal que a cada palabra $w \in L_1$ le corresponde una palabra $f(w) \in L_2$, y a cada palabra $z \notin L_1$ le corresponde una palabra $f(z) \notin L_2$.
3. **Construcción de la M_f**:
   i. Podemos definir una MT M_f de cómputo que compute la función f y que siempre se detenga:
   ii. $M_f = \langle Q, \Sigma, \Gamma, \delta, q_0, q_d \rangle$ donde:
      - $Q = \{q_0\}$
      - $\Sigma = \{0, 1\}$
      - $\Gamma = \{0, 1, B\}$
      - $\delta(q_0, 0) = (q_0, 1, D)$

- $\delta(q_0, 1) = (q_0, 0, D)$
- $\delta(q_0, B) = (q_d, B, S)$

iii. La MT M_f lee la cadena de entrada y va reemplazando cada 0 por un 1 y cada 1 por un 0. Cuando llega al símbolo de blanco, se detiene.

4. **Verificación de las condiciones de la reducción**:
   i. M_f es total?
      a. Sí, porque está definida para todos los elementos de Σ^*. ✅
   ii. M_f es computable?
      a. Sí, porque siempre se detiene, debido a que el alfabeto es finito, la cadena de entrada es finita, y no hay bucles infinitos en ninguna transición. ✅
   iii. Para cada palabra $w \in L_1$, $f(w) \in L_2$?
      a. Si la cadena de entrada es de la forma 0^n1, que pertenece a L_1, al aplicar $f(0^n1)$ se obtiene 1^n0, que pertenece a L_2. ✅
         a. Por ejemplo, si $w = 001 \in L_1$, al aplicar $f(001)$ se obtiene $110 \in L_2$.
   iv. Para cada palabra $z \notin L_1$, $f(z) \notin L_2$?
      a. Si la cadena de entrada es λ, que no pertenece a L_1, al aplicar $f(\lambda)$ se obtiene λ, que no pertenece a L_2. ✅
      b. Si la cadena de entrada no es de la forma 0^n1, que no pertenece a L_1, al aplicar f a esa cadena, se obtiene una cadena que no es de la forma 1^n0, por lo que $f(z) \notin L_2$. ✅
         a. Por ejemplo, si $z = 101 \notin L_1$, al aplicar $f(101)$ se obtiene $010 \notin L_2$.

Por lo tanto, $L_1 \ \alpha \ L_2$.

b. Demuestre que existe una reducción $(L_1 \ \alpha \ L_3)$

1. **Análisis de los lenguajes**:
   i. Las primeras cadenas de L_1 son: $1, 01, 001, 0001, 00001, \ldots$
   ii. La única cadena de L_3 es λ, la cadena vacía.
   iii. λ no pertenece a L_1 porque por más que $n = 0$, siempre se tiene un 1 en la cadena.
   iv. Se puede ver claramente que L_3 simplemente convierte cualquier cadena de L_1 en la cadena vacía.
2. **Definición de la reducción**:
   i. $L_1 \ \alpha \ L_3$ si existe una función total computable $f : \Sigma^* \to \Sigma^*$ tal que a cada palabra $w \in L_1$ le corresponde una palabra $f(w) \in L_3$, y a cada palabra $z \notin L_1$ le corresponde una palabra $f(z) \notin L_3$.
   ii. Es decir, para cada cadena válida de L_1 (de la forma 0^n1), lo que la función debe hacer es dejar la cadena vacía en la cinta, y para cada cadena inválida de L_1 (de cualquier otra forma), debe dejar algo en la cinta que NO sea la cadena vacía.
3. **Construcción de la** M_f:

i. Podemos definir una MT M_f de cómputo que compute la función f y que siempre se detenga:

ii. $M_f = \langle Q, \Sigma, \Gamma, \delta, q_0, q_d \rangle$ donde:

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, B\}$
- $\delta(q_0, B) = (q_d, 0, S)$
- $\delta(q_0, 0) = (q_0, 0, D)$
- $\delta(q_0, 1) = (q_1, 1, D)$
- $\delta(q_1, B) = (q_2, B, I)$
- $\delta(q_1, 0) = (q_d, 0, S)$
- $\delta(q_1, 1) = (q_d, 1, S)$
- $\delta(q_2, 0) = (q_2, B, I)$
- $\delta(q_2, 1) = (q_2, B, I)$
- $\delta(q_2, B) = (q_d, B, S)$

iii. Lo que hace esta MT es:

a. Si la cadena de entrada es λ, deja un 0 en la cinta y termina.

b. Si la cadena empieza con 0, se mueve hacia la derecha hasta encontrar un 1 o un blanco.

a. Si encuentra un 1, va al paso 3.

b. Si encuentra blanco, deja un cero y termina.

c. Si la cadena empieza con 1, se asegura que el próximo símbolo sea blanco.

a. Si es blanco, borra toda la cinta.

b. Si es un 0 o un 1, deja la cadena tal cual y termina.

4. **Verificación de las condiciones de la reducción**:

i. M_f es total?

a. Sí, porque está definida para todos los elementos de Σ^*. ✅

ii. M_f es computable?

a. Sí, porque siempre se detiene, debido a que el alfabeto es finito, la cadena de entrada es finita, y no hay bucles infinitos en ninguna transición. ✅

iii. Para cada palabra $w \in L_1$, $f(w) \in L_3$?

a. Si la cadena de entrada es de la forma 0^n1, que pertenece a L_1, al aplicar $f(0^n1)$ se obtiene λ, que pertenece a L_3. ✅

a. Por ejemplo, si $w = 001 \in L_1$, al aplicar $f(001)$ se obtiene $\lambda \in L_3$.

iv. Para cada palabra $z \notin L_1$, $f(z) \notin L_3$?

a. Si la cadena de entrada es λ, que no pertenece a L_1, al aplicar $f(\lambda)$ se obtiene 0, que no pertenece a L_3. ✅

b. Si la cadena de entrada no es de la forma 0^n1, que no pertenece a L_1, al aplicar f a esa cadena, se obtiene una cadena que es igual a w o w concatenada a un cero al final. En ambos casos, $f(z) \notin L_3$. ✅

a. Por ejemplo, si $z = 101 \notin L_1$, al aplicar $f(101)$ se obtiene $101 \notin L_3$.

Por lo tanto, $L_1 \ \alpha \ L_3$.

c. Demuestre que existe una reducción $(L_1 \ \alpha \ L_4)$

1. **Análisis de los lenguajes**:
   i. Las primeras cadenas de L_1 son: $1, 01, 001, 0001, 00001, \ldots$
   ii. Las primeras cadenas de L_4 son: $10, 110, 1110, 11110, \ldots$
   iii. λ no pertenece a ninguno de los dos lenguajes.
   iv. Se puede ver claramente que la transformación que hay que hacer para pasar de L_1 a L_4 es, primero invertir todos los bits uno por uno yendo hacia la derecha, y al llegar al símbolo blanco, volver al inicio de la cadena y agregar un uno a la izquierda de su comienzo. Por ejemplo, 001 se transforma en 110 al invertir todos sus bits, y luego le agregamos un uno al principio, 1110, por lo que 001 se convierte en 1110, que es $\in L_4$.
2. **Definición de la reducción**:
   i. $L_1 \ \alpha \ L_4$ si existe una función total computable $f : \Sigma^* \to \Sigma^*$ tal que a cada palabra $w \in L_1$ le corresponde una palabra $f(w) \in L_4$, y a cada palabra $z \notin L_1$ le corresponde una palabra $f(z) \notin L_4$.
3. **Construcción de la** M_f:
   i. Podemos definir una MT M_f de cómputo que compute la función f y que siempre se detenga:
   ii. $M_f = \langle Q, \Sigma, \Gamma, \delta, q_0, q_d \rangle$ donde:
      - $Q = \{q_0, q_1, q_2\}$
      - $\Sigma = \{0, 1\}$
      - $\Gamma = \{0, 1, B\}$
      - $\delta(q_0, B) = (q_d, B, S)$
      - $\delta(q_0, 0) = (q_1, 0, S)$
      - $\delta(q_0, 1) = (q_1, 1, S)$
      - $\delta(q_1, 0) = (q_1, 1, D)$
      - $\delta(q_1, 1) = (q_1, 0, D)$
      - $\delta(q_1, B) = (q_2, B, I)$
      - $\delta(q_2, 0) = (q_2, 0, I)$
      - $\delta(q_2, 1) = (q_2, 1, I)$
      - $\delta(q_2, B) = (q_d, 1, S)$
   iii. Lo que hace esta MT es:
      a. Si la cadena de entrada es λ, se detiene dejando λ.
      b. Si la cadena de entrada comienza con cero o uno, invierte todos los bits yendo hacia la derecha hasta encontrar el símbolo blanco.

