Demostrando cotas inferiores: reducciones

Hemos visto dos técnicas para obtener cotas inferiores para el número de operaciones realizadas por una clase de algoritmos: mejor estrategia del adversario y árboles de decisión

Demostrando cotas inferiores: reducciones

Hemos visto dos técnicas para obtener cotas inferiores para el número de operaciones realizadas por una clase de algoritmos: mejor estrategia del adversario y árboles de decisión

Vamos a introducir una tercera técnica para obtener cotas inferiores que está basada en la idea de reducir un problema F a otro problema G

- De esta forma podemos usar una cota inferior para F para deducir una cota inferior para G

Una noción de reducción

Sean $F : \Sigma^* \to \Sigma^*$ y $G : \Sigma^* \to \Sigma^*$ dos funciones.

Una noción de reducción

Sean $F : \Sigma^* \to \Sigma^*$ y $G : \Sigma^* \to \Sigma^*$ dos funciones.

Una función $H : \Sigma^* \to \Sigma^*$ es una reducción de F a G si para cada $w \in \Sigma^*$ tenemos que $F(w) = G(H(w))$

- H transforma una entrada w de F en una entrada $H(w)$ de G con la cual se tiene la misma salida

Una cota inferior a partir de la reducción

Suponga lo siguiente:

- Todo algoritmo que calcula F debe realizar al menos $f(n)$ operaciones para las entradas de largo n
- Existe un algoritmo que calcula H y que realiza $h(n)$ operaciones para las entradas de largo n
- Para cada $w \in \Sigma^*$ se tiene que $|H(w)| = \ell(|w|)$

Una cota inferior a partir de la reducción

Sea $\mathcal{A}$ un algoritmo que calcula G

- Recuerde que la función $t_{\mathcal{A}} : \mathbb{N} \to \mathbb{N}$ nos da el número de operaciones realizadas por $\mathcal{A}$ en el peor caso

Una cota inferior a partir de la reducción

Sea $\mathcal{A}$ un algoritmo que calcula G

- Recuerde que la función $t_{\mathcal{A}} : \mathbb{N} \to \mathbb{N}$ nos da el número de operaciones realizadas por $\mathcal{A}$ en el peor caso

Dado que H es una reducción de F en G, concluimos por los supuestos anteriores que $f(n) \leq h(n) + t_{\mathcal{A}}(\ell(n))$

Una cota inferior a partir de la reducción

Sea $\mathcal{A}$ un algoritmo que calcula G

- Recuerde que la función $t_{\mathcal{A}} : \mathbb{N} \to \mathbb{N}$ nos da el número de operaciones realizadas por $\mathcal{A}$ en el peor caso

Dado que H es una reducción de F en G, concluimos por los supuestos anteriores que $f(n) \leq h(n) + t_{\mathcal{A}}(\ell(n))$

La ecuación $f(n) \leq h(n) + t_{\mathcal{A}}(\ell(n))$ es usada para obtener una cota inferior para $t_{\mathcal{A}}(n)$

- Si $\mathcal{A}$ es un algoritmo arbitrario en una clase $\mathcal{C}$ de algoritmos, entonces obtenemos una cota inferior para el número de operaciones que debe realizar cualquier algoritmo en $\mathcal{C}$ en el peor caso

Algunas consideraciones importantes

- Los supuestos iniciales pueden ser relajados
  - Por ejemplo, podemos consider una cota inferior para F en el peor caso o podemos dejar de exigir que $|H(w)| = \ell(|w|)$
- Otras nociones de reducción pueden ser utilizadas
  - Por ejemplo, podemos pedir que existan funciones H_1 y H_2 tales que $F(w) = H_2(G(H_1(w)))$
- Otros supuestos pueden ser considerados
  - Por ejemplo, la función f es no decreciente o estrictamente creciente

Algunas consideraciones importantes

Bajo estas nuevas condiciones se deduce otra ecuación que nos permite dar una cota inferior para el número de operaciones realizadas por un algoritmo que calcula G

Vamos a ver un ejemplo de esta técnica en el próximo capítulo