Lossy Kernelization

Parameterized Complexity Summer School

ALGO 2017

Vienna

M. S. Ramanujan

TU Wien

LOSSY Kernelization

APPROXIMATION ALGORITHMS

KERNELIZATION

LOSSY Kernelization

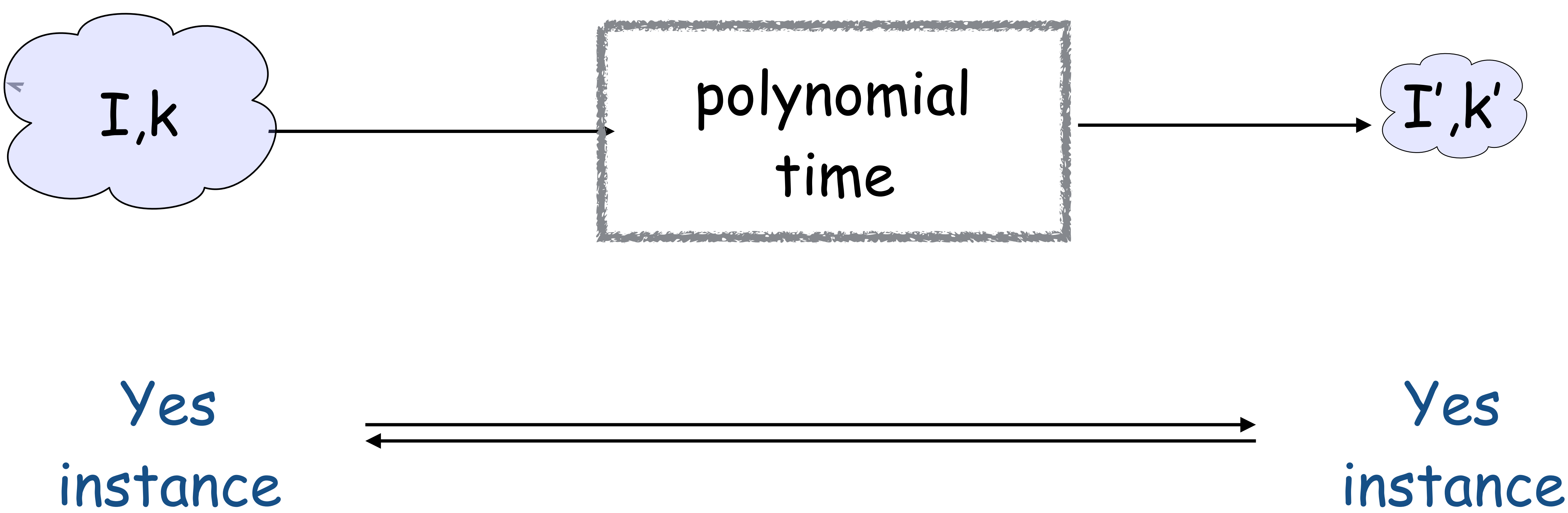
APPROXIMATION ALGORITHMS

KERNELIZATION

LOSSY

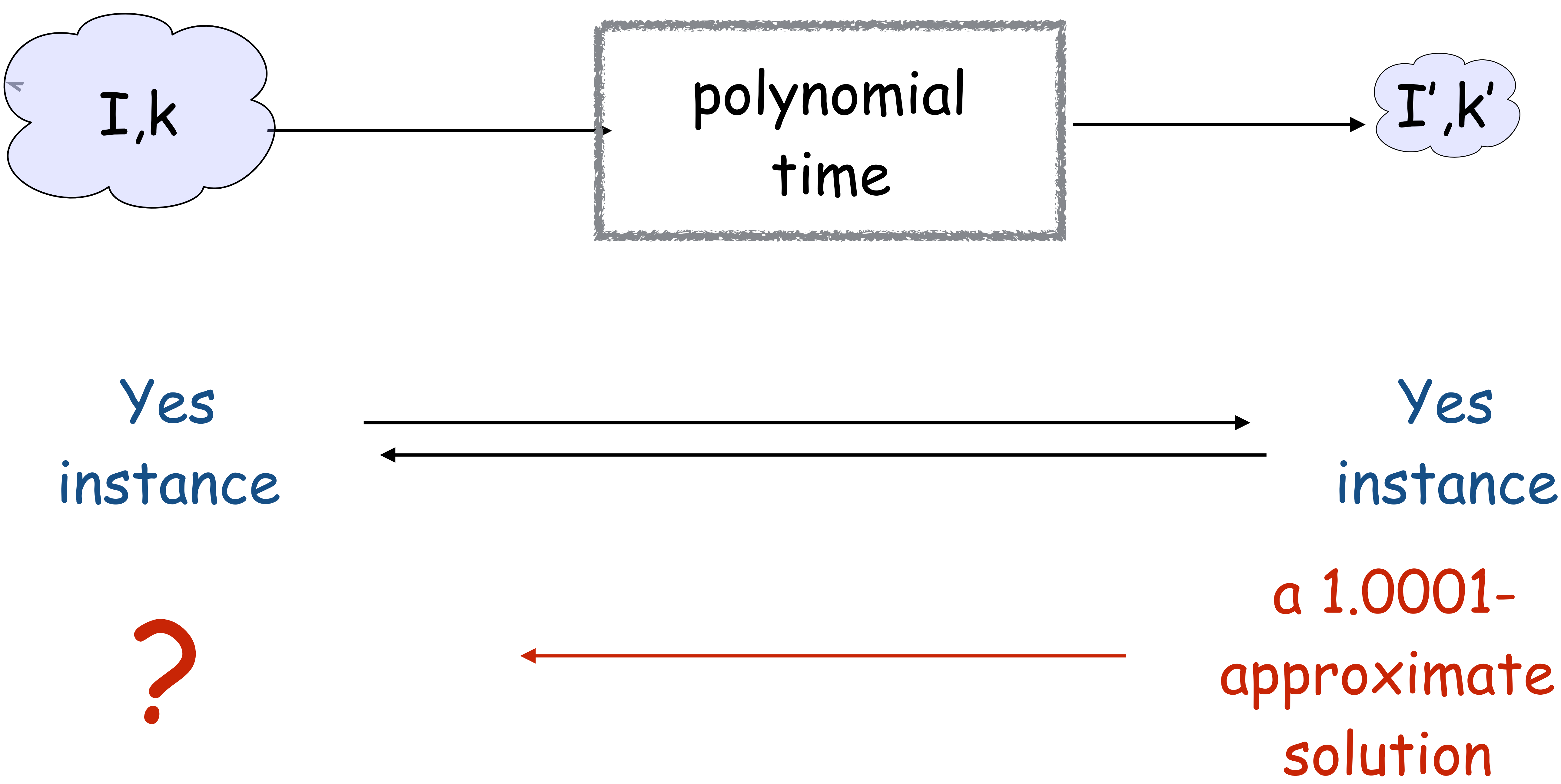
KERNELS

Kernelization

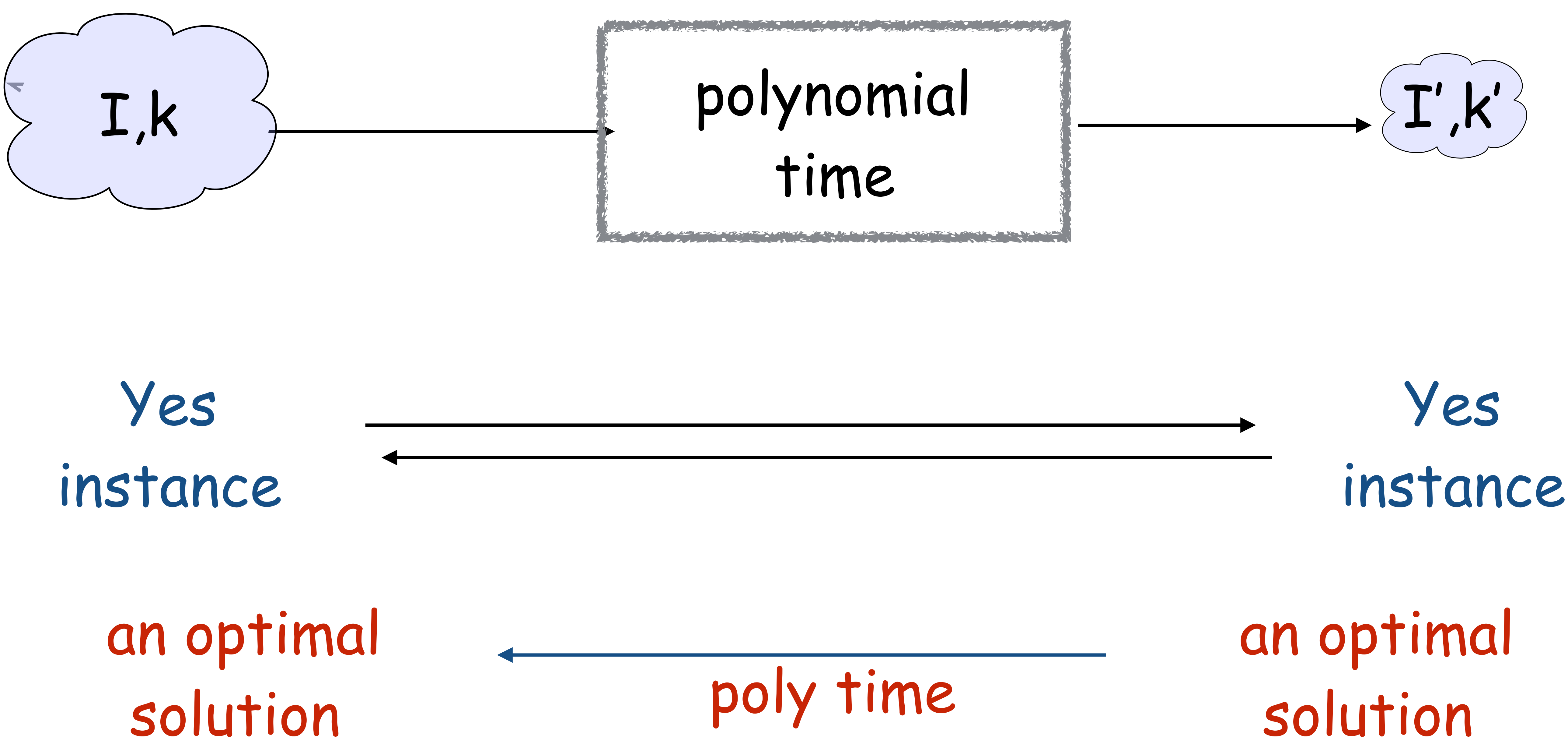


Polynomial Kernel if |I'|+k' < poly(k)

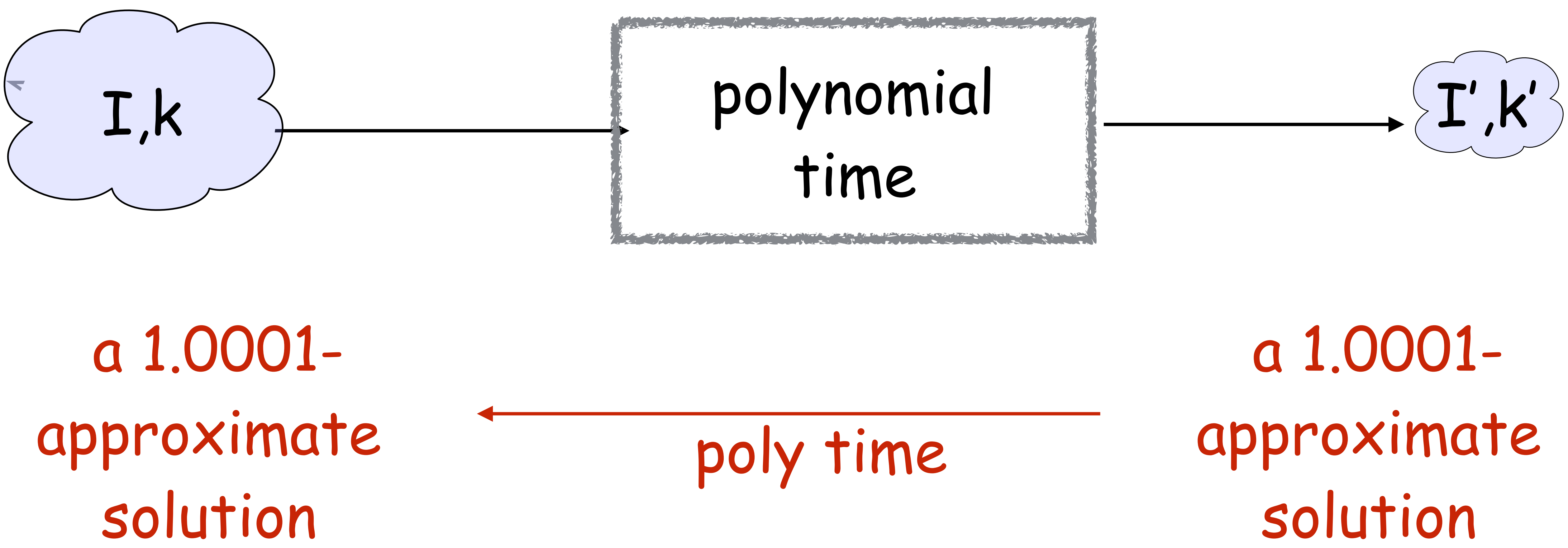
Kernelization



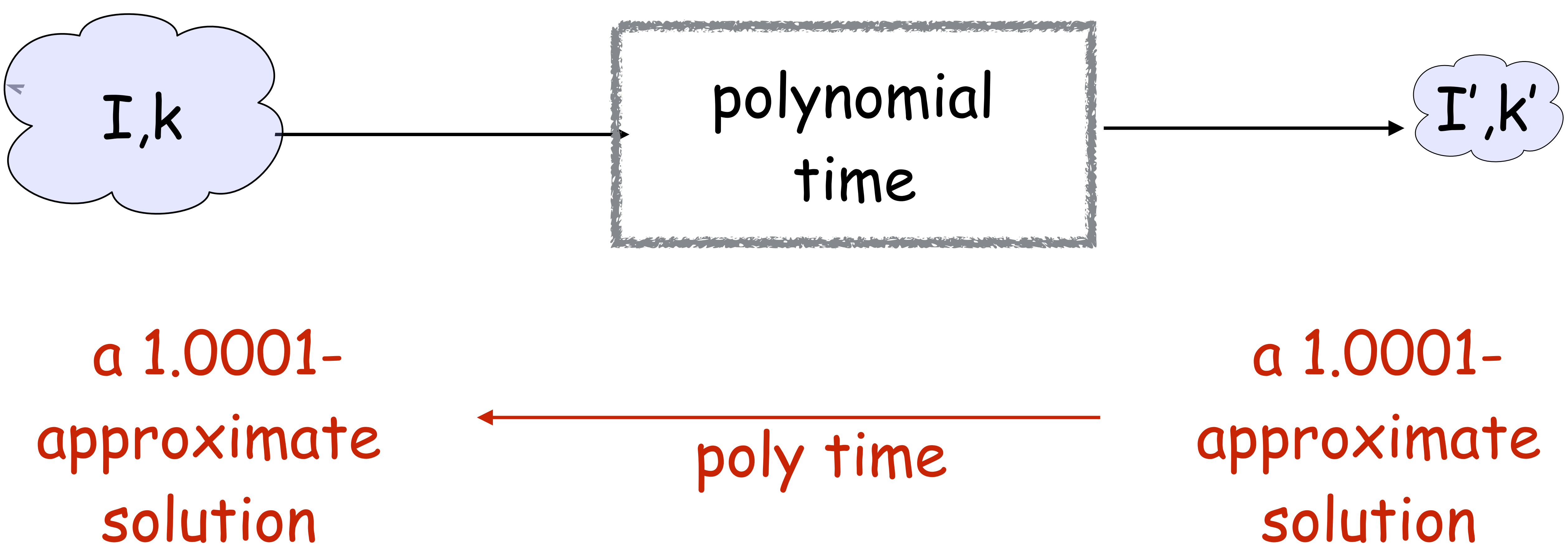
Kernelization



Kernelization



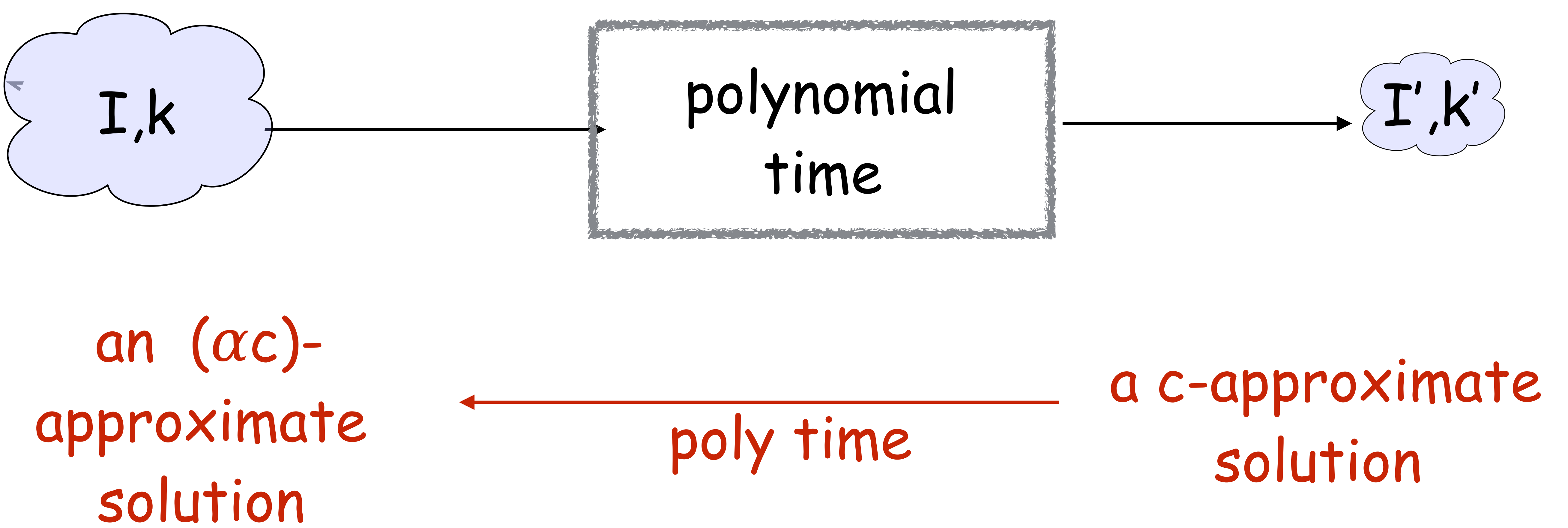
approximate kernelization



But if we are allowing a loss while solving I′, makes sense to allow a slight loss while lifting a solution back to I

α-approximate Kernelization

[Lokshtanov, Panolan, R., Saurabh, 17]



Polynomial α-approximate Kernel if |I'|+k' < poly(k)

Previously

A problem is FPT
if and only if
it has a kernel.