    c. Al encontrar el símbolo blanco, vuelve al inicio de la cadena y agrega un uno al comienzo.

4. **Verificación de las condiciones de la reducción**:
    i. M_f es total?
        a. Sí, porque está definida para todos los elementos de Σ^*. ✅
    ii. M_f es computable?
        a. Sí, porque siempre se detiene, debido a que el alfabeto es finito, la cadena de entrada es finita, y no hay bucles infinitos en ninguna transición. ✅
    iii. Para cada palabra $w \in L_1$, $f(w) \in L_4$?
        a. Si la cadena de entrada es de la forma 0^n1, que pertenece a L_1, al aplicar $f(0^n1)$ se obtiene algo de la forma $1^{n+1}0$, que pertenece a L_4. ✅
            a. Por ejemplo, si $w = 001 \in L_1$, al aplicar $f(001)$ se obtiene $1110 \in L_4$.
    iv. Para cada palabra $z \notin L_1$, $f(z) \notin L_4$?
        a. Si la cadena de entrada es λ, que no pertenece a L_1, al aplicar $f(\lambda)$ se obtiene λ, que no pertenece a L_4. ✅
        b. Si la cadena de entrada no es de la forma 0^n1, que no pertenece a L_1, al aplicar f a esa cadena, se obtiene una cadena que no es válida para L_4, $f(z) \notin L_4$. ✅
            a. Por ejemplo, si $z = 101 \notin L_1$, al aplicar $f(101)$ se obtiene $1010 \notin L_4$.

Por lo tanto, $L_1 \ \alpha \ L_4$.

2. Sean L_1 y L_2, dos lenguajes tales que existe una reducción ($L_1 \ \alpha \ L_2$)

a. Qué se puede afirmar de L_1 si se sabe que $L_2 \in R$

1. Por enunciado: $L_1 \ \alpha \ L_2$
2. Por teorema se sabe que: $L_2 \in R \implies L_1 \in R$

Por lo tanto, se puede afirmar que $L_1 \in R$.

b. Qué se puede afirmar de L_1 si se sabe que $L_2 \in (\text{CO-RE} - RE)$

1. Por enunciado: $L_1 \ \alpha \ L_2$
2. Por teorema se sabe que: $L_2 \in RE \implies L_1 \in RE$
3. $L_2 \in (\text{CO-RE} - RE)$
4. $\implies (L_2 \in \text{CO-RE}) \land (L_2 \notin RE)$ (Por def. de diferencia de conjuntos)
5. $\implies (\overline{L_2} \in RE) \land (L_2 \notin RE)$ (Por def. de CO-RE)

6. $\implies \overline{L_2} \in RE \rightarrow \overline{L_1} \in RE$ (Por teorema)
7. $\implies \overline{L_1} \in RE \implies L_1 \in \text{CO-RE}$ (Por def. de CO-RE)
8. $\implies L_2 \in (\text{CO-RE} - RE) \implies L_1 \in \text{CO-RE}$

