.UBA fiuba
FACULTAD DE INGENIERÍA

Reducciones polinomiales

Teoría de Algoritmos I (75.29 / 95.06)
Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Introducción

Buscamos clasificar problemas computacionales

De acuerdo a la complejidad que requiere su resolución

Definimos 2 grandes clasificaciones

Clase P

Clase NP

(existen muchas mas dentro del “ecosistema”…)

Necesitamos una herramienta

Que nos permita comparar 2 problemas

Asignar un problema a una determinada clase

Reducciones

Reducir un problema

a otro conocido para resolverlo

Es un procedimiento

Que utilizamos profusamente en la resolución de problemas computacionales

Se lo puede pensar como

una caja negra

Reducciones – Caja negra

Dado una instancia "y" de un problema Y

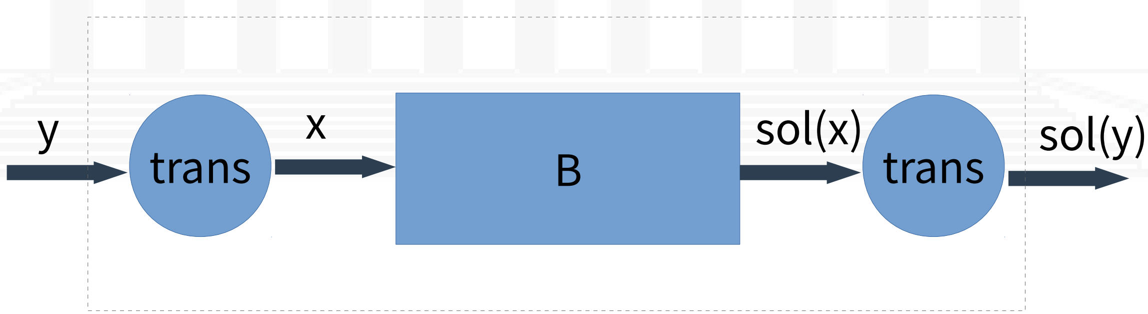
- Realizamos una transformación a una instancia "x" del problema X

Resolvemos "x" mediante un algoritmo "B"

- Obteniendo el resultado sol(x)

Transformamos el resultado sol(x)

- En el resultado sol(y) del problema "y"



Reducción polinomial

Corresponde a una reducción

En la que ambas transformaciones se realizan en tiempo polinomial

Sean

X, Y problemas

Diremos

$Y \leq_p X$ (se lee: “Y” es polinomialmente reducible (en tiempo) a “X”)

Si

podemos transformar cualquier instancia de y en una instancia de x en tiempo polinómico

Usos de las reducciones polinómicas

Como “caja negra”

Para resolver problemas (de forma tractable)

Como “medida” de complejidad

Para comparar y clasificar problemas

Origen

Propuesto en:

"Reducibility among combinatorial problems", Richard M. Karp (1972)

https://people.eecs.berkeley.edu/~luca/cs172/karp.pdf

Comparar problemas con reducciones

Sean

X, Y problemas

Si

$Y \leq_p X$

Diremos

Que el problema X es al menos tan difícil que el problema Y

Ejemplo

Sea

El problema “hallar el matching mas grande en un grafo bipartito” (MAX-MATCHING)

Se puede

Reducir polinomialmente al problema de “flujo máximo en una red de flujo” (MAX-FLOW).

Por lo tanto

MAX-MATCHING $\leq_p$ MAX-FLOW

(MAX-FLOW es al menos tan difícil de resolver que MAX-MATCHING)

Acotar un problema a la clase “P”

Sean

X,Y Problemas

Si

X $\in$“P”

Y $\leq_p$ X

Entonces

Y $\in$“P” (por que X es igual o mas complicado que Y)

En el ejemplo anterior

MAX-MATCHING $\leq_p$ MAX-FLOW

MAX-FLOW $\in$“P” $\Rightarrow$ MAX-MATCHING $\in$ “P”

Acotar un problema a la clase "P" (cont.)

Sean

X,Y Problemas

Si

$Y \notin$ "P"

$Y \leq_p X$

Entonces

$X \notin$ "P" (por que X es igual o más complicado que Y)

Propiedad: Equivalencia

Sean

X,Y Problemas

Si

$Y \leq_p X$

$X \leq_p Y$

Entonces

X e Y tienen la misma complejidad

Propiedad: Transitividad

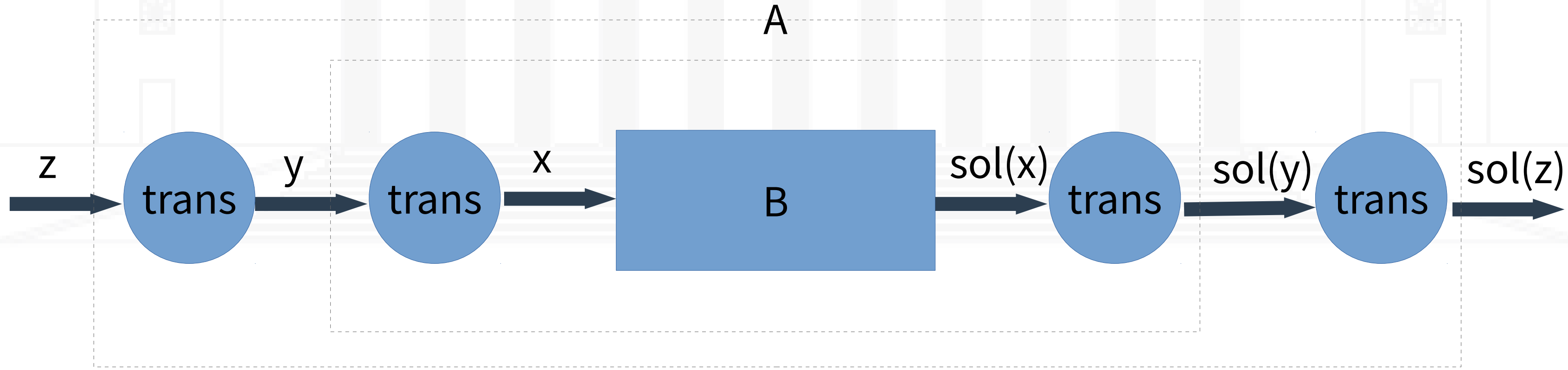
Si

$Z \leq_p Y$

$Y \leq_p X$

Entonces

$Z \leq_p X$



Presentación realizada en Junio de 2020