A problem has an FPT time α-approximation
if and only if
it has a α-approximate kernel

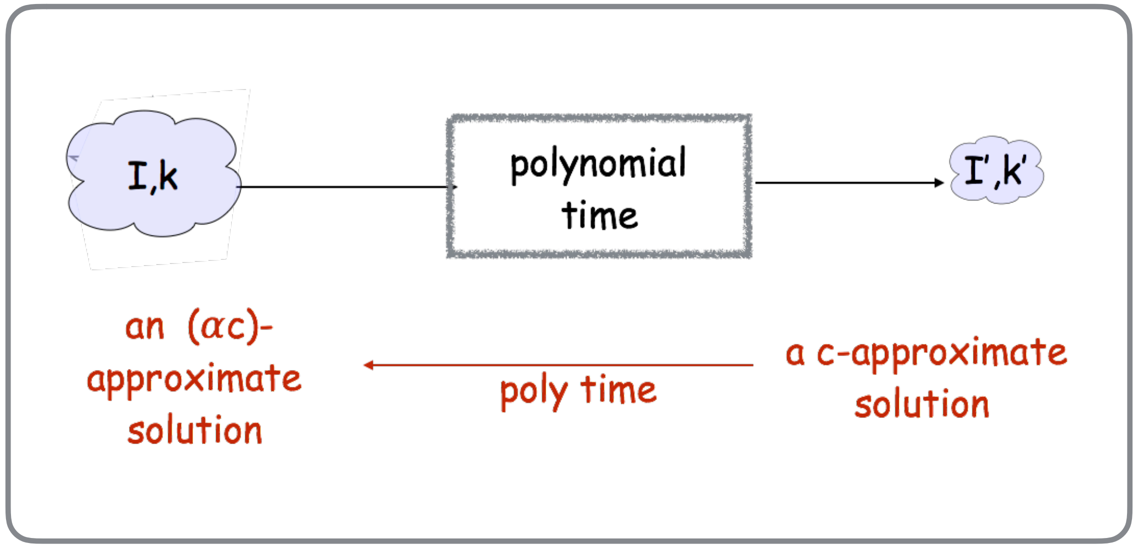
Now

Previously

A problem is in P
if and only if
it has a constant size kernel.

A problem has a poly time α-approximation
if and only if
it has a constant size α-approximate kernel

Now



want to define this object

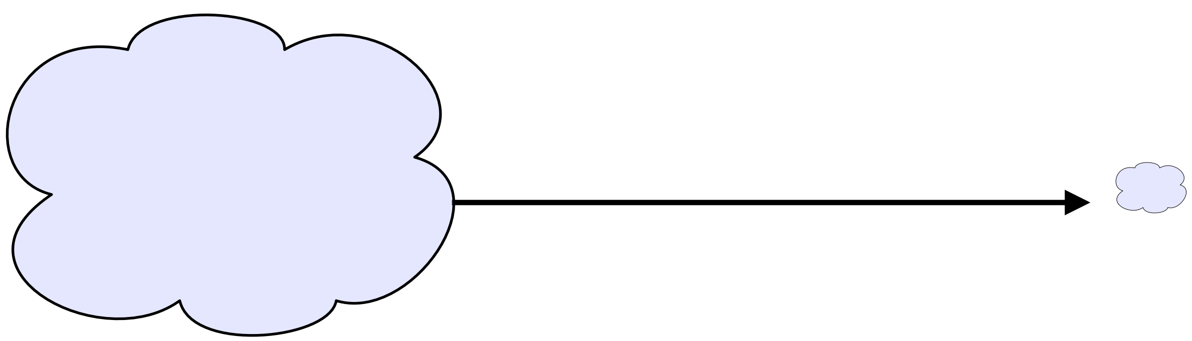
A problem has a poly time α-approximation
if and only if
it has a constant size α-approximate kernel

we are given a poly time α-approx algo

(I,k)

(I′,k′)

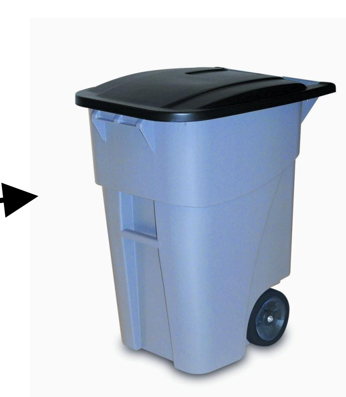
an arbitrary instance of constant size

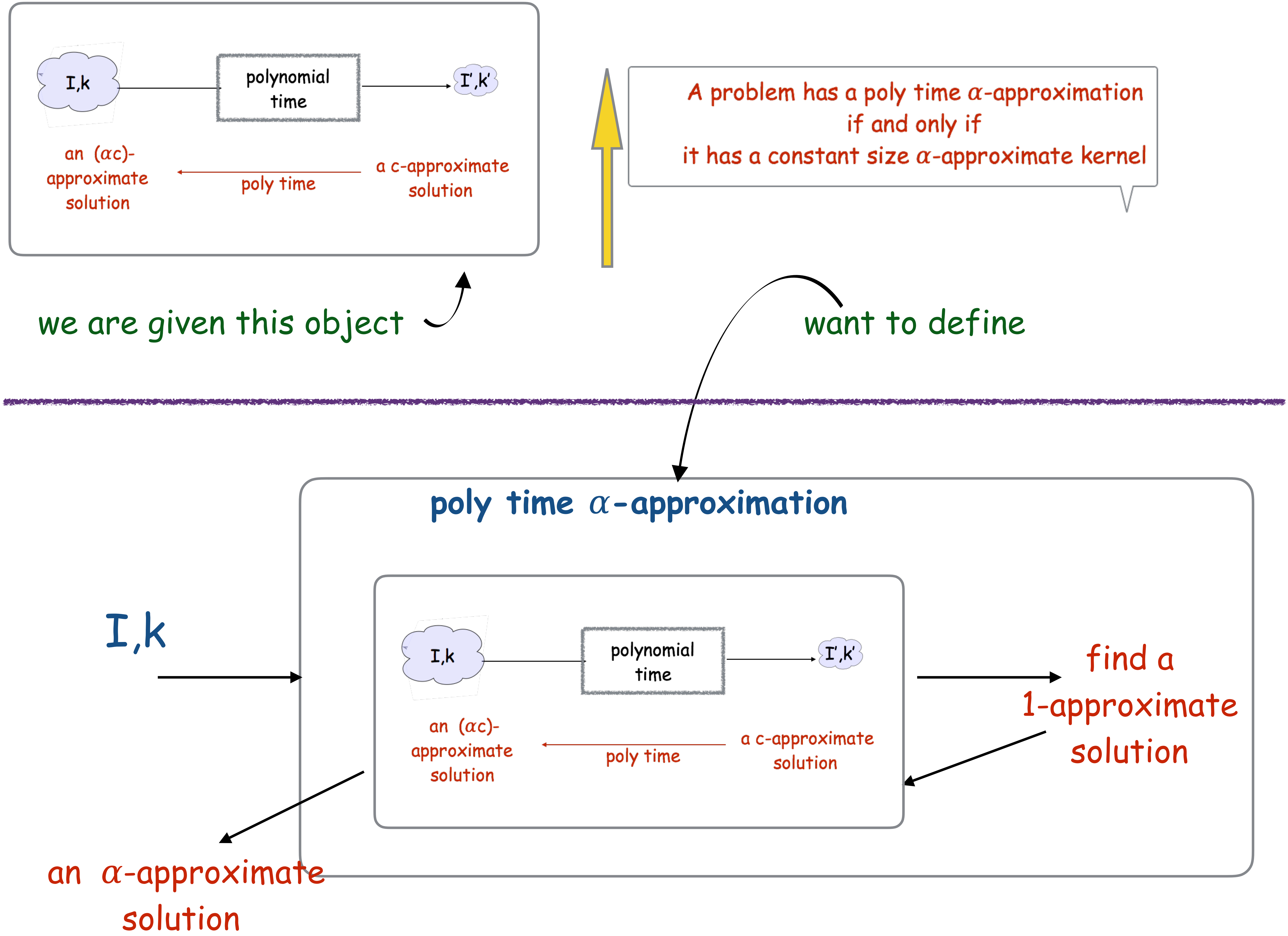


an (αc)-approximate solution

run poly time α-approximation on (I,k)

a c-approximate solution





I,k
polynomial time
I',k'
an (αc)-approximate solution
poly time
a c-approximate solution
A problem has a poly time α-approximation
if and only if
it has a constant size α-approximate kernel
we are given this object
want to define
poly time α-approximation
I,k
I,k
polynomial time
I',k'
an (αc)-approximate solution
poly time
a c-approximate solution
find a 1-approximate solution
an α-approximate solution

α-approximate Kernelization

A problem has a poly time α-approximation
if and only if
it has a constant size α-approximate kernel

want to design α-approximate kernel where

α beats best known approx bound

size beats best known kernel bound

Max SAT

Given a CNF formula, what is the maximum number of clauses which can be satisfied?

APX-hard
[BGLR 93]

no kernel of size poly(n)
[Fortnow Santhanam, 08]

want to design α-approximate kernel where

α=(1-ϵ)

size is polynomial

APX-hard
[BGLR 93]

Max SAT

no kernel of size poly(n)
[FS 08]

Set d=log $(2/\epsilon)$

if at most $\epsilon m/2$ clauses have size at most d

then a random assignment will leave at most $\epsilon m/2+m/2^d = \epsilon m$ clauses unsatisfied

at least $(1-\epsilon)m$ clauses satisfied!

otherwise

$\epsilon m/2 <=$ # small clauses $<= n^{\log(2/\epsilon)}$

$m=O(n^{\log(2/\epsilon)})$

APX-hard
[BGLR 93]

Max SAT

no kernel of size poly(n)
[FS 08]

Set $d=\log(2/\epsilon)$

at least $(1-\epsilon)m$ clauses satisfied!

$m=O(n^{\log(2/\epsilon)})$

find a $(1-\epsilon)$-approximation **OR** already have an $O(n^c)$ kernel

APX-hard
[BGLR 93]

Max SAT

no kernel of size poly(n)
[FS 08]

Set $d=\log(2/\epsilon)$

at least $(1-\epsilon)m$ clauses satisfied!

$m=O(n^{\log(2/\epsilon)})$

find a $(1-\epsilon)$-approximation

OR

already have an $O(n^c)$ kernel

a $(1-\epsilon)$-approximate kernel of size $O(n^c)$

For free!

Problems parameterized by solution value

Problems parameterized by solution value

What is k in this context? k is a threshold on the size of solutions we want to output

Connected Vertex Cover

A subset of vertices such that EVERY edge of G is incident on some vertex in this subset.

Given a graph G, find a smallest vertex cover which induces a connected subgraph?

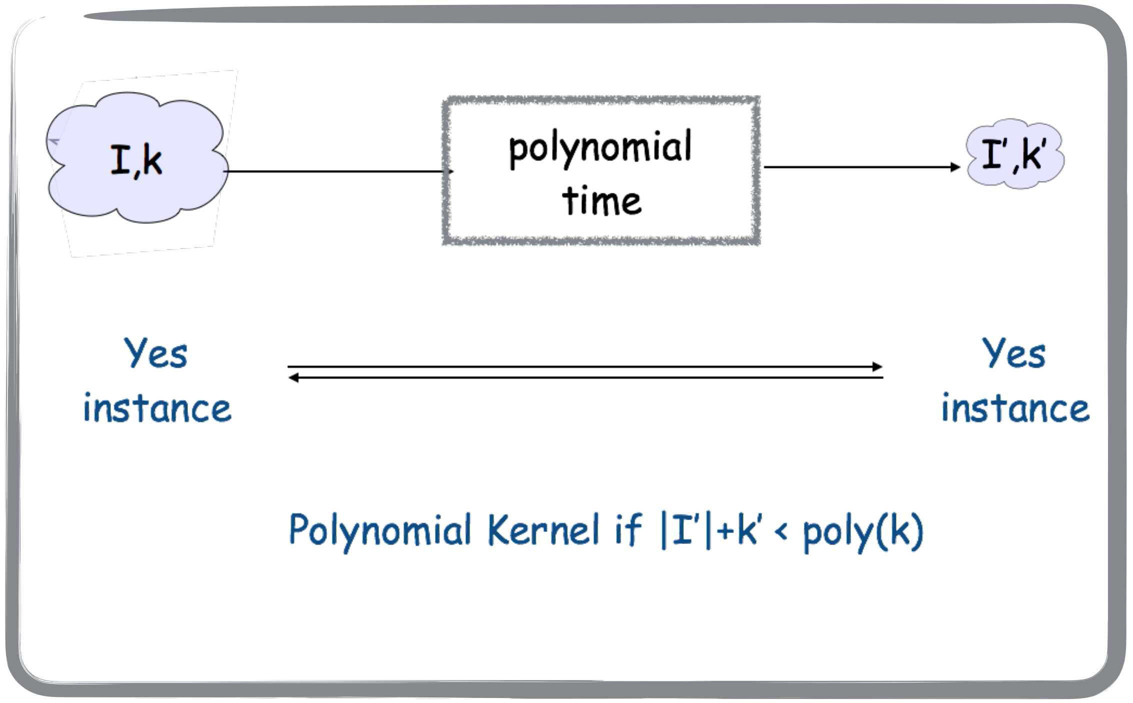
2-approximable [AHH 93]

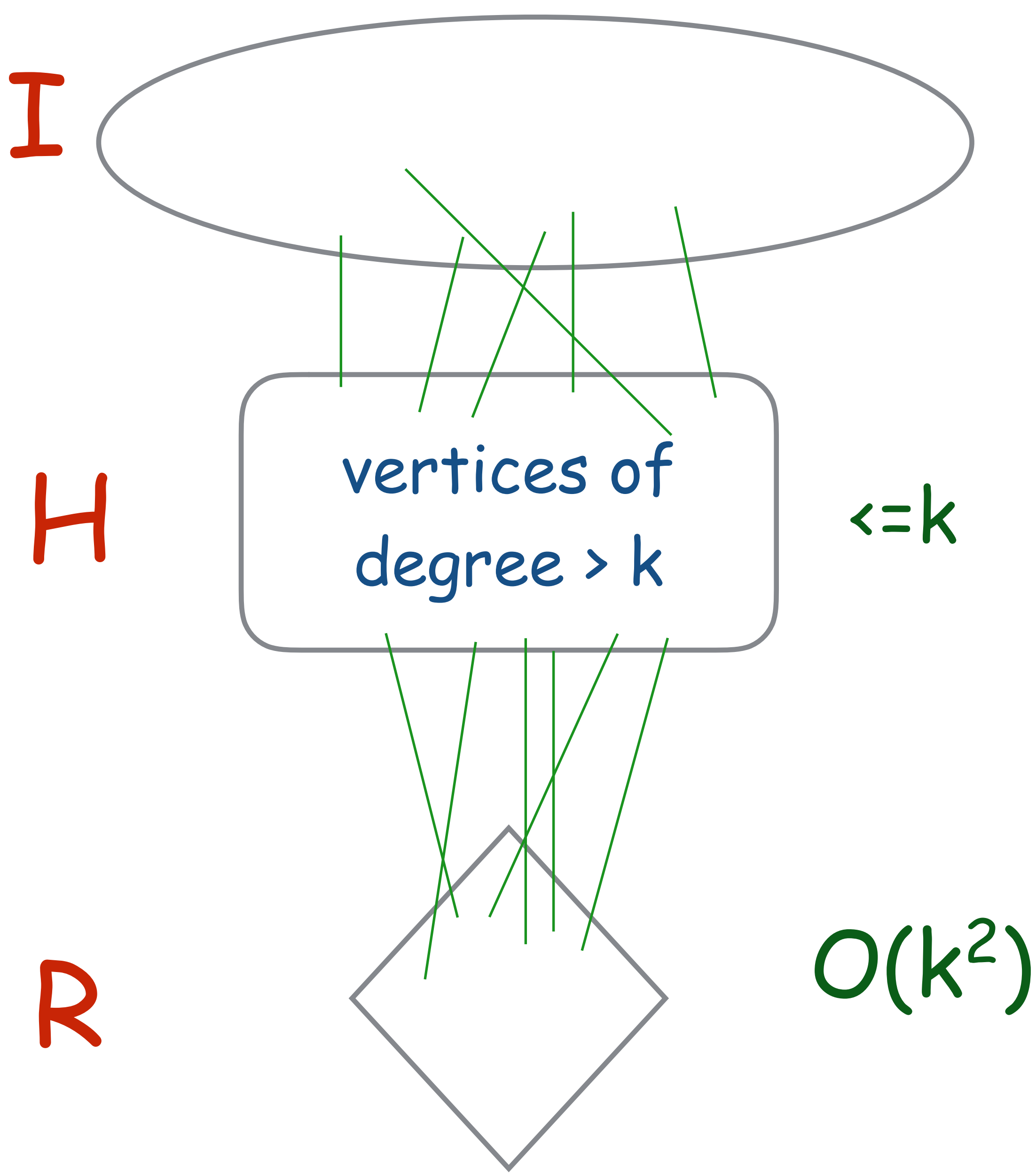
2^k kernel

no $(2-\epsilon)$ approximation under UGC [KR 08]

no polynomial kernel [DLS 09]