Por lo tanto, se puede afirmar que si se sabe que $L_2 \in (\text{CO-RE} - RE)$, entonces esto implica que $L_1 \in \text{CO-RE}$.

c. Qué se puede afirmar de L_2 si se sabe que $L_1 \in R$

1. Por enunciado: $L_1 \ \alpha \ L_2$
2. Por teorema se sabe que: $L_2 \in R \implies L_1 \in R$
3. La contrarrecíproca del teorema es: $L_1 \notin R \implies L_2 \notin R$
4. La reducción establece una relación de grado de dificultad entre los lenguajes, por lo cual si L_1 es decidible, L_2 podría serlo también, o ser solo recursivamente enumerable, o incluso no ser ni siquiera recursivamente enumerable.

Por lo tanto, no se puede afirmar nada sobre L_2 si se sabe que $L_1 \in R$.

d. Qué se puede afirmar de L_2 si se sabe que $L_1 \in (\text{CO-RE} - RE)$

1. Por enunciado: $L_1 \ \alpha \ L_2$
2. Por teorema se sabe que: $L_2 \in RE \implies L_1 \in RE$
3. La contrarrecíproca del teorema es: $L_1 \notin RE \implies L_2 \notin RE$
4. $L_1 \in (\text{CO-RE} - RE) \implies L_1 \notin RE$
5. $L_1 \notin RE \implies L_2 \notin RE$ (Por contrarrecíproca del teorema)

Por lo tanto, se puede afirmar que si se sabe que $L_1 \in (\text{CO-RE} - RE)$, entonces esto implica que $L_2 \notin RE$.

3. Sean HP y L_u los lenguajes Halting Problem y Lenguaje Universal respectivamente.

$$\mathrm{HP} = \{(\langle M \rangle, w) \mid \mathrm{M\ se\ detiene\ con\ input\ w}\}$$

$$L_u = \{(\langle M \rangle, w) \mid \mathrm{M\ acepta\ w}\}$$

Demuestre que existe una reducción $\mathrm{HP}\ \alpha\ L_u$

1. **Análisis de los lenguajes**:
   i. HP contiene todos los pares (código binario de MT, cadena de entrada) tales que la MT se detiene, ya sea en q_A o en q_R, cuando recibe esa cadena de entrada.
   ii. L_u contiene todos los pares (código binario de MT, cadena de entrada) tales que la MT se detiene en q_A cuando recibe esa cadena de entrada.
2. **Definición de la reducción**:
   i. $\mathrm{HP}\ \alpha\ L_u$ si existe una función total computable $f : \Sigma^* \to \Sigma^*$ tal que a cada palabra $w \in \mathrm{HP}$ le corresponde una palabra $f(w) \in L_u$, y a cada palabra $z \notin \mathrm{HP}$ le corresponde una palabra $f(z) \notin L_u$.
3. **Construcción de la M_f**:
   i. Intuitivamente, lo que se debe hacer es:
      a. Para cada par válido perteneciente a HP, se debe construir un par válido perteneciente a L_u.
      b. Para cada par inválido NO perteneciente a HP, se debe construir un par inválido NO perteneciente a L_u.
   ii. Podemos definir una MT M_f de cómputo que compute la función f y que siempre se detenga:
   iii. La máquina M_f recibe como entrada un par $(\langle M \rangle, w)$ y hace lo siguiente:
      a. Si $(\langle M \rangle, w)$ es un **par inválido**, es decir que no hay un código de MT + una cadena de entrada, o que solo hay uno de los dos, o que la coma separadora no está, etc, entonces $(\langle M \rangle, w) \notin HP$, y por lo tanto se debe dejar algo en la cinta $\notin L_u$. Para esto simplemente la M_f no hace absolutamente nada, ya que como el par es inválido, automáticamente no forma parte de L_u.
      b. Si $(\langle M \rangle, w)$ es un **par válido pero $\langle M \rangle$ no es una codificación válida de MT**, se asume que M es una MT que se detiene sin hacer nada, es decir, $L(M) = \emptyset$. Claramente esta MT se detiene con input w, porque se sabe que $\emptyset \in R$. Entonces, $(\langle M \rangle, w) \in HP$, y por lo tanto se debe dejar algo en la cinta $\in L_u$. Para esto, M_f deja en la cinta un nuevo par $(\langle M' \rangle, w)$, donde M' es una MT que acepta toda cadena de entrada, es decir, $L(M') = \Sigma^*$. La cadena de entrada w es la misma que la del par original. Así, como M' acepta toda cadena de entrada, en particular acepta w, por lo que $(\langle M' \rangle, w) \in L_u$.

