FPT is Characterized by Useful Obstruction Sets

Bart M. P. Jansen

Joint work with
Michael R. Fellows, Charles Darwin Univ.

Universiteit Utrecht

June 21st 2013, WG 2013, Lübeck

A NEW CHARACTERIZATION OF FPT

Well-Quasi-Orders

- A **quasi-order** is a transitive, reflexive, binary relation $\preceq$ on a (usually infinite) set S.
  - If $x \preceq y$, then x **precedes** y.

Universiteit Utrecht

Well-Quasi-Orders

- A **quasi-order** is a transitive, reflexive, binary relation $\preceq$ on a (usually infinite) set S.
  - If $x \preceq y$, then x **precedes** y.

- A quasi-order $\preceq$ is a **well-quasi-order** on S if
  - for every infinite sequence $x_1, x_2, \ldots$ over S,
  - there are indices $i<j$ such that $x_i \preceq x_j$.

Well-Quasi-Orders

- A **quasi-order** is a transitive, reflexive, binary relation $\preceq$ on a (usually infinite) set S.
  - If $x \preceq y$, then x **precedes** y.

- A quasi-order $\preceq$ is a **well-quasi-order** on S if
  - for every infinite sequence $x_1, x_2, \ldots$ over S,
  - there are indices $i<j$ such that $x_i \preceq x_j$.

- Set $L \subseteq S$ is a **lower ideal** of S under $\preceq$ if
  - $\forall x,y \in S$: if $x \in L$ and $y \preceq x$, then $y \in L$.

Well-Quasi-Orders

- A **quasi-order** is a transitive, reflexive, binary relation $\preceq$ on a (usually infinite) set S.
  - If $x \preceq y$, then x **precedes** y.

- A quasi-order $\preceq$ is a **well-quasi-order** on S if
  - for every infinite sequence $x_1, x_2, \ldots$ over S,
  - there are indices $i<j$ such that $x_i \preceq x_j$.

- Set $L \subseteq S$ is a **lower ideal** of S under $\preceq$ if
  - $\forall x,y \in S$: if $x \in L$ and $y \preceq x$, then $y \in L$.

- A quasi-order is **polynomial** if $x \preceq y$ can be tested in poly(|x|+|y|) time.

The Obstruction Principle

If $\preceq$ is a WQO on S, and L ⊆ S is a lower ideal, then there is a **finite obstruction set** OBS(L) ⊆ S, such that for all x ∈ S:

x ∈ L iff no element in OBS(L) precedes x.

The Obstruction Principle

If $\preceq$ is a WQO on S, and L ⊆ S is a lower ideal, then there is a **finite obstruction set** Obs(L) ⊆ S, such that for all x ∈ S:

x ∈ L iff no element in Obs(L) precedes x.

- Decide membership in a lower ideal by testing containment of an obstruction.

The Obstruction Principle

If $\preceq$ is a WQO on S, and L ⊆ S is a lower ideal, then there is a **finite obstruction set** OBS(L) ⊆ S, such that for all x ∈ S:

x ∈ L iff no element in OBS(L) precedes x.

- Decide membership in a lower ideal by testing containment of an obstruction.
- Any element y ∈ S \ L is an **obstruction.**

The Obstruction Principle

If $\preceq$ is a WQO on S, and L ⊆ S is a lower ideal, then there is a **finite obstruction set** OBS(L) ⊆ S, such that for all x ∈ S:

x ∈ L iff no element in OBS(L) precedes x.

- Decide membership in a lower ideal by testing containment of an obstruction.
- Any element y ∈ S \ L is an **obstruction.**
- An obstruction is **minimal** if all elements strictly preceding it belong to L.

Algorithmic Applications of WQO's

- Fellows & Langston, JACM 1988:
  - *k*-PATH,
  - *k*-VERTEX COVER,
  - *k*-FEEDBACK VERTEX SET,

  can be solved in $\mathcal{O}(n^3)$ time, for each fixed k.

Algorithmic Applications of WQO's

- Fellows & Langston, JACM 1988:
  - *k*-PATH,
  - *k*-VERTEX COVER,
  - *k*-FEEDBACK VERTEX SET,

  can be solved in $\mathcal{O}(n^3)$ time, for each fixed k.

Obstruction principle	Lower ideals	Efficient order testing
• Graphs are well-quasi-ordered by the minor relation.	• YES or NO instances are closed under taking minors.	• $f(H)n^3$ time for each fixed graph H.