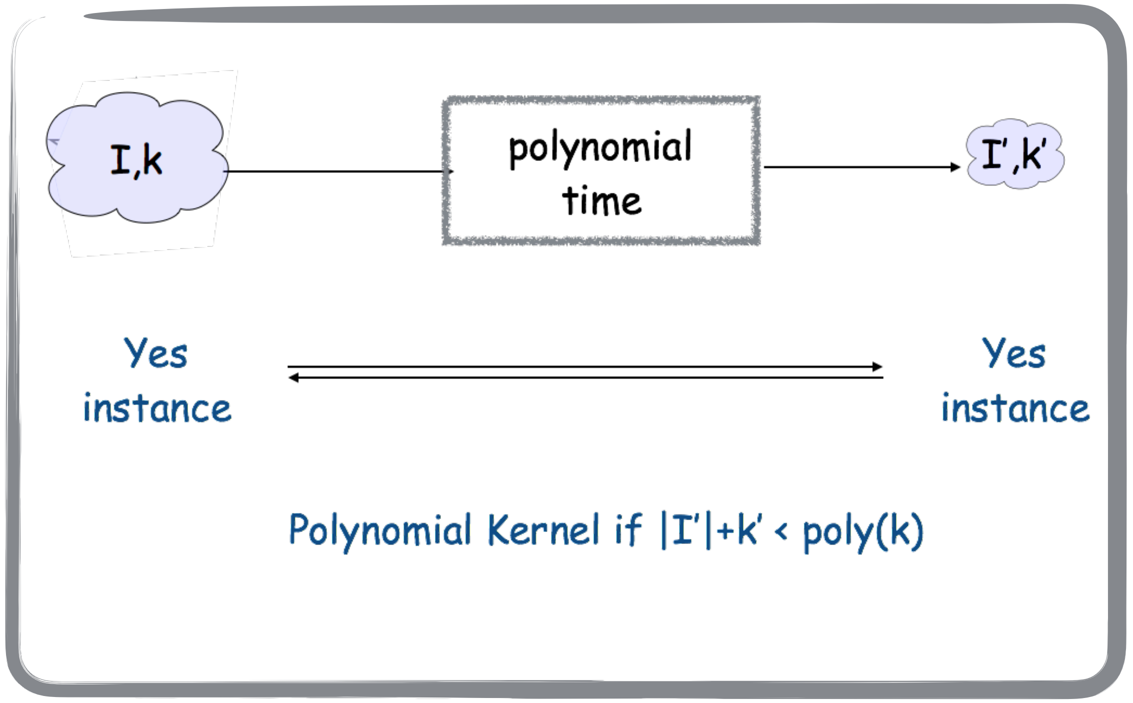
Vertex Cover



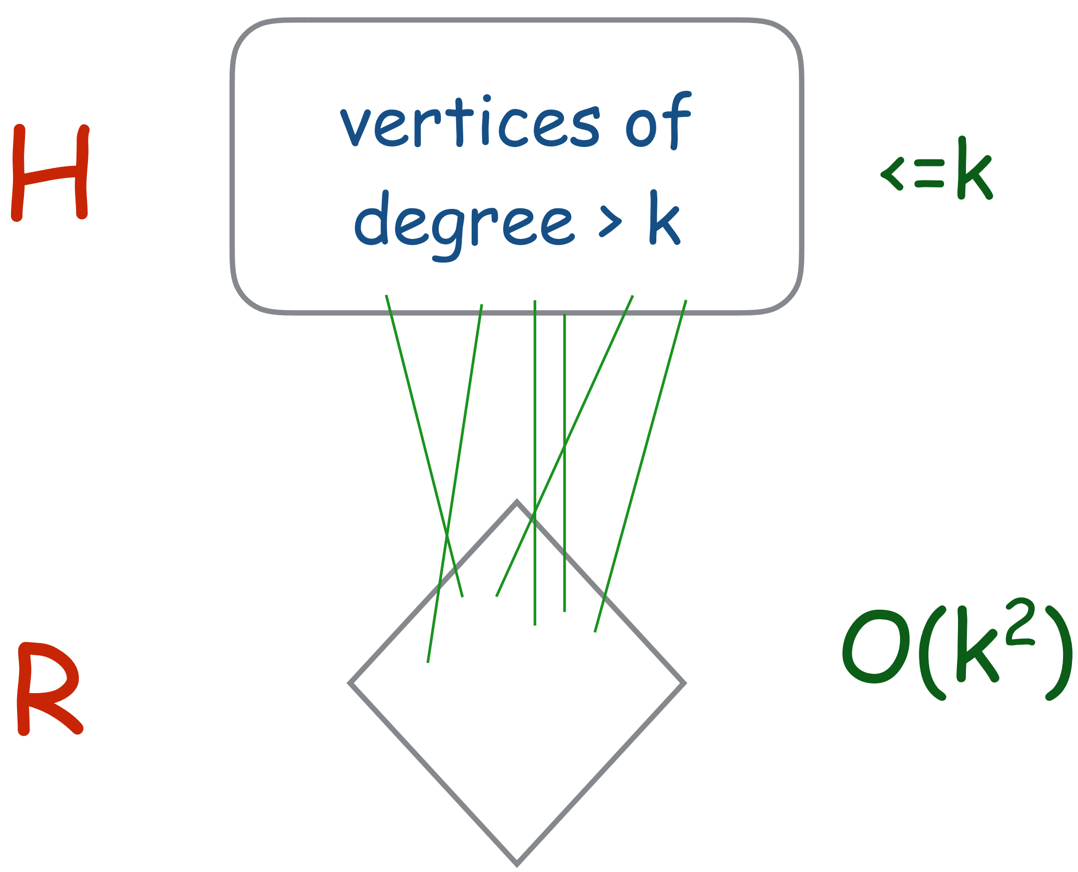


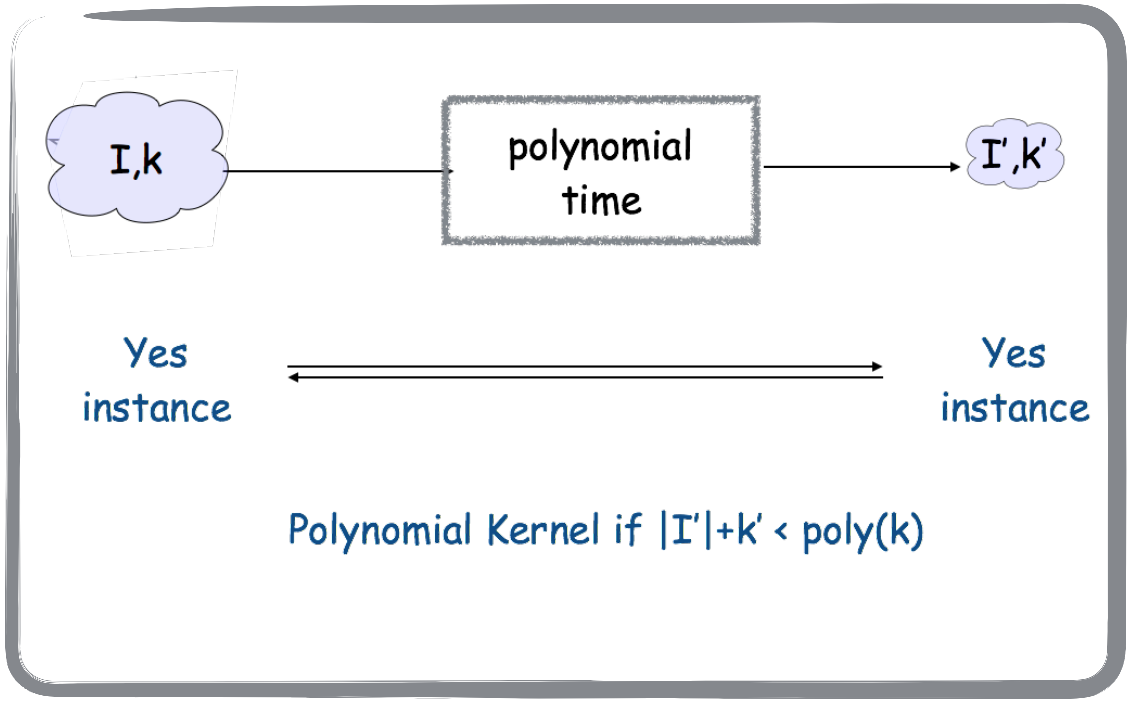
- **H** = vertices of degree at least k+1
- **R** = vertices with at least one neighbor not in H.
- **I** = remaining vertices, have all neighbors in H, must be independent.

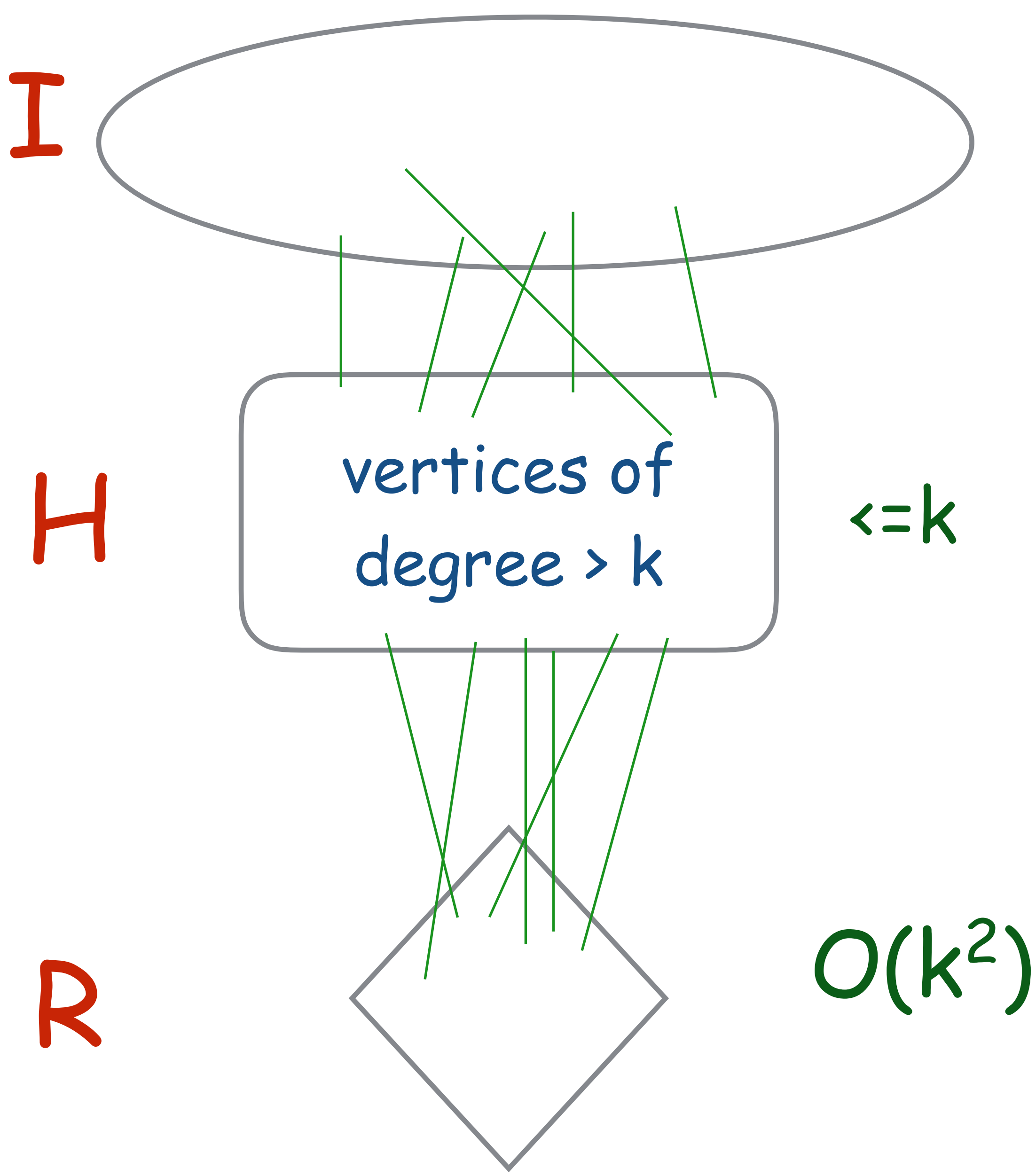
Vertex Cover



- H = vertices of degree at least k+1
- R = vertices with at least one neighbor not in H.
- I = remaining vertices, have all neighbors in H, must be independent.



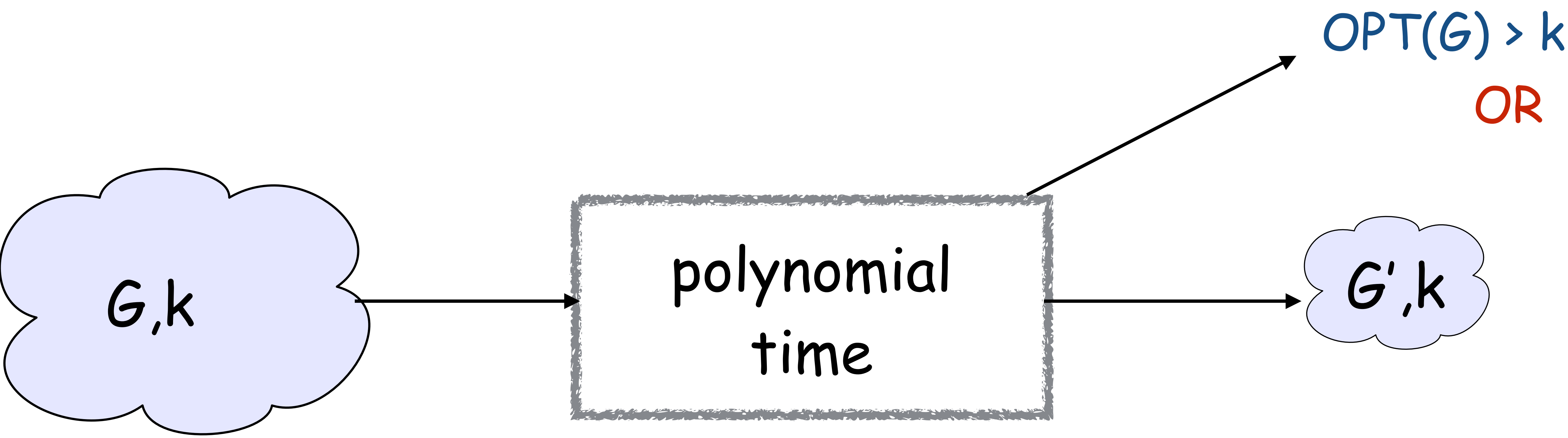




CONNECTED Vertex Cover

- H = vertices of degree at least k+1
- R = vertices with at least one neighbor not in H.
- I = remaining vertices, have all neighbors in H, must be independent.