c. Si $(\langle M \rangle, w)$ es un **par válido en todo sentido**, es decir que $\langle M \rangle$ es una codificación válida de MT y w es una cadena de entrada, y está la coma separadora, entonces M_f intercambia todas las transiciones hacia el estado q_R por el estado q_A, y deja en la cinta el nuevo par $(\langle M'' \rangle, w)$, donde M'' es la MT resultante de hacer ese cambio en M. La cadena de entrada w es la misma que la del par original.

4. **Verificación de las condiciones de la reducción**:
   i. M_f es total?
      a. Sí, porque está definida para todos los elementos de Σ^*. ✅
   ii. M_f es computable?
      a. Sí, porque siempre se detiene, debido a que el alfabeto es finito, la cadena de entrada es finita, y no hay bucles infinitos en ninguna transición. ✅
   iii. Para cada palabra $w \in \text{HP}$, $f(w) \in L_u$?
      a. Si el par de entrada es válido pero $\langle M \rangle$ no es una codificación válida de MT, entonces $(\langle M \rangle, w) \in HP$, y al aplicar f se obtiene $(\langle M' \rangle, w) \in L_u$ porque M' acepta toda cadena de entrada. ✅
      b. Si el par de entrada es válido en todo sentido, y M se detiene en q_R con input w, entonces $(\langle M \rangle, w) \in HP$, y al aplicar f se obtiene $(\langle M'' \rangle, w)$, donde M'' se detiene en q_A con input w, por lo que $(\langle M'' \rangle, w) \in L_u$. ✅
      c. Si el par de entrada es válido en todo sentido, y M se detiene en q_A con input w, entonces $(\langle M \rangle, w) \in HP$, y al aplicar f se obtiene exactamente el mismo par, por lo que $(\langle M'' \rangle, w) \in L_u$. ✅
   iv. Para cada palabra $z \notin \text{HP}$, $f(z) \notin L_u$?
      a. Si el par de entrada es inválido, entonces $(\langle M \rangle, w) \notin HP$, y al aplicar f se obtiene ese mismo par inválido que no pertenece a L_u. ✅
      b. Si el par de entrada es válido en todo sentido, y M loopea con input w, entonces $(\langle M \rangle, w) \notin HP$, y al aplicar f se obtiene $(\langle M'' \rangle, w)$, donde M'' loopea con input w, por lo que $(\langle M'' \rangle, w) \notin L_u$. ✅

Por lo tanto, $\text{HP} \ \alpha \ L_u$.

4. Sea HP_λ el Halting Problem a partir de la cinta en blanco:

$$\text{HP}_\lambda = \{\langle M \rangle \mid \text{M se detiene con input } \lambda\}$$

Demuestre que existe una reducción $\text{HP} \ \alpha \ \text{HP}_\lambda$

1. **Análisis de los lenguajes**:

i. HP contiene todos los pares (código binario de MT, cadena de entrada) tales que la MT se detiene, ya sea en q_A o en q_R, cuando recibe esa cadena de entrada.

ii. HP_λ contiene todos los códigos binarios de MT tales que la MT se detiene, ya sea en q_A o en q_R, cuando recibe λ como cadena de entrada.

