.mdf

universitätbonn

Big Data Kernelization

Matthias Mnich

max planck institut informatik

June 11, 2014

Efficient Compression of Large Instances

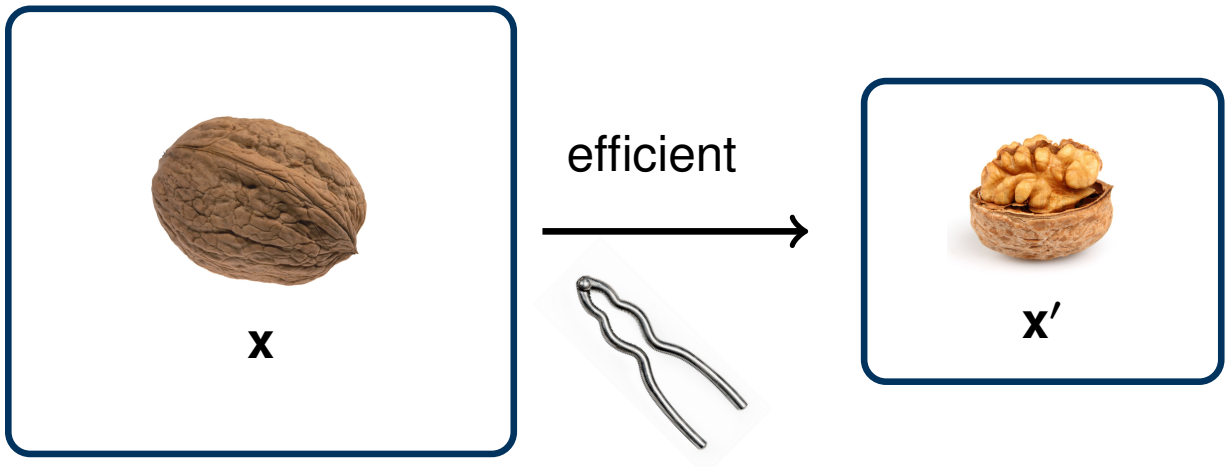
Let x be an instance of a(n NP-)hard optimization problem.

> Can we simplify x to its “hard kernel”?

Efficient Compression of Large Instances

Let x be an instance of a(n NP-)hard optimization problem.

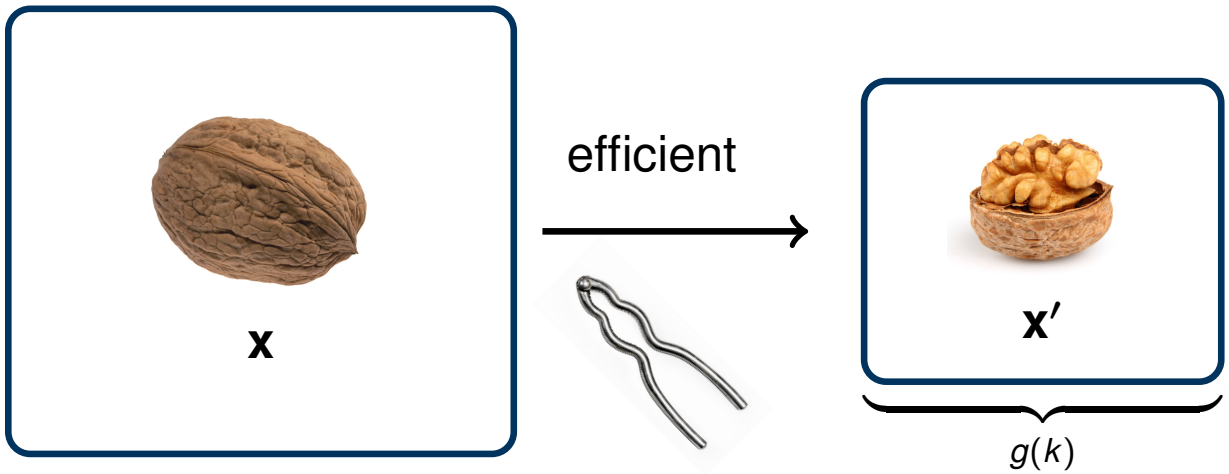
Can we simplify x to its "hard kernel"?



Efficient Compression of Large Instances

Let x be an instance of a(n NP-)hard optimization problem.

Can we simplify x to its "hard kernel"?