Connected Vertex Cover

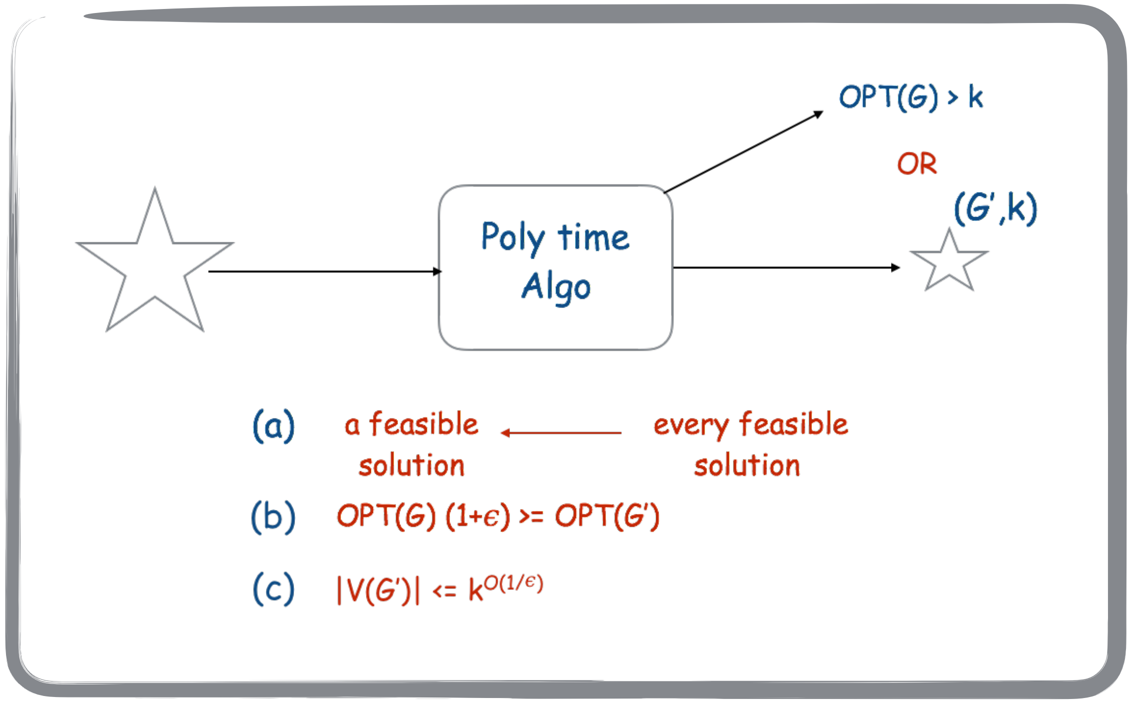


(a) a feasible solution ← every feasible solution

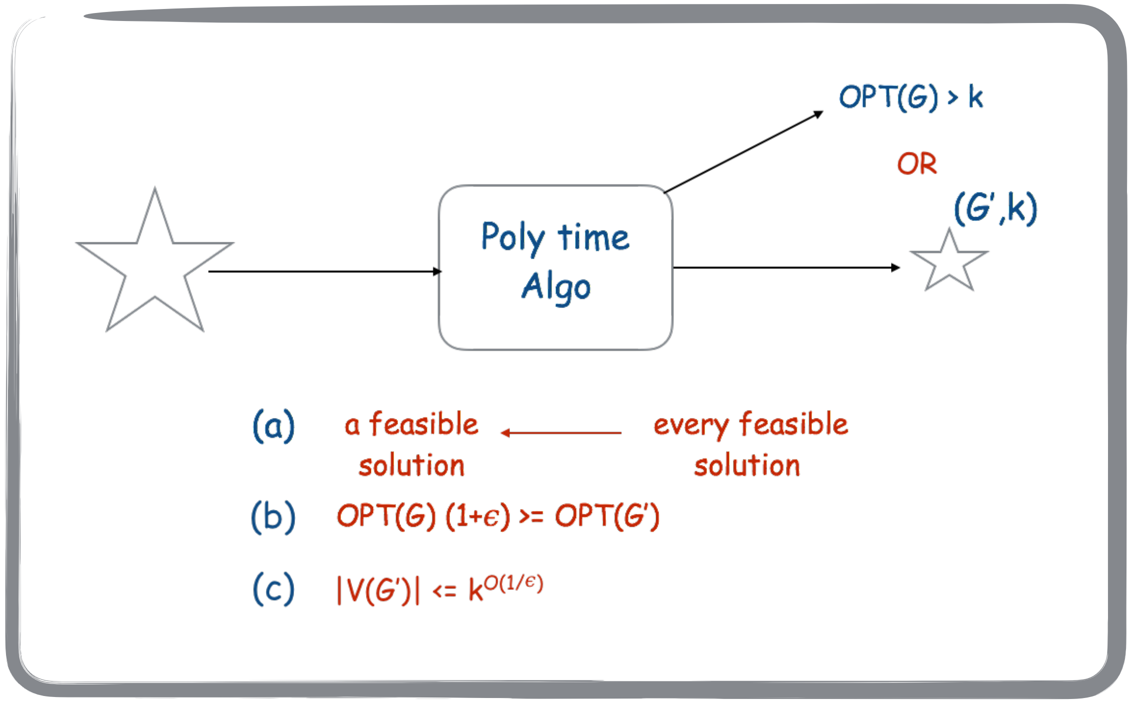
(b) OPT(G) $(1+\epsilon)$ >= OPT(G')

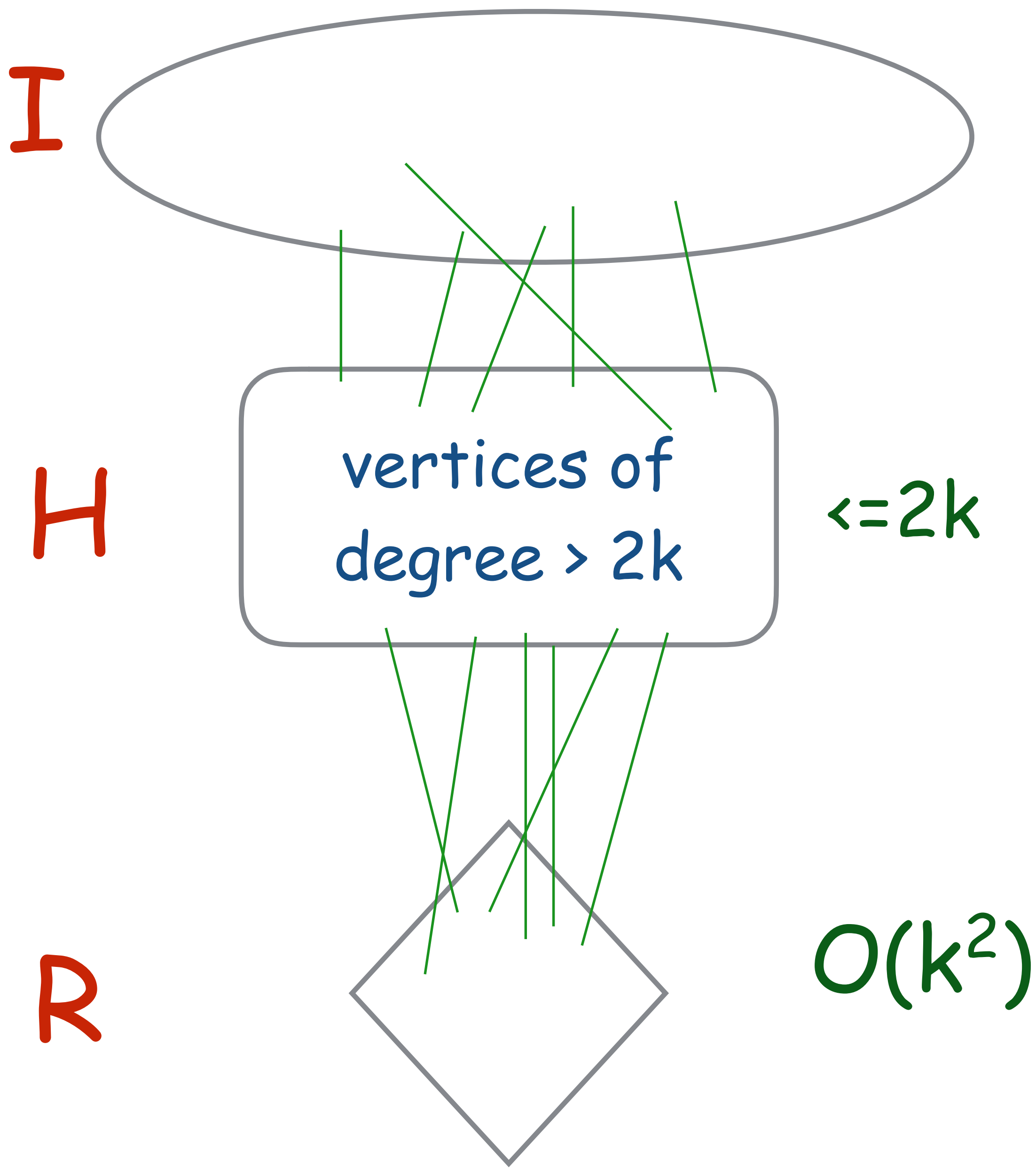
(c) $|V(G')| <= k^{O(1/\epsilon)}$

Hint: Somehow store interesting solutions within $k^{O(1/\epsilon)}$ vertices.

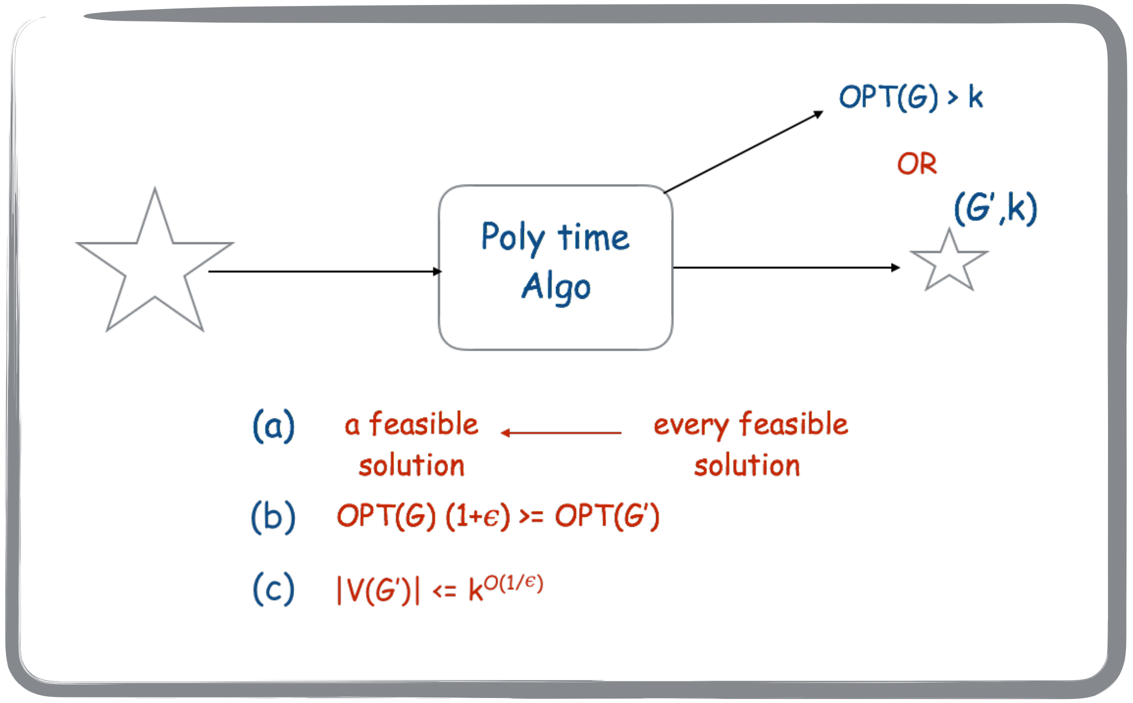


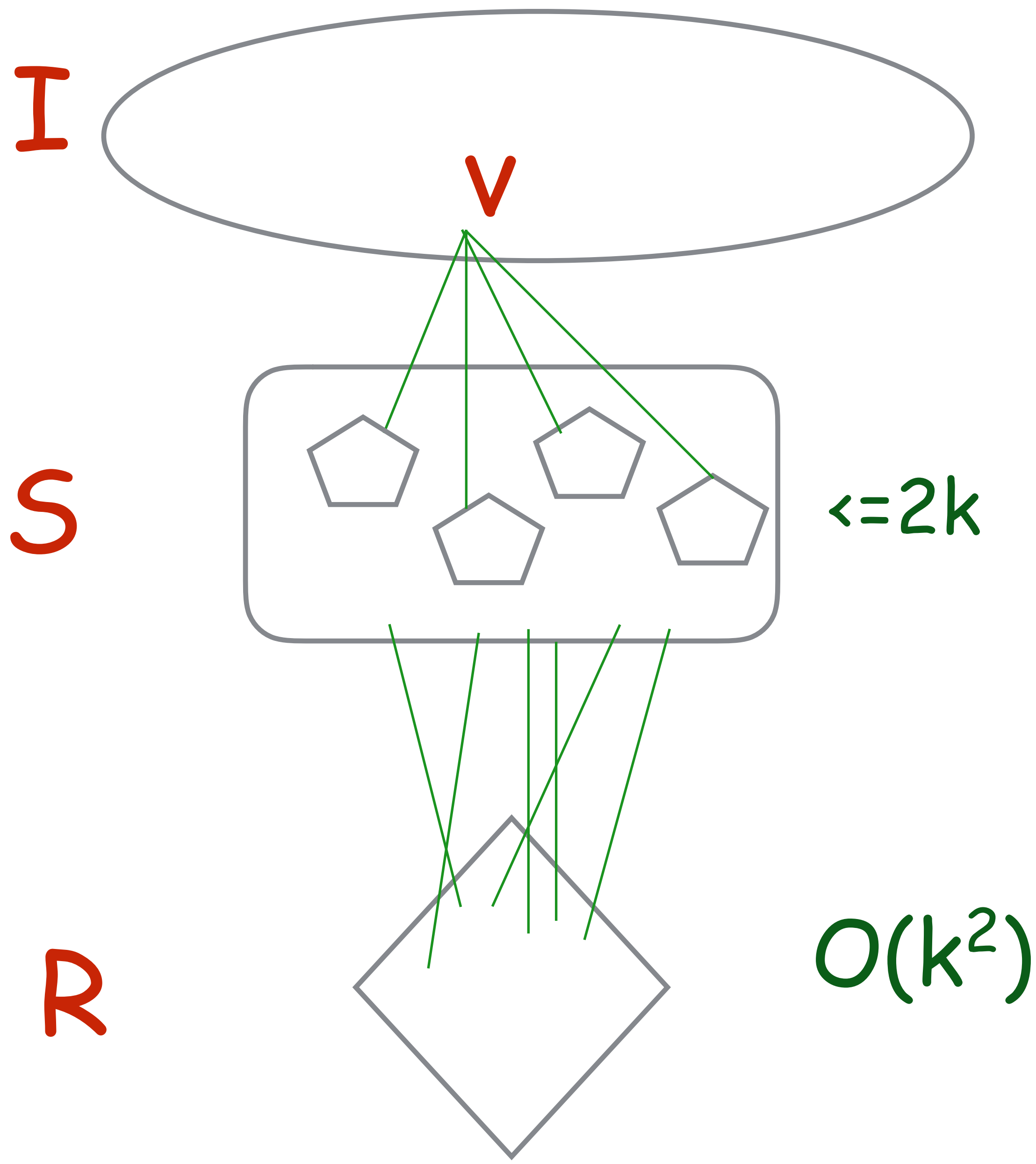
- Run the 2-approximation algorithm.
- If output>2k, then say OPT(G)>k and stop.
- Otherwise, OPT(G)<=2k