2. **Definición de la reducción**:

i. $\mathrm{HP}\ \alpha\ \mathrm{HP}_\lambda$ si existe una función total computable $f : \Sigma^* \to \Sigma^*$ tal que a cada palabra $w \in \mathrm{HP}$ le corresponde una palabra $f(w) \in \mathrm{HP}_\lambda$, y a cada palabra $z \notin \mathrm{HP}$ le corresponde una palabra $f(z) \notin \mathrm{HP}_\lambda$.

3. **Construcción de la M_f**:

i. Intuitivamente, lo que se debe hacer es:

a. Para cada par válido perteneciente a HP, se debe construir un código de MT válido perteneciente a HP_λ.

b. Para cada par inválido NO perteneciente a HP, se debe construir un código de MT inválido NO perteneciente a HP_λ.

ii. Podemos definir una MT M_f de cómputo que compute la función f y que siempre se detenga:

iii. La máquina M_f recibe como entrada un par $(\langle M \rangle, w)$ y hace lo siguiente:

a. Si $(\langle M \rangle, w)$ es un **par inválido**, es decir que no hay un código de MT + una cadena de entrada, o que solo hay uno de los dos, o que la coma separadora no está, etc, entonces $(\langle M \rangle, w) \notin HP$, y por lo tanto se debe dejar en la cinta algo $\notin HP_\lambda$. Para esto simplemente la M_f convierte toda la cinta a blancos, dejando así λ en la cinta tal que $\lambda \notin HP_\lambda$.

b. Si $(\langle M \rangle, w)$ es un **par válido pero $\langle M \rangle$ no es una codificación válida de MT**, se asume que M es una MT que se detiene sin hacer nada, es decir, $L(M) = \emptyset$. Claramente esta MT se detiene con input w, porque se sabe que $\emptyset \in R$. Claramente esta MT se detiene con input λ, porque lo único que hace es leer el símbolo blanco y detenerse. Entonces, $(\langle M \rangle, w) \in HP$, y por lo tanto se debe dejar algo en la cinta $\in HP_\lambda$. Para esto, M_f deja en la cinta un nuevo código $\langle M' \rangle$, donde M' es una MT que acepta toda cadena de entrada, es decir, $L(M') = \Sigma^*$. Así, como M' acepta toda cadena de entrada, obviamente acepta λ también, por lo que $\langle M' \rangle \in HP_\lambda$.

c. Si $(\langle M \rangle, w)$ es un **par válido en todo sentido**, es decir que $\langle M \rangle$ es una codificación válida de MT y w es una cadena de entrada, y está la coma separadora, entonces M_f

...

4. **Verificación de las condiciones de la reducción**:

i. M_f es total?

a. Sí, porque está definida para todos los elementos de Σ^*. ✅

ii. M_f es computable?

a. Sí, porque siempre se detiene, debido a que el alfabeto es finito, la cadena de entrada es finita, y no hay bucles infinitos en ninguna transición. ✅

iii. Para cada palabra $w \in \mathrm{HP}$, $f(w) \in \mathrm{HP}_\lambda$?

...

iv. Para cada palabra $z \notin \mathrm{HP}$, $f(z) \notin \mathrm{HP}_\lambda$?

...

Por lo tanto, $\mathrm{HP}\ \alpha\ \mathrm{HP}_\lambda$.

5. Demuestre que $L_V \notin RE$:

$$L_V = \{\langle M \rangle \mid L(M) = \emptyset\}.$$

Considere que si $\langle M \rangle$ es un código inválido de máquina de Turing también pertence a L_V (ya que no reconoce ningún string). Así L_V es el complemento del lenguaje $L_{NV} = \{\langle M \rangle \mid L(M) \neq \emptyset\}$

(Ayuda: puede utilizar el complemento de L_u para encontrar una reducción)

...