Explicación práctica

Una reducción es una función, es total computable y tiene una entrada y salida. Todo $w \in L_1$ la función de reducción hace que este $w \in L_2$.

a una región
Las reducciones se usan para probar la pertenencia? Es una función entre lenguajes.

Los pasos son:

1. Cómo se comporta la máquina ante una entrada
2. Probar que es total computable
3. Probar que la reducción es correcta.

Para probar la pertenencia usamos el teorema:

$Si\ L_1 \leq L_2,$ entonces $L_2 \in R \to L_1 \in R$

$\cdot Si\ L_1 \leq L_2,$ entonces $L_2 \in RE \to L_1 \in RE$

El corolario, para mostrar que no pertenece:

$Si\ L_1 \leq L_2$ entonces $L_1 \notin R \to L_2 \notin R$

$Si\ L_1 \leq L_2$ entonces $L_1 \notin RE \to L_2 \notin RE$

Propiedades:

- Reflexibilidad: se puede hacer una reducción a sí mismo.
- Transitividad: si $L_1 \leq L_2$ y $L_2 \leq L_3$ entonces $L_1 \leq L_3$.
- Otra: $L_1 \leq L_2 \leftrightarrow L_1^C \leq L_2^C$.

La propiedad de simetría no la tiene. Puede ser que $L_1 \leq L_2$ pero no al revés.

Todos los lenguajes de RE se reducen a L_u y L_{HP}.

- **Autómata restringido o finito**: una MT con restricciones.
- **Autómata con pila**: como la restringida pero con pila.

Todos los L ER reducen entre sí.

Práctica 4

①

A. se siguen los pasos que se explican.

1. Para cumplir que $f(x) = x$, fuera una función de reducción de HP a L_u válida, se debe cumplir que:
   - Dado un $w \in HP$, se da que $f(w) \in L_u$
   - Dado un $w \notin HP$, se da que $f(w) \notin L_u$
   - f es computable

f sí es computable porque dada una entrada w se copia en una cantidad finita de pasos.

Como $HP \in L_u$, existe un caso donde $w \in (L_u - HP)$ eso quiere decir que $w \notin HP$ pero $f(w) \in L_u$, por lo tanto f no es válida de HP a L_u.

B. Las MT M' tienen invertidas las q_R por q_A, por lo tanto dado un w nunca pasará en q_R. Termina o en q_A o loopeando.

C. La función para reducir HP a L_u también sirve para demostrar la reducción de HP^c a L_u^c debido a una propiedad de la reducción donde

$$L_1 \alpha L_2 \Leftrightarrow L_1^c \alpha L_2^c$$

D. Por la propiedad de la simetría, la cual indica que la reducción no cumple con la simetría.

e. No se puede garantizar que M_F se detenga, por lo tanto f no es totalmente computable, entonces no es una función de reducción válida.

2 Para que $f(x) = \langle M_{\Sigma^*} \rangle$ fuera una función de reducción de L_U a L_{Σ^*} válida se debe cumplir:

1. Dado un $w \in L_U$, se da que $f(w) \in L_{\Sigma^*}$
2. Dado un $w \notin L_U$, se da que $f(w) \notin L_{\Sigma^*}$
3. f es totalmente computable

f siempre es computable porque siempre copia $f(M_{\Sigma^*})$ en una cantidad finita de pasos.

Dado un $w \notin L_U$ pero $f(w) \in L_{\Sigma^*}$ entonces no es una función de reducción válida.

Se quiere demostrar que $L_{\Sigma^*} \notin RE$, mediante un teorema donde $L_1 \propto L_2$ entonces $L_2 \notin RE \rightarrow L_1 \notin RE$ pero como no se puede obtener una función de reducción válida no se puede ejecutar el teorema y ∴ demostrar que $L_{\Sigma^*} \notin RE$.

3 Sabiendo que existe una reducción de L_U^C a $L_\emptyset$ y que $L_U^C \notin$ CO-RE, se puede usar el teorema que indica que:

$L_1 \propto L_2$, entonces $L_1 \notin RE \rightarrow L_2 \notin RE$

Entonces, $L_\emptyset$ no pertenece a RE y pertenece a CO-RE.

(4) La función que copia $(\langle M_i \rangle, w_i)$ a la salida:
1) Dado un $(\langle M_i \rangle, w_i) \in D_{HP}$, M_i para sobre w_i por lo que $f((\langle M_i \rangle, w_i)) = (\langle M \rangle, w)$, donde M para sobre w, por lo que $(\langle M \rangle, w) \in HP$.
2) Dado un $(\langle M \rangle, w) \notin D_{HP}$, M_i no para sobre w_i por lo que $f((\langle M_i \rangle, w_i)) = (\langle M \rangle, w)$ donde M no para sobre w, por lo que $(\langle M \rangle, w) \notin HP$.
3) Dado un $x \notin D_{HP}$, no es una string válida, por lo que $f(x) = x$, y $x \notin HP$.

No sé si es computable.

(5) $M = \{Q, \Sigma, \delta, q_A, q_R, q_0\}$

$Q = \{q_0, q_A, q_R, q_1\}$

q_0 = estado inicial

q_1 = busca q_1

$\delta(q_0, 0) = q_1$

$\delta(q_0, B) = q_A$

$\delta(q_1, 1) = q_0$

$\delta(q_1, B) = q_A$

$\delta(q_1, 0) = q_R$

¿y $\delta(q_0, 1)$?

(6) La cadena de entrada cuenta con $|w|$ caracteres, si MT tiene $|Q|$ estados y su lenguaje $|\Sigma|$ símbolos entonces puede haber $|Q| \cdot |\Sigma|^{|w|}$ pasos antes de entrar en bucle. ¿Por qué es recursivo?