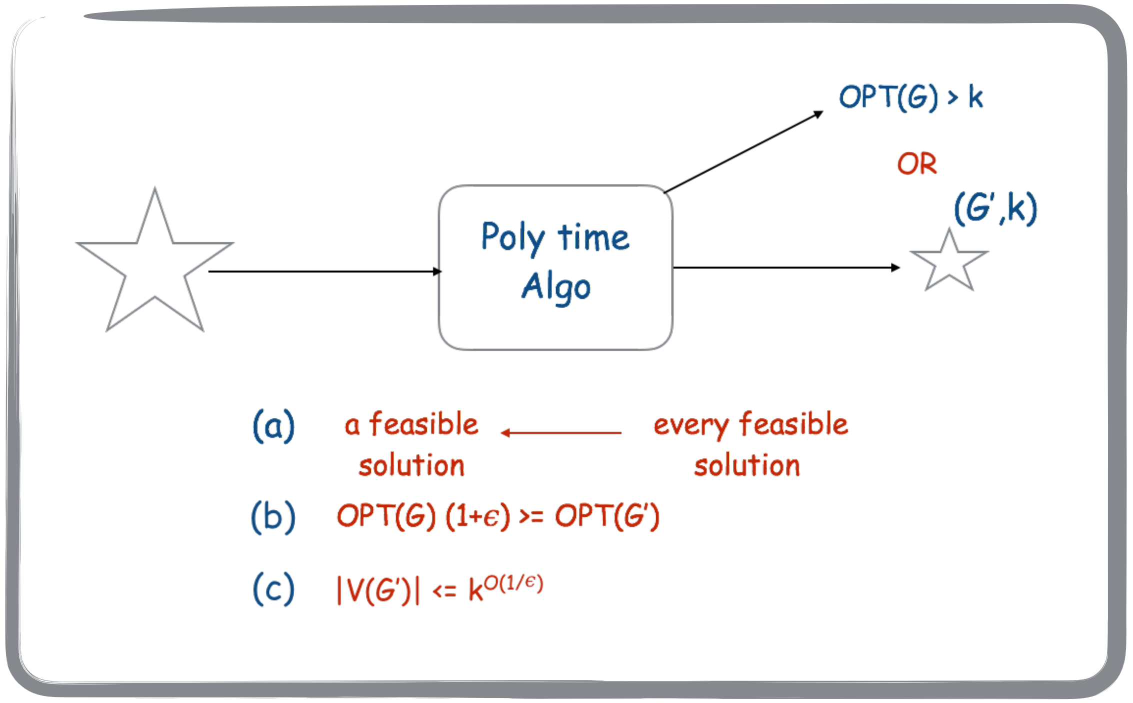


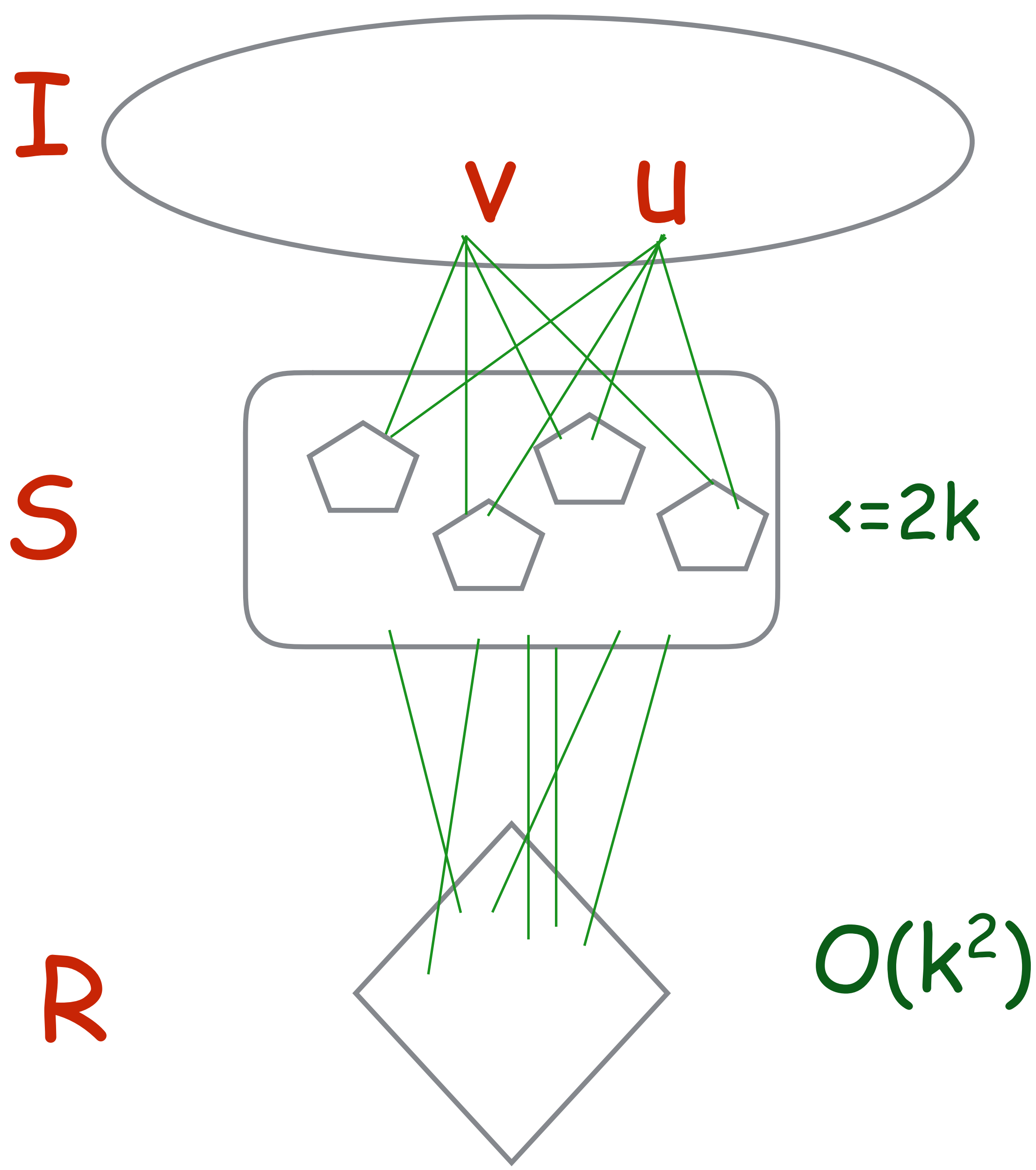
- H = vertices of degree at least 2k+1
- R = vertices incident on at least one edge not incident on H.
- I = remaining vertices, must be independent.
- add a pendant to vertices in H.





- S:= H, $d=2/\epsilon$
- As long as there is a vertex v in I seeing >=d components of G[S], set S:=S ∪ {v}, add pendant to v.
- Procedure stops after <= $\epsilon/2$ |H| <= ϵ OPT(G) steps
- So, OPT(G') <= OPT(G) $(1+\epsilon)$
- What about size of G'?





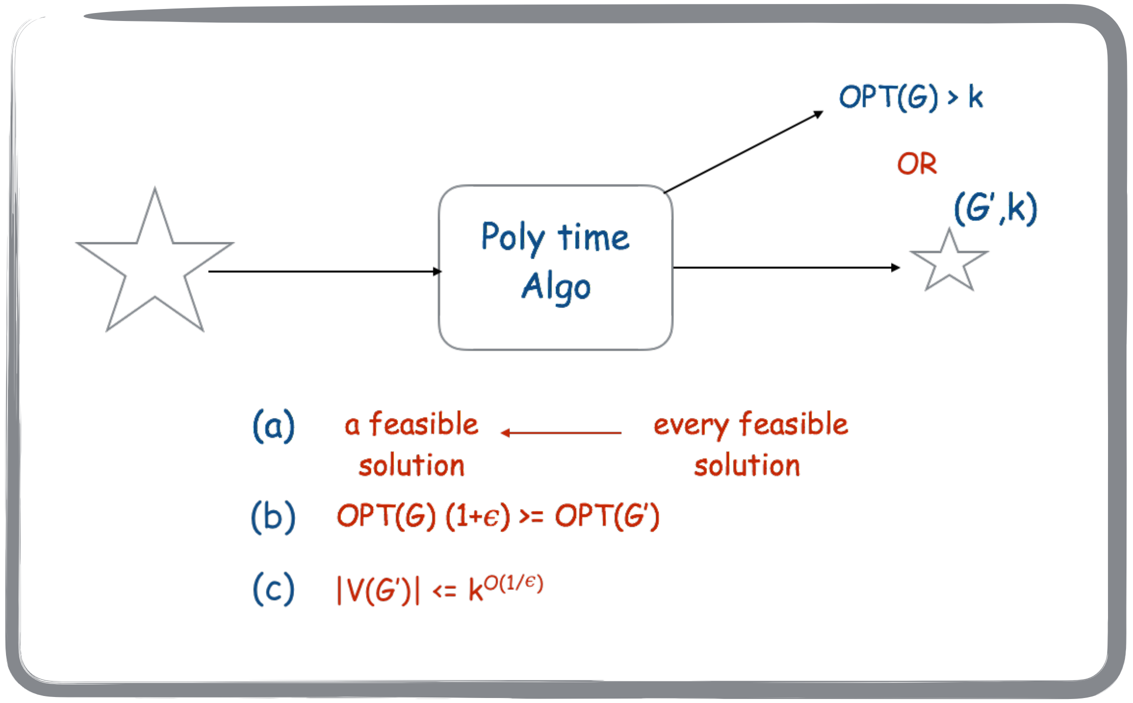
- if u and v have the same neighbouring components in G[S] just delete one.

- Finally, we are left with

$$|I| \leq k^{O(d)} = k^{O(1/\epsilon)}$$

- Size bound holds.

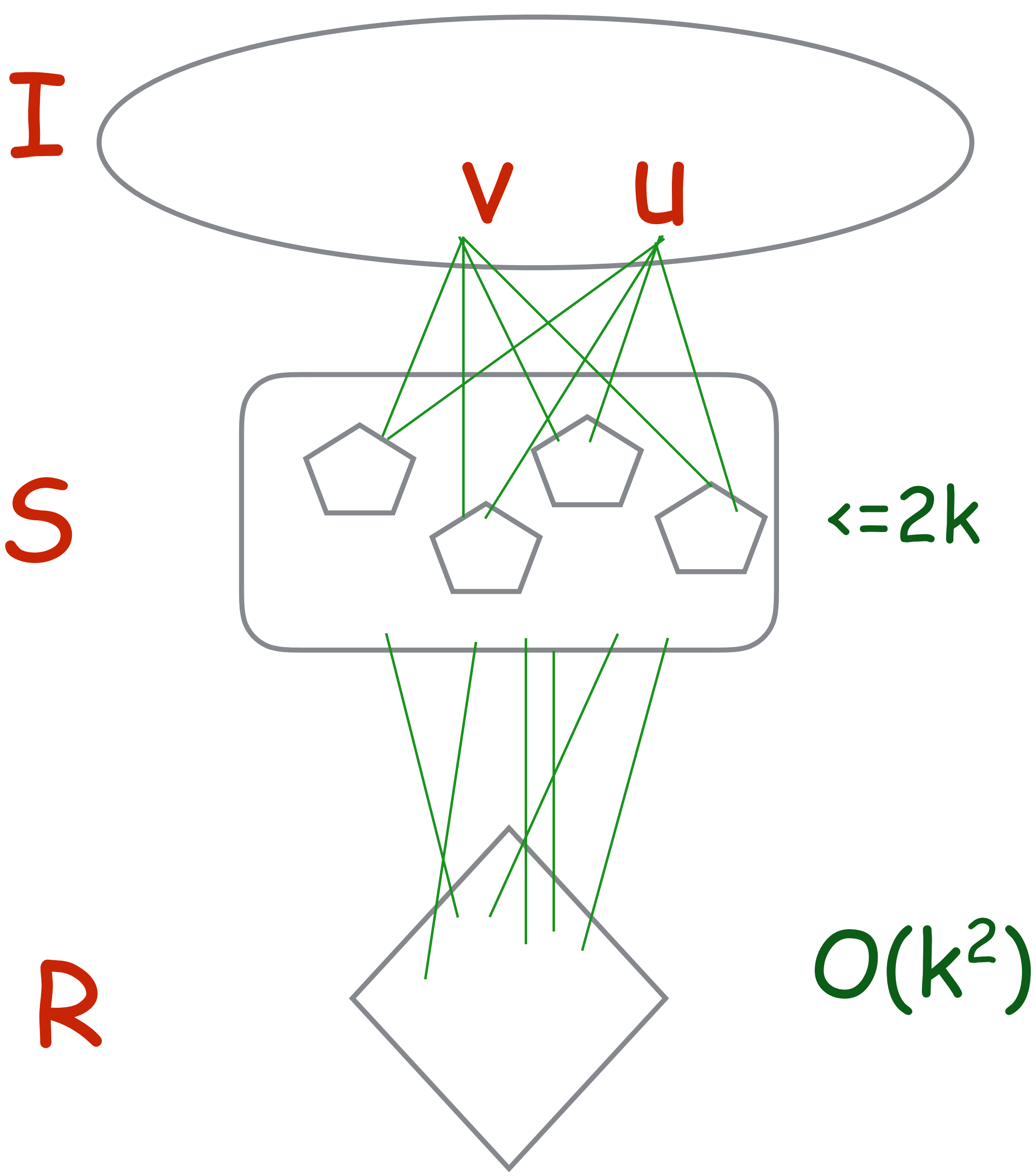
- (1+ϵ)-approximate kernel of size $k^{O(1/\epsilon)}$ for Conn. Vertex Cover.



Open Problem:

Is there a (1+ϵ)-approx. kernel of size $f(1/\epsilon)\ k^{O(1)}$

Efficient PSAKS?



- (1+ϵ)-approximate kernel of size $k^{O(1/\epsilon)}$ for Conn. Vertex Cover.

Polynomial Size Approximate

Kernelization Scheme

We just saw a PSAKS for connected vertex cover.

Another example: H-hitting set

Given a graph G, find a smallest H-hitting set which induces a connected subgraph?

H is a fixed finite family of finite graphs.

We want to hit every copy of a graph in H, which is in G.

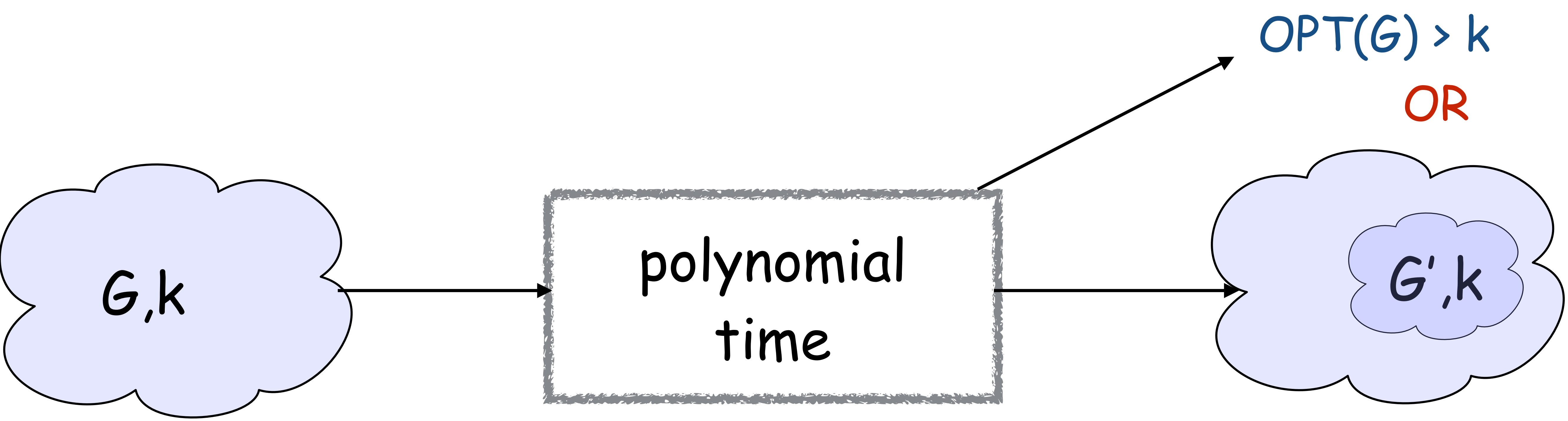
When $H=\{K_2\}$, then we have the Connected Vertex Cover problem.

An approximate kernel for H-hitting set

The connected H-hitting set problem has a $(1+\epsilon)$-approximate kernelization of polynomial size for every $0<\epsilon<1$.

[Eiben, Hermelin, R. 17]

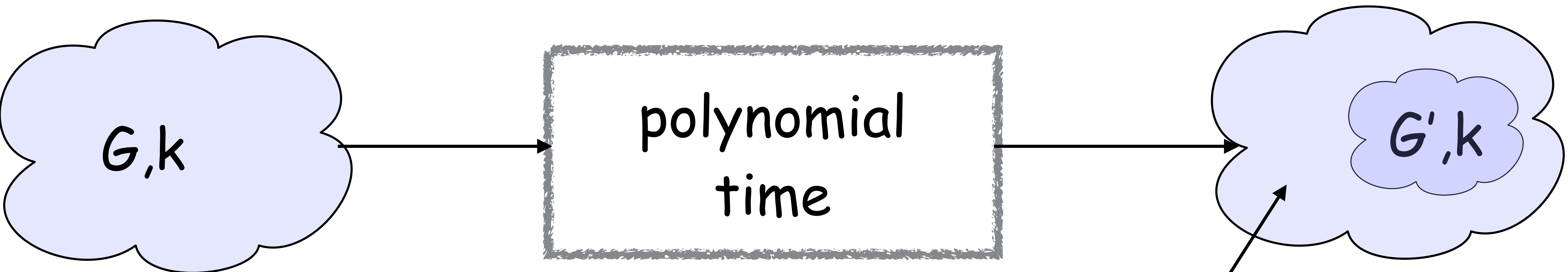
Known kernel for H-hitting set



Every (not necessarily connected) H-hitting set of G' of size at most k is a (not necessarily connected) H-hitting set of G.