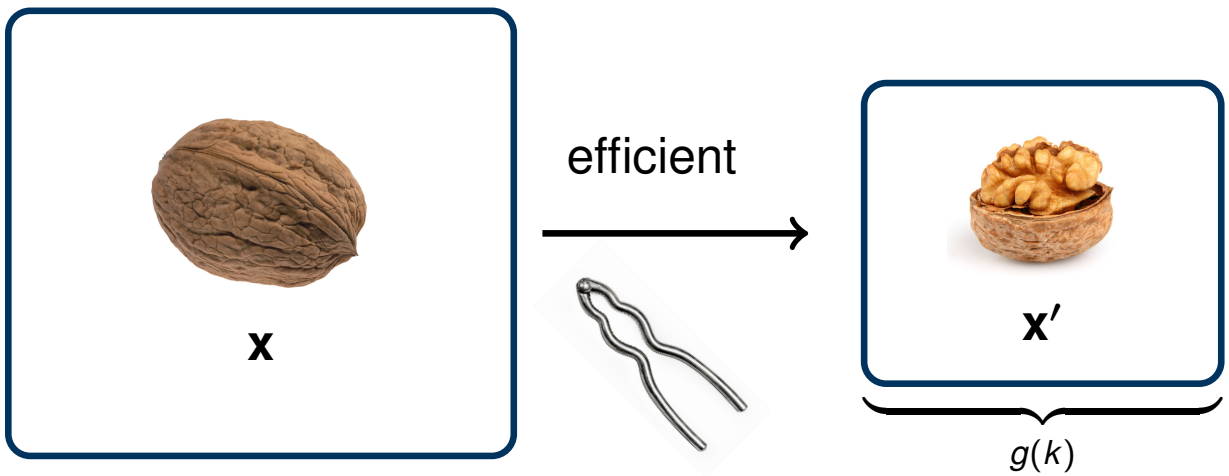


Kernel

Efficient Compression of Large Instances

Let x be an instance of a(n NP-)hard optimization problem.

Can we simplify x to its "hard kernel"?

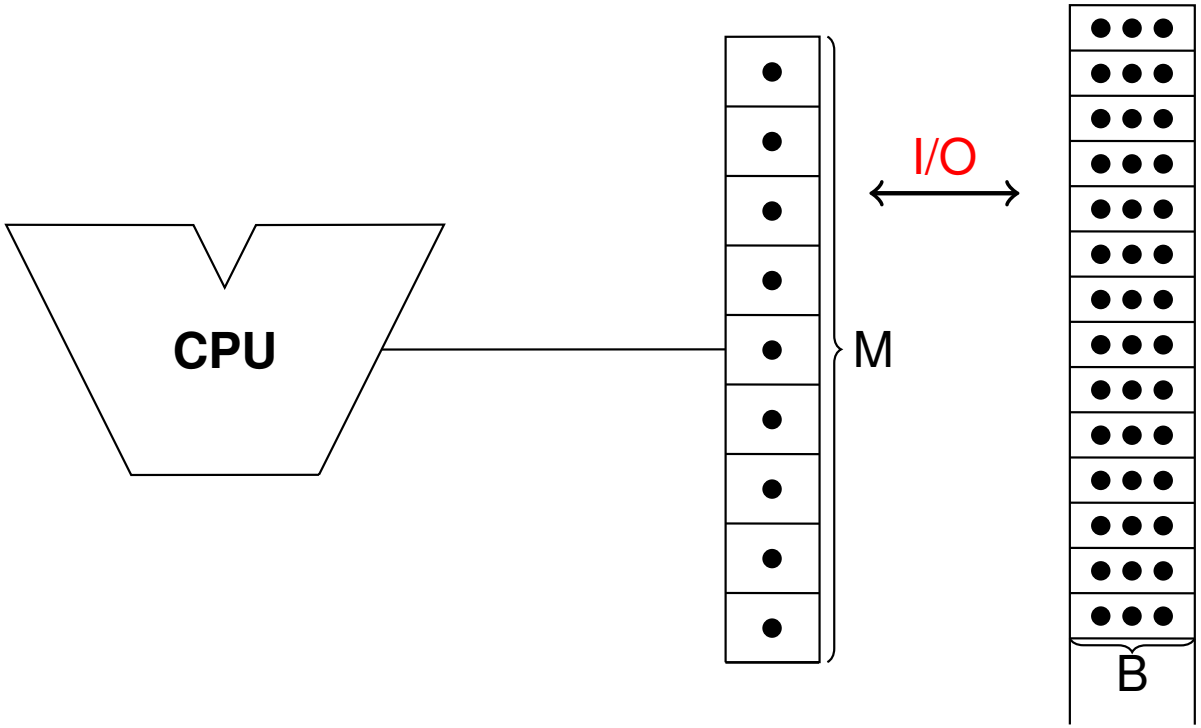


Kernel

Goal: Kernel as small as possible, ideally $\mathcal{O}(k^c)$ or $\mathcal{O}(k)$

- VERTEX COVER: Kernel with $2k$ vertices
- k-PATH: Kernel with $\mathcal{O}(1.66^k)$ vertices

Input-Output Efficient Algorithms



I/O = read/write B data items from RAM to disk
running "time" of an algorithm: number of I/Os

Basic Operations

- scanning — $\text{scan}(N) = \Theta(N/B)$
- permuting — $\text{perm}(N) = \Theta(\min\{N, \text{sort}(N)\})$
- sorting — $\text{sort}(N) = \Theta((N/B)\log_{M/B}(N/B))$

$\text{scan}(N) < \text{sort}(N) \ll N$

Input-Output Efficient Kernelization