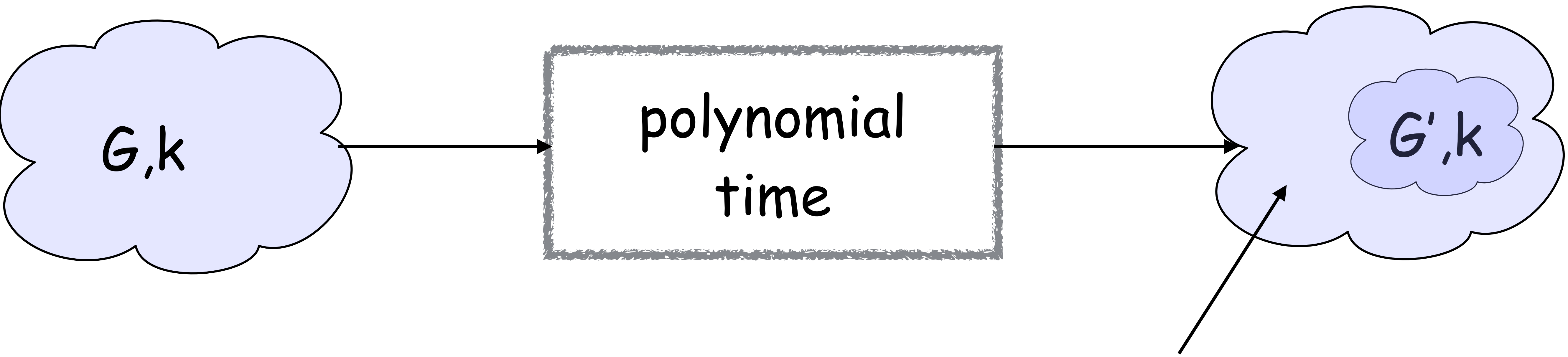
$|V(G')| = k^{O(d)}$ where d=size of largest graph in H

Known kernel for H-hitting set



Like for connected vertex cover, vertices outside G'
are only needed for connectivity

Known kernel for H-hitting set



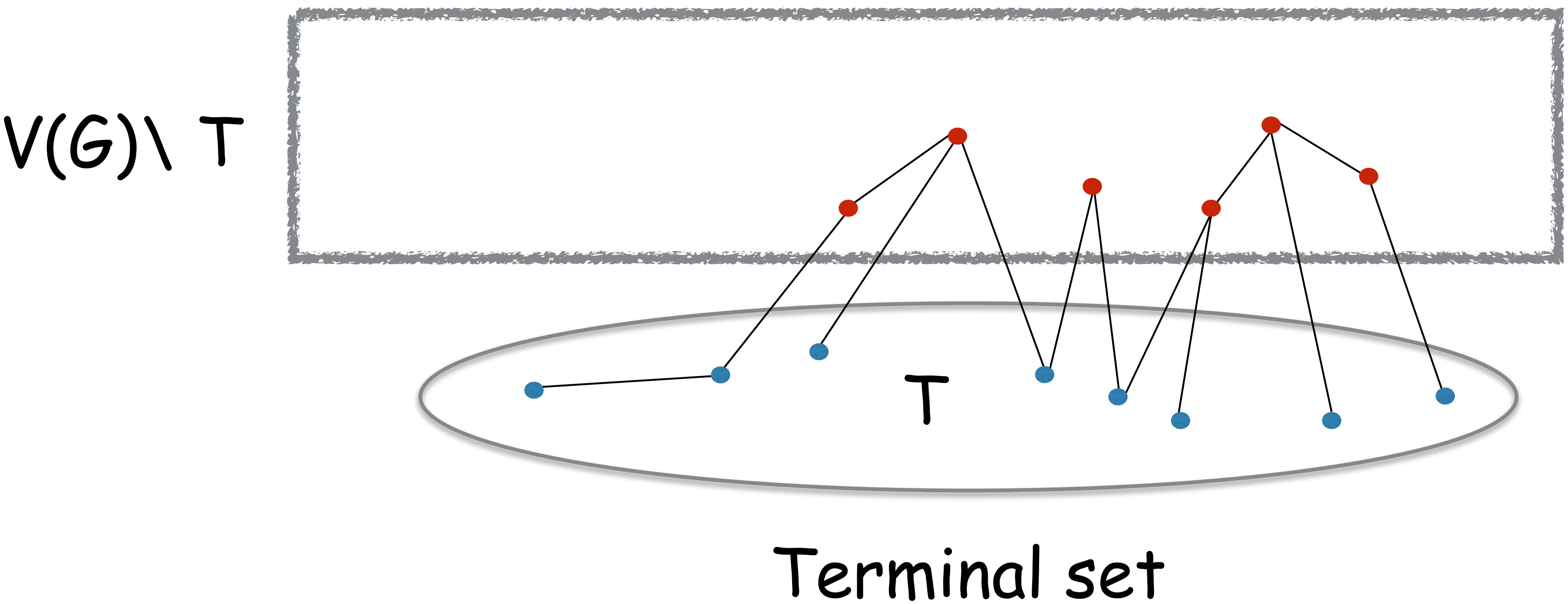
Like for connected vertex cover, vertices outside G' are only needed for connectivity

But again, which vertices? Even worse, $G \backslash G'$ is not so simple as for vertex cover!

Hint: We only want to preserve approximate connectivity between solution vertices in $V(G')$.

Digression: Steiner tree approximation

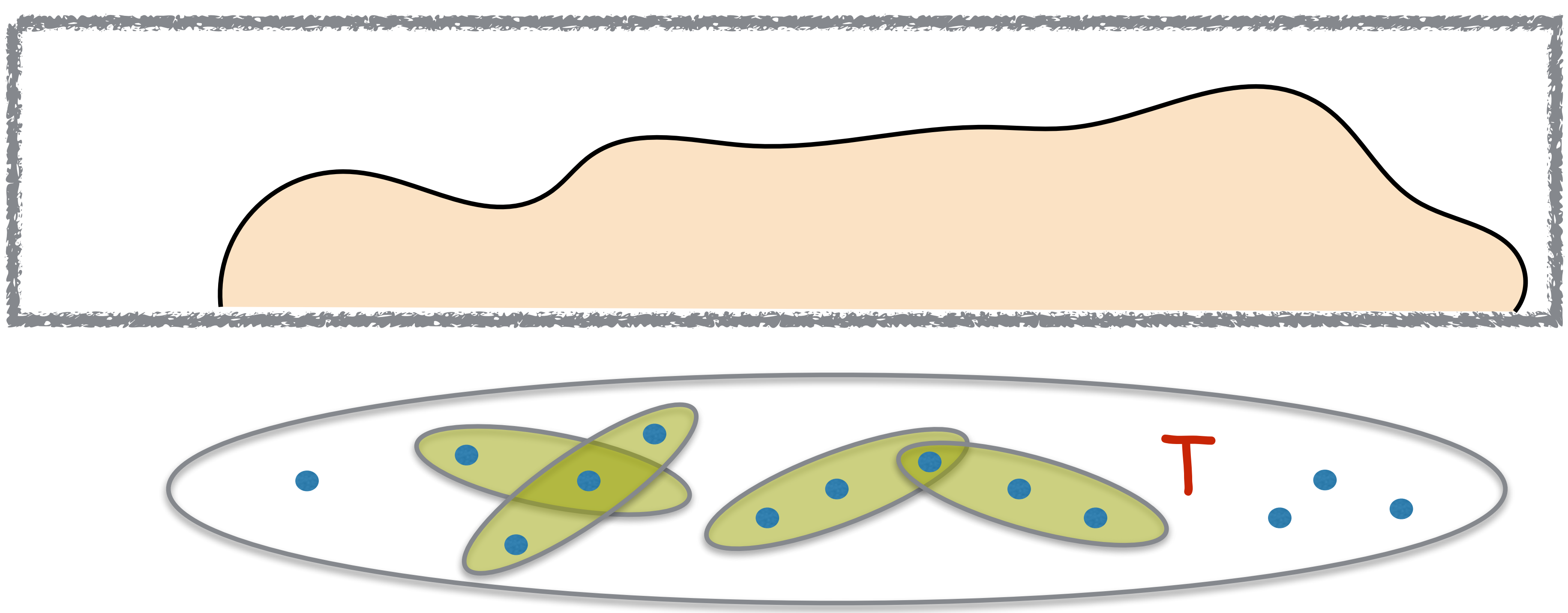
A Steiner tree for a set of terminals T is a connected subgraph of G spanning the vertices in T.



Digression: Steiner tree approximation

For every error parameter $0<\epsilon<1$, there is a $p(1/\epsilon)$ such that any set containing the optimal steiner tree for EVERY $p(1/\epsilon)$-sized subset of T, also contains a $(1+\epsilon)$ approximate Steiner Tree for every $R \subseteq T$.

Borchers and Du, 95

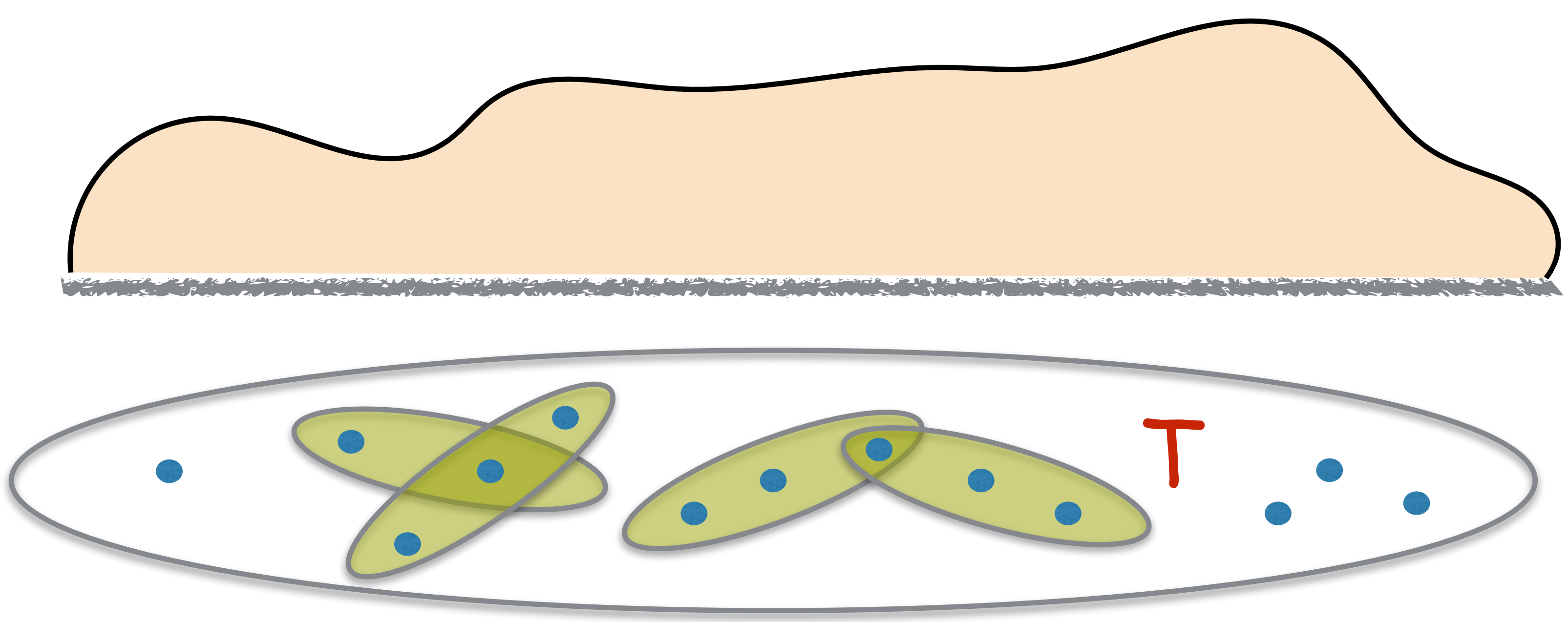


Digression: Steiner tree approximation

For every error parameter $0<\epsilon<1$, there is a $p(1/\epsilon)$ such that any set containing the optimal steiner tree for EVERY $p(1/\epsilon)$-sized subset of T, also contains a $(1+\epsilon)$ approximate Steiner Tree for every $R \subseteq T$.

Borchers and Du, 95

Total number of vertices <= $|T|+q.|T|^{p(1/\epsilon)}$

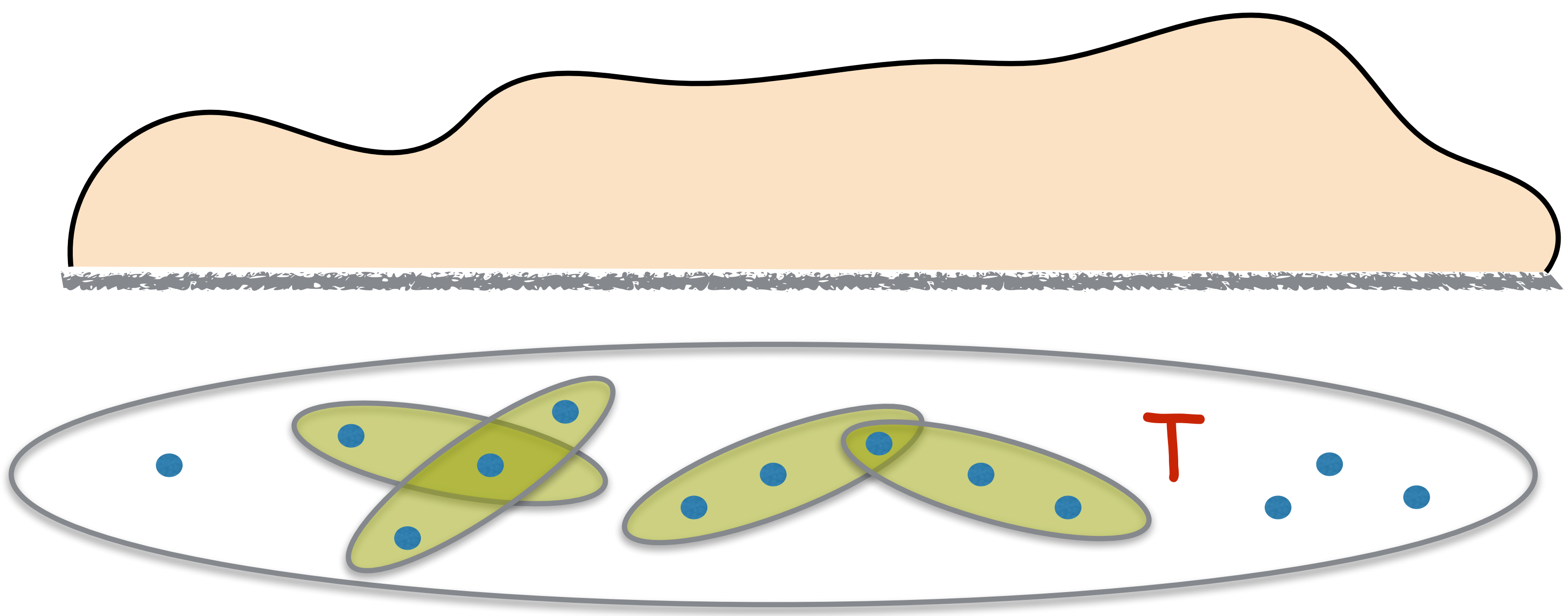


Steiner tree approximation in our setting

For every error parameter $0<\epsilon<1$, there is a $p(1/\epsilon)$ such that any set containing the optimal steiner tree for EVERY $p(1/\epsilon)$-sized subset of T, also contains a $(1+\epsilon)$ approximate Steiner Tree for every $R \subseteq T$.

Borchers and Du, 95

Total number of vertices <= $|T|+q.|T|^{p(1/\epsilon)}$



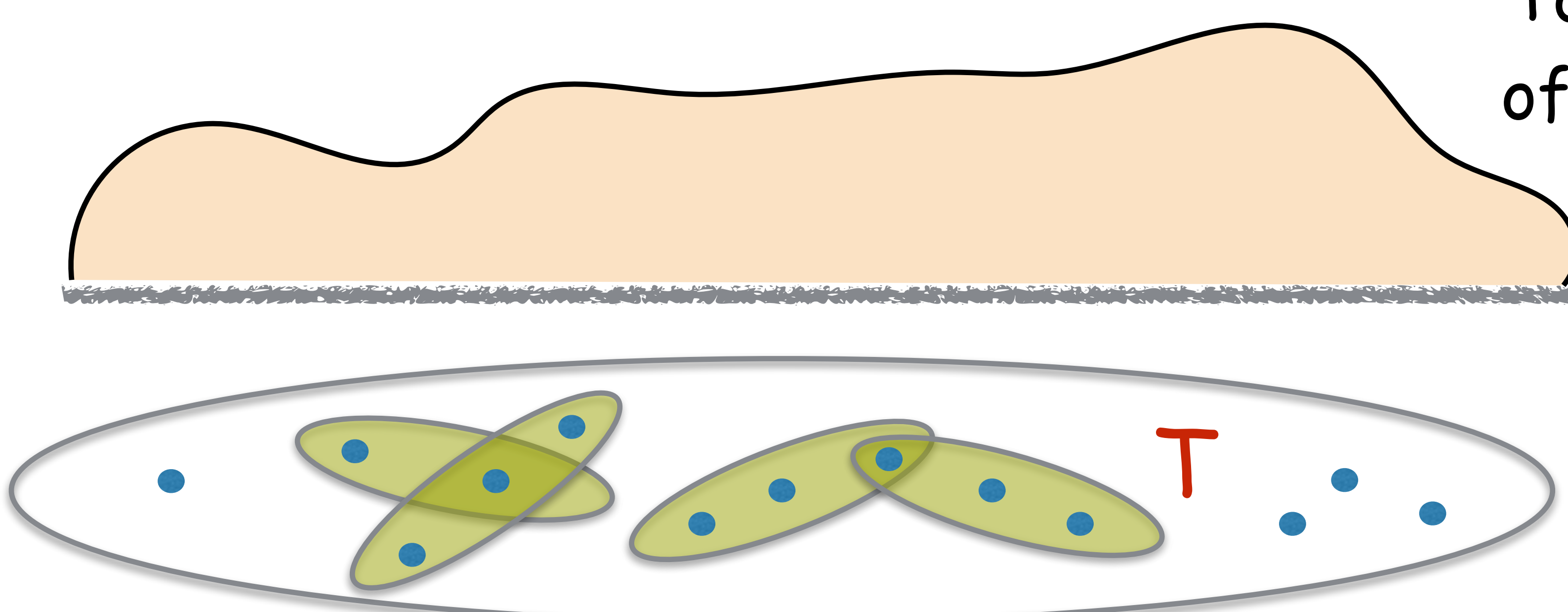
But for us, $q = k$ and $|T|=|V(G')|=k^{O(d)}$

Steiner tree approximation in our setting

For every error parameter $0<\epsilon<1$, there is a $p(1/\epsilon)$ such that any set containing the optimal steiner tree for EVERY $p(1/\epsilon)$-sized subset of T, also contains a $(1+\epsilon)$ approximate Steiner Tree for every $R \subseteq T$.

Borchers and Du, 95

Total number of vertices <= $|T|+q.|T|^{p(1/\epsilon)}$



Total number of vertices <= $k^{O(d\ p(1/\epsilon))}$

But for us, $q = k$ and $|T|=|V(G')|=k^{O(d)}$

Steiner Tree

2^k kernel

1.39-Appx

Steiner Tree (Terminals)

No poly kernel

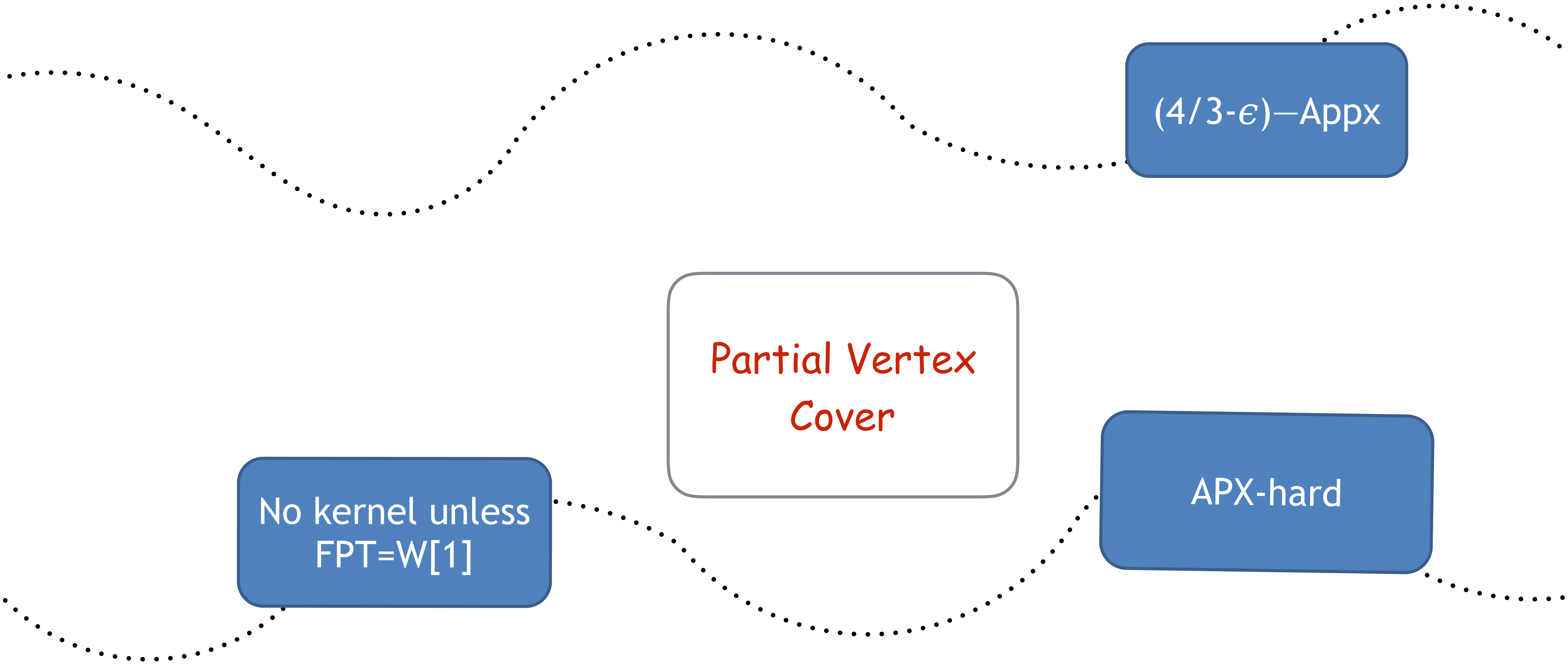
APX-hard

(1+ϵ)-approximate kernel of polynomial size

For every error parameter $0<\epsilon<1$, there is a $p(1/\epsilon)$ such that any set containing the optimal steiner tree for EVERY $p(1/\epsilon)$-sized subset of T, also contains a $(1+\epsilon)$ approximate Steiner Tree for every $R \subseteq T$.

Borchers and Du, 95

Partial Vertex Cover



[Marx, 2008]

Partial Vertex Cover

(4/3-ϵ)—Appx

Partial Vertex Cover

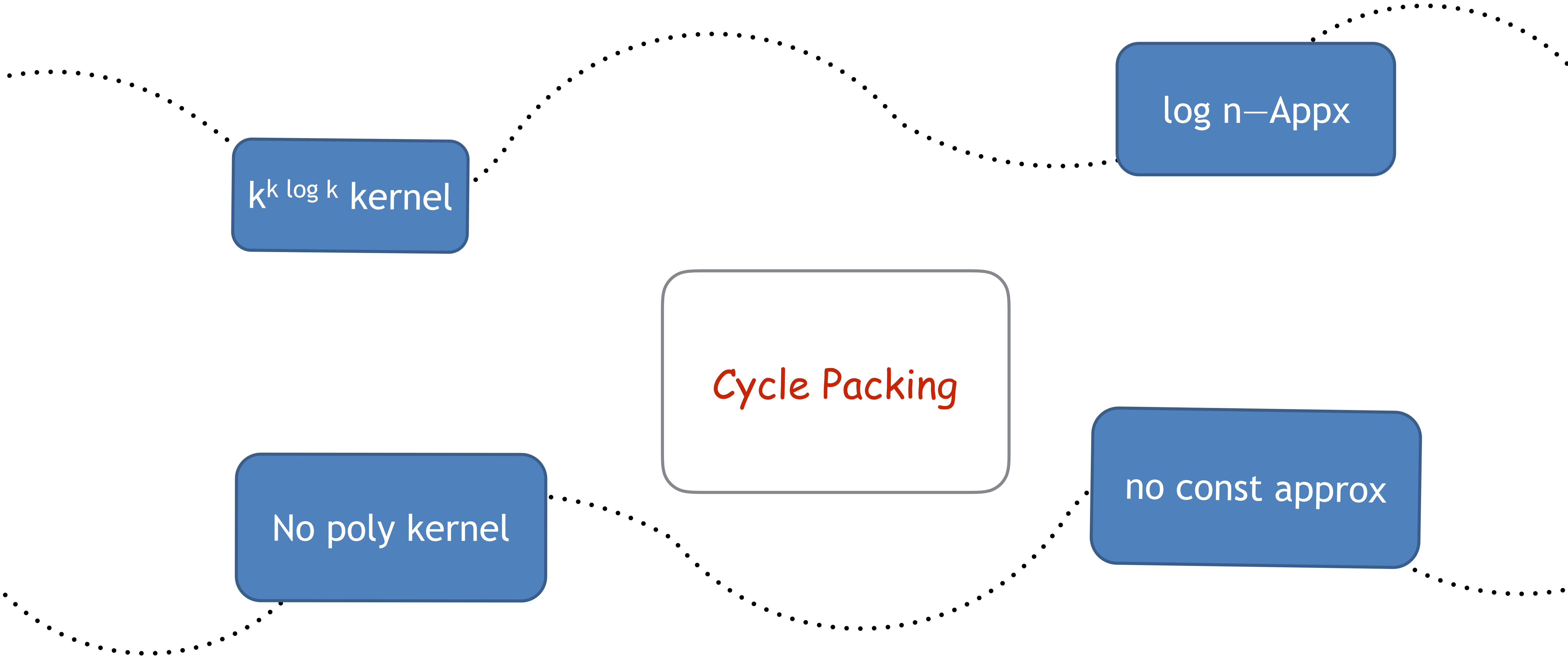
No kernel unless FPT=W[1]

APX-hard

(1+ϵ)-approximate kernel of polynomial size

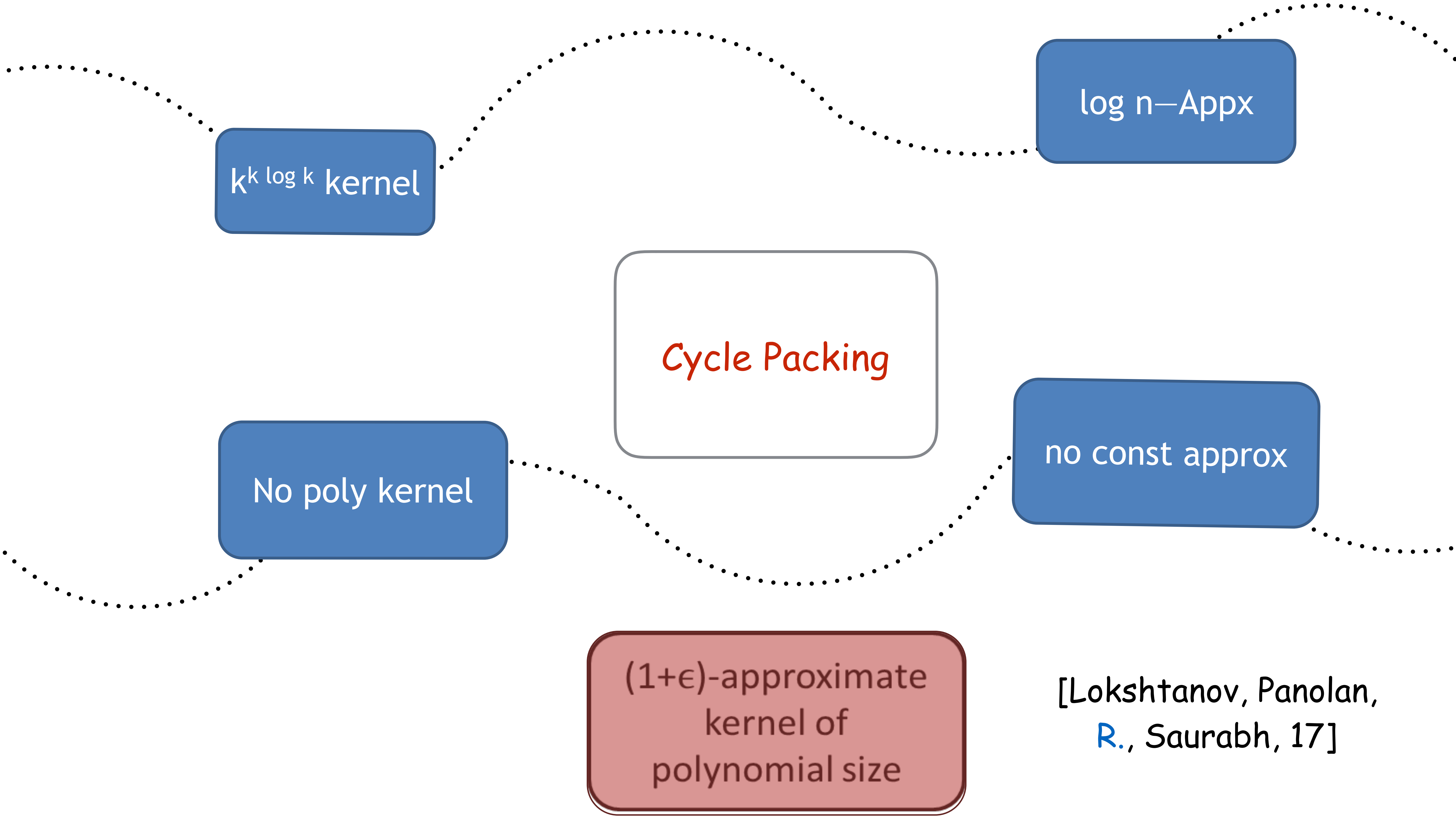
[Marx, 2008]

Cycle Packing



[Lokshtanov, Panolan,
R., Saurabh, 17]

Cycle Packing



[Lokshtanov, Panolan, R., Saurabh, 17]

More approximate kernels

Problem Name	Apx.	Apx. Hardness	Kernel	Apx. Ker. Fact.	Appx. Ker. Size
Connected V.C.	2 [4,52]	$(2-\epsilon)$ [41]	no $k^{\mathcal{O}(1)}$ [22]	$1<\alpha$	$k^{f(\alpha)}$
Cycle Packing	$\mathcal{O}(\log n)$ [51]	$(\log n)^{\frac{1}{2}-\epsilon}$ [33]	no $k^{\mathcal{O}(1)}$ [8]	$1<\alpha$	$k^{f(\alpha)}$
Disjoint Factors	2	no PTAS	no $\lvert\Sigma\rvert^{\mathcal{O}(1)}$ [8]	$1<\alpha$	$\lvert\Sigma\rvert^{f(\alpha)}$
Longest Path	$\mathcal{O}(\frac{n}{\log n})$ [2]	$2^{(\log n)^{1-\epsilon}}$ [39]	no $k^{\mathcal{O}(1)}$ [6]	any α	no $k^{\mathcal{O}(1)}$
Set Cover/n	$\ln n$ [55]	$(1-\epsilon)\ln n$ [47]	no $n^{\mathcal{O}(1)}$ [22]	any α	no $n^{\mathcal{O}(1)}$
Hitting Set/n	$\mathcal{O}(\sqrt{n})$ [48]	$2^{(\log n)^{1-\epsilon}}$ [48]	no $n^{\mathcal{O}(1)}$ [22]	any α	no $n^{\mathcal{O}(1)}$
Vertex Cover	2 [55]	$(2-\epsilon)$ [21,41]	$2k$ [15]	$1<\alpha<2$	$2(2-\alpha)k$ [30]
d-Hitting Set	d [55]	$d-\epsilon$ [20,41]	$\mathcal{O}(k^{d-1})$ [1]	$1<\alpha<d$	$\mathcal{O}((k\cdot\frac{d-\alpha}{\alpha-1})^{d-1})$ [30]
Steiner Tree	1.39 [11]	no PTAS [12]	no $k^{\mathcal{O}(1)}$ [22]	$1<\alpha$	$k^{f(\alpha)}$
OLA/v.c.	$\mathcal{O}(\sqrt{\log n}\log\log n)$ [28]	no PTAS [3]	$f(k)$ [43]	$1<\alpha<2$	$f(\alpha)2^k k^4$
Partial V.C.	$(\frac{4}{3}-\epsilon)$ [27]	no PTAS [49]	no $f(k)$ [35]	$1<\alpha$	$f(\alpha)k^5$

Lot of these problems have played a critical role in the development of kernel lower bound theory

Longest Path

Longest Path

Given a graph G, what is the length of a longest path in G?

Longest Path

Longest Path

No poly kernel

no const approx

No c-approximate poly kernel unless NP in coNP/Poly

Longest Path

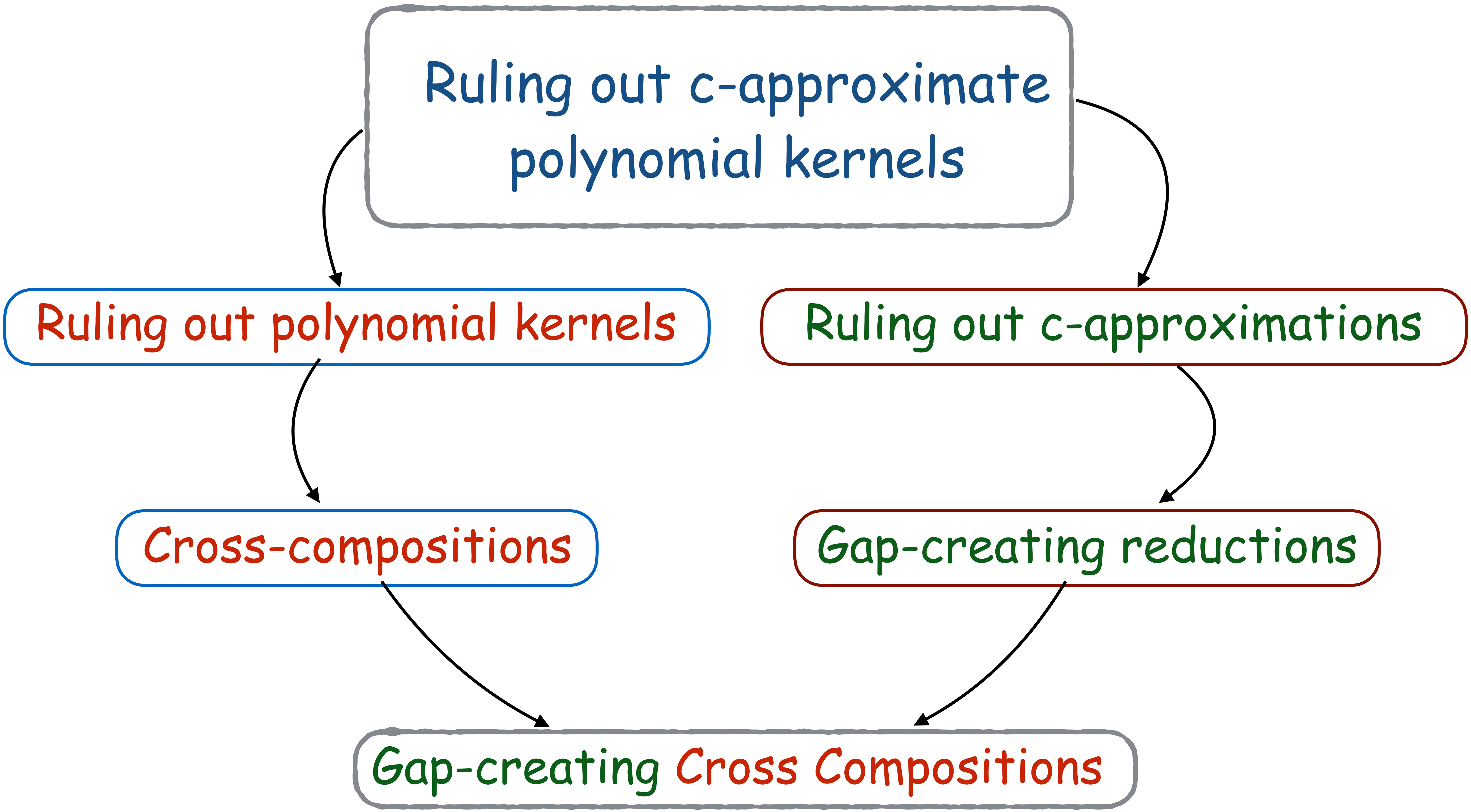
Longest Path

No poly kernel

no const approx

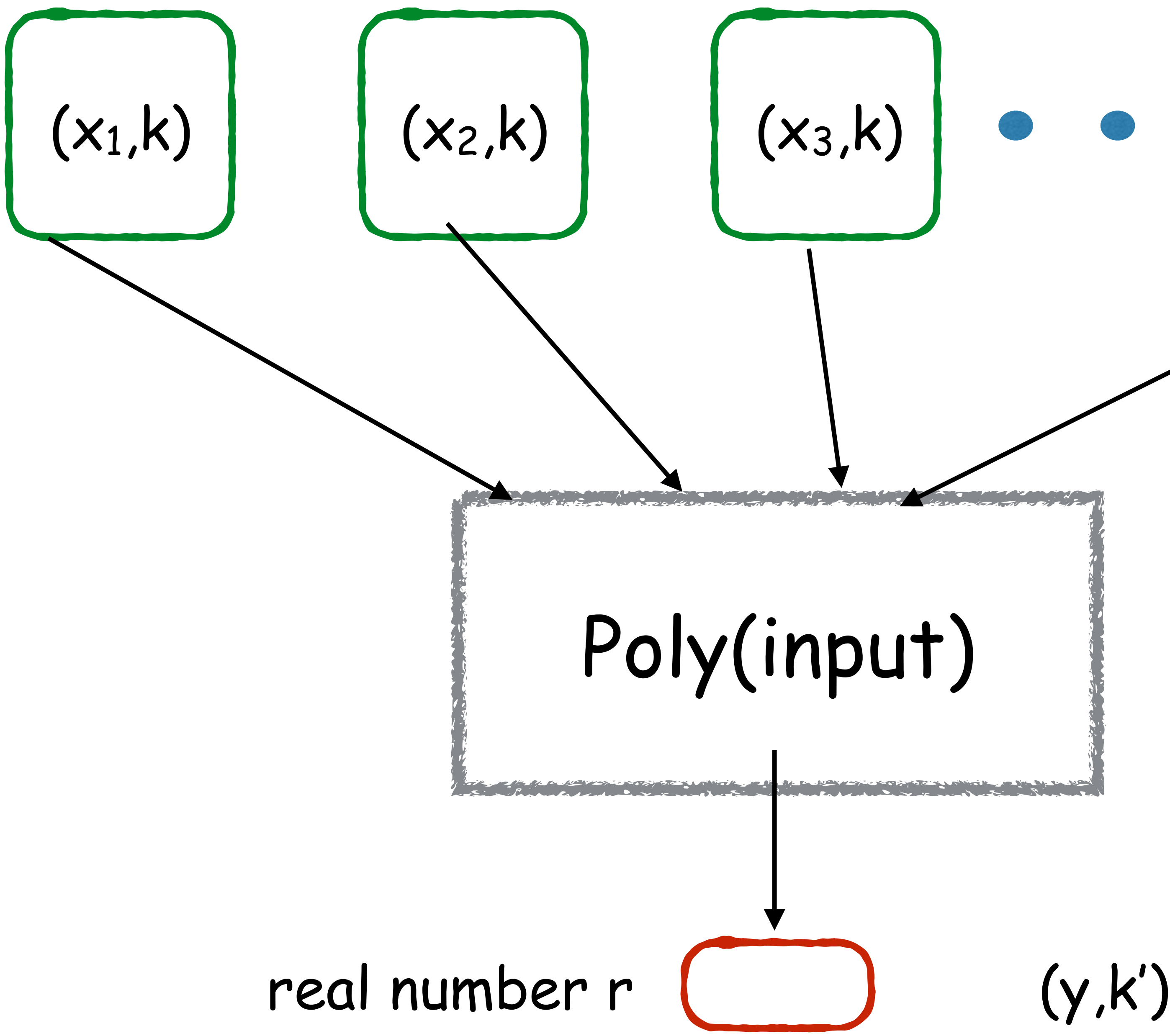
No c-approximate poly kernel unless NP in coNP/Poly

Lower Bounds



Lower Bounds

Gap-creating Cross Compositions



MAXIMIZATION PROBLEM

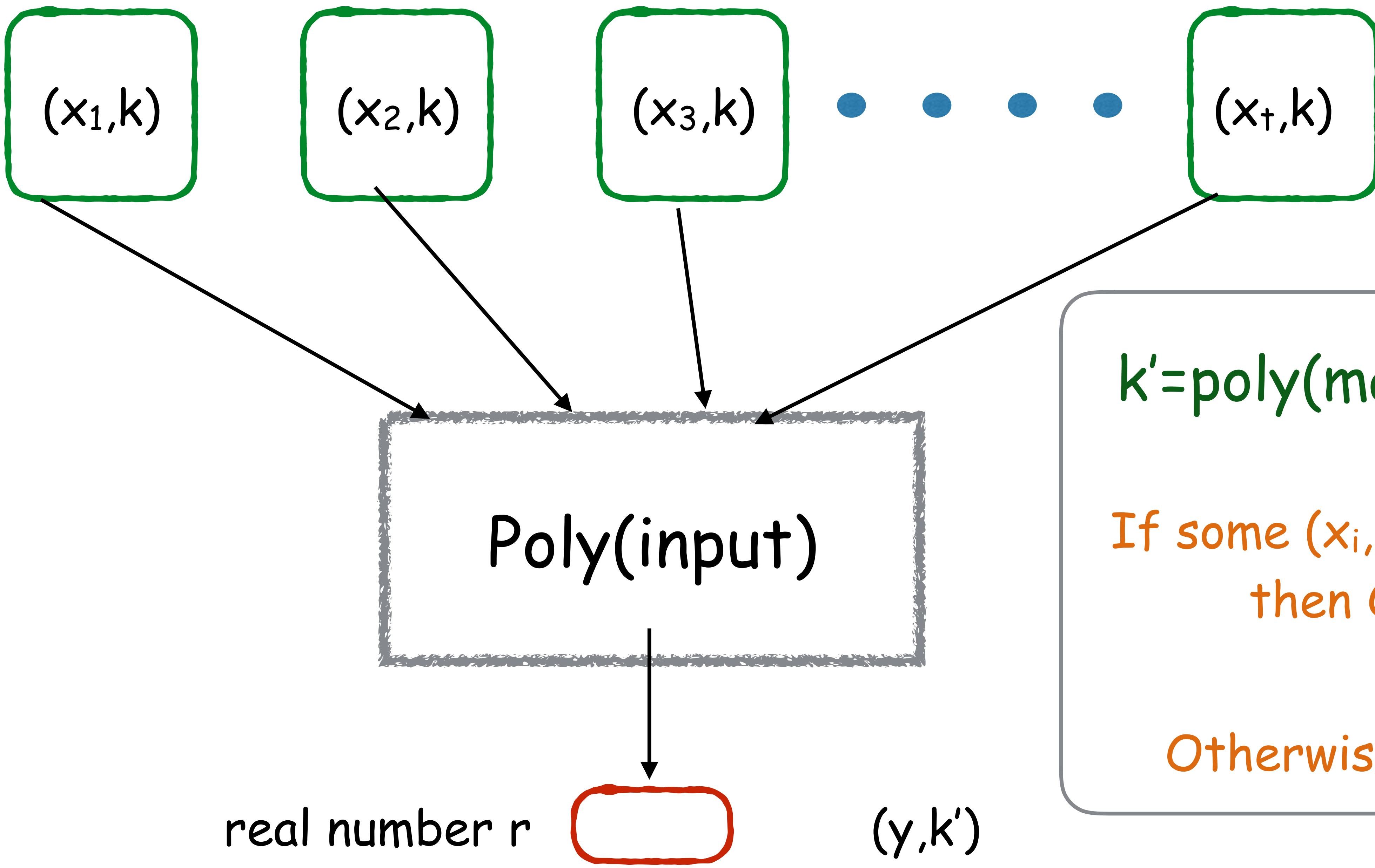
$k' = poly(\max(|x_i|) + \log t)$

If some (x_i,k) is a yes-instance then OPT(y,k') is LARGE

Otherwise OPT(y,k') is SMALL

Lower Bounds

``α'' **Gap-creating Cross Compositions**



MAXIMIZATION PROBLEM

$k' = poly(\max(|x_i|) + \log t)$

If some (x_i, k) is a yes-instance then $OPT(y,k') >= r$

Otherwise $OPT(y,k') < r/\alpha$

Lower Bounds

````$\alpha$'''' Gap-creating Cross Compositions

Problem P

$\alpha$-Gap-creating Cross Composition +
$\alpha$-approximate polynomial
compression


````

Lower Bounds

α-gap Long Path cross composes into Longest Path

↓

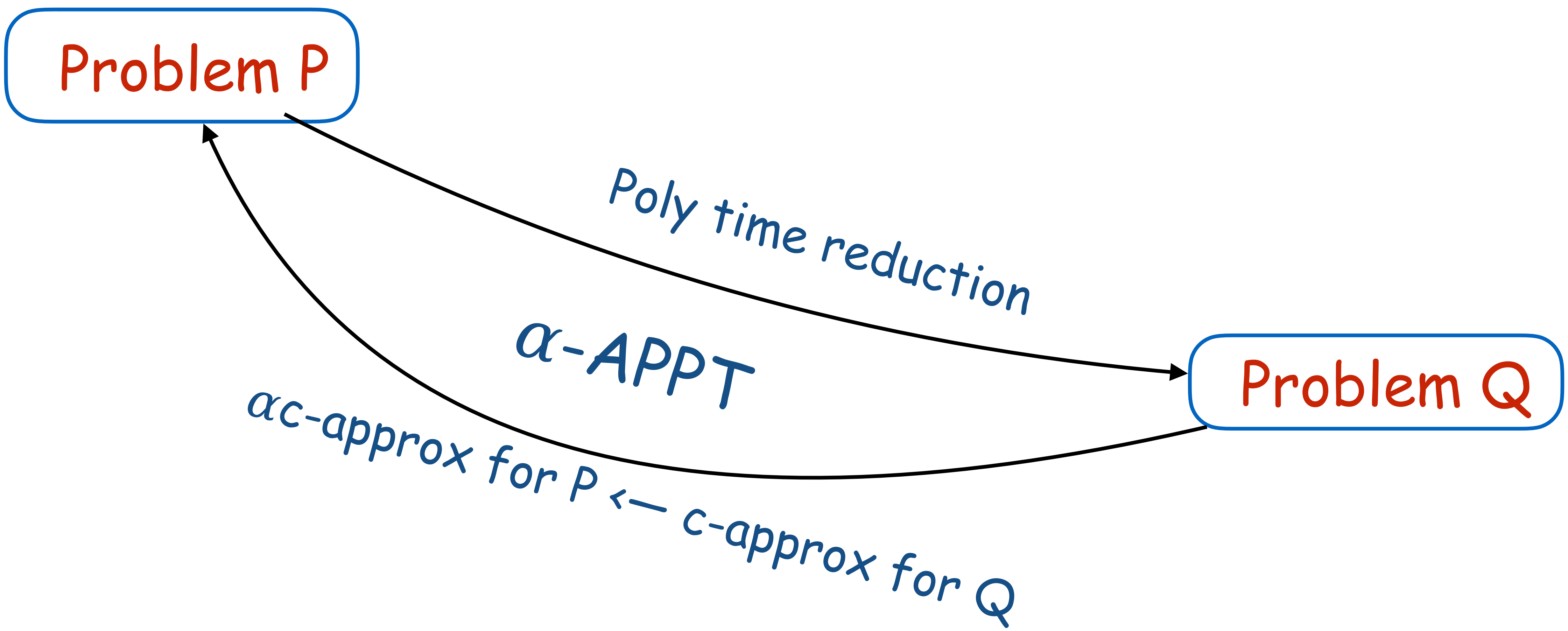
Longest Path has no α-approximate polynomial kernel unless NP in coNP/poly.

More Lower Bounds

Set Cover (universe size) has no α-approximate polynomial kernel unless NP in coNP/poly.

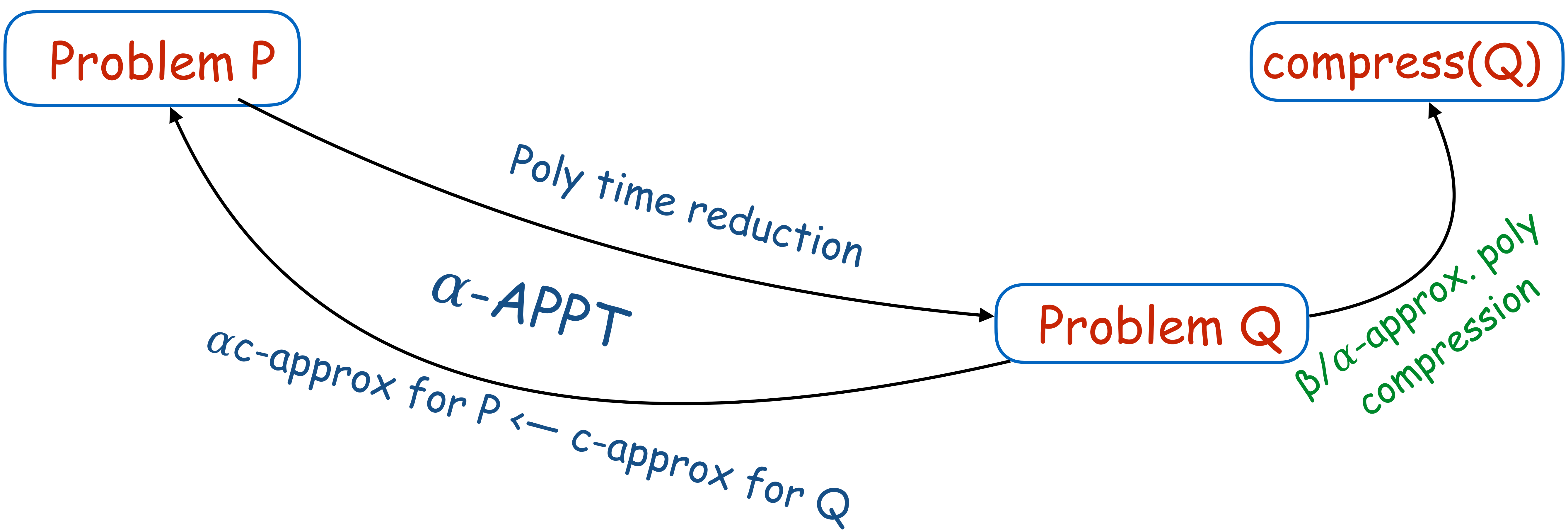
α-approximate Poly. Param. Transformations

Reductions to rule out approximate polynomial kernels



α-approximate Poly. Param. Transformations

Reductions to rule out approximate polynomial kernels



α-approximate Poly. Param. Transformations

Reductions to rule out approximate polynomial kernels

no β-approximate polynomial compression for P + α-APPT from P to Q —>
no β/α-approx. poly compression for Q

Other questions to attack using lossy kernels

Other questions to attack using lossy kernels

Other questions to attack using lossy kernels

No reason why the study of approximate kernels should be restricted to problems without polynomial kernels

approx factor vs kernel size

Dominating Set on d-degenerate graphs

a vertex set S such that every vertex in V(G)\S is adjacent to a vertex in S.

a graph where every subgraph has a vertex of degree at most d

There is a d^2-approximation [Jones et al.]

There is a kernel of size $k^{O(d^2)}$ [Philip et al.]

Cannot have a kernel of size $k^{o(d^2)}$ [Cygan et al.]

approx factor vs kernel size

Dominating Set on d-degenerate graphs

a vertex set S such that every vertex in V(G)\S is adjacent to a vertex in S.

a graph where every subgraph has a vertex of degree at most d

There is a d^2-approximate kernel of size O(1)

There is a 1-approximate kernel of size $k^{O(d^2)}$

approx factor vs kernel size

Dominating Set on d-degenerate graphs

a vertex set S such that every vertex in V(G)\S is adjacent to a vertex in S.

a graph where every subgraph has a vertex of degree at most d

There is a d^2-approximate kernel of size O(1)

There is a 1-approximate kernel of size $k^{O(d^2)}$

Is there a curve interpolating between these two extremes?

approx factor vs kernel size

Dominating Set on d-degenerate graphs

a vertex set S such that every vertex in V(G)\S is adjacent to a vertex in S.

a graph where every subgraph has a vertex of degree at most d

[Eiben, Hermelin, R. 17]

For every ρ in {1,..,d} there is a
(d/ρ)-approximate kernel of size $k^{O(\rho d)}$

approx factor vs kernel size

Open Problem:

What about d-hitting set?

There is a d-approximate kernel of size O(1)

There is a 1-approximate kernel of size $k^{O(d)}$

Is there a curve interpolating between these two extremes?

Uniform vs non-uniform kernels

Treewidth-t deletion problem has a kernel of size $k^{f(t)}$

[Fomin, Misra, Lokshtanov, Saurabh, 12]

Treewidth-t deletion problem does not have a kernel of size $f(t)\ k^{O(1)}$ unless NP in coNP/poly.

[Giannopolou, Jansen, Lokshtanov, Saurabh, 15]

Treewidth-t deletion problem has a $(1+\epsilon)$-approximate kernel of size $f(t)\ k^3$

[Koutecký, Lokshtanov, Misra, Saurabh, Sharma, Zehavi 17]

Take home message

- For many problems, allowing a small loss in accuracy, gives a dramatic improvement in kernel size.
- Interesting questions even for problems with poly kernels!
- Does not work for all problems! There is a lower bound machinery.
- If you like kernelization and/or approximation, you'll probably like their  as well!

Open Problems

- Many many many open problems.

- Full version of main paper on arxiv has a list of problems.

What about Directed Feedback Vertex Set?

Thank you for your attention!

Thank you for your attention!

Thank you for your attention!

Thank you for your attention!