Algorithmic Applications of WQO's

- Fellows & Langston, JACM 1988:
  - *k*-PATH,
  - *k*-VERTEX COVER,
  - *k*-FEEDBACK VERTEX SET,

  can be solved in $O(n^3)$ time, for each fixed k.

Obstruction principle
- Graphs are well-quasi-ordered by the minor relation.

Lower ideals
- YES or NO instances are closed under taking minors.

Efficient order testing
- $f(H)n^3$ time for each fixed graph H.

- Results led to the development of parameterized complexity.

The class FPT

The class FPT

- A **parameterized problem** is a set $Q \subseteq \Sigma^* \times \mathbb{N}$
  - Each instance $(x,k) \in \Sigma^* \times \mathbb{N}$ has a *parameter k*.
  - The **size** of an instance (x,k) is $|x| + k$.

The class FPT

- A **parameterized problem** is a set $\mathbb{Q} \subseteq \Sigma^* \times \mathbb{N}$
  - Each instance $(x,k) \in \Sigma^* \times \mathbb{N}$ has a *parameter k*.
  - The **size** of an instance (x,k) is |x| + k.

Strongly Uniform FPT (Fixed-Parameter Tractable)
A parameterized problem $\mathbb{Q}$ is strongly uniform FPT if there is an algorithm that decides whether $(x,k) \in \mathbb{Q}$ in $f(k)|x|^c$ time.
(for a computable function f and constant c)

The class FPT

- A **parameterized problem** is a set $\mathbb{Q} \subseteq \Sigma^* \times \mathbb{N}$
  - Each instance $(x,k) \in \Sigma^* \times \mathbb{N}$ has a *parameter k*.
  - The **size** of an instance (x,k) is |x| + k.

Strongly Uniform FPT (Fixed-Parameter Tractable)
A parameterized problem $\mathbb{Q}$ is strongly uniform FPT if there is an algorithm that decides whether $(x,k) \in \mathbb{Q}$ in $f(k)|x|^c$ time.
(for a computable function f and constant c)

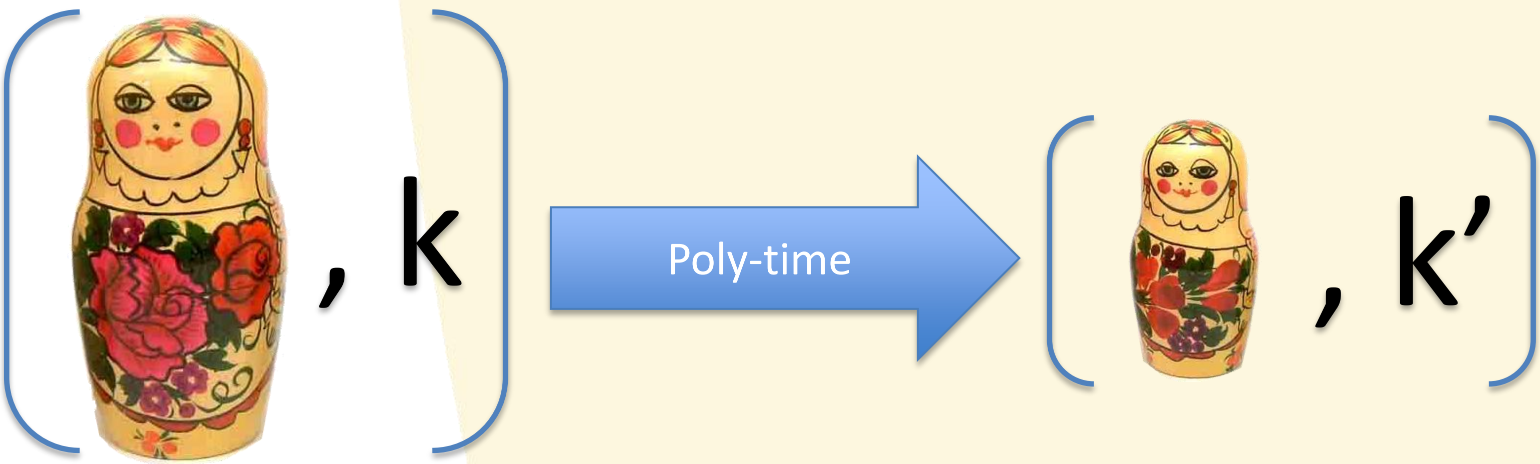
- There are weaker notions. (non-uniform, non-computable f)

Kernelization

- A **kernel of size f(k)** for a parameterized problem Q is a polynomial-time algorithm that transforms (x,k) into (x',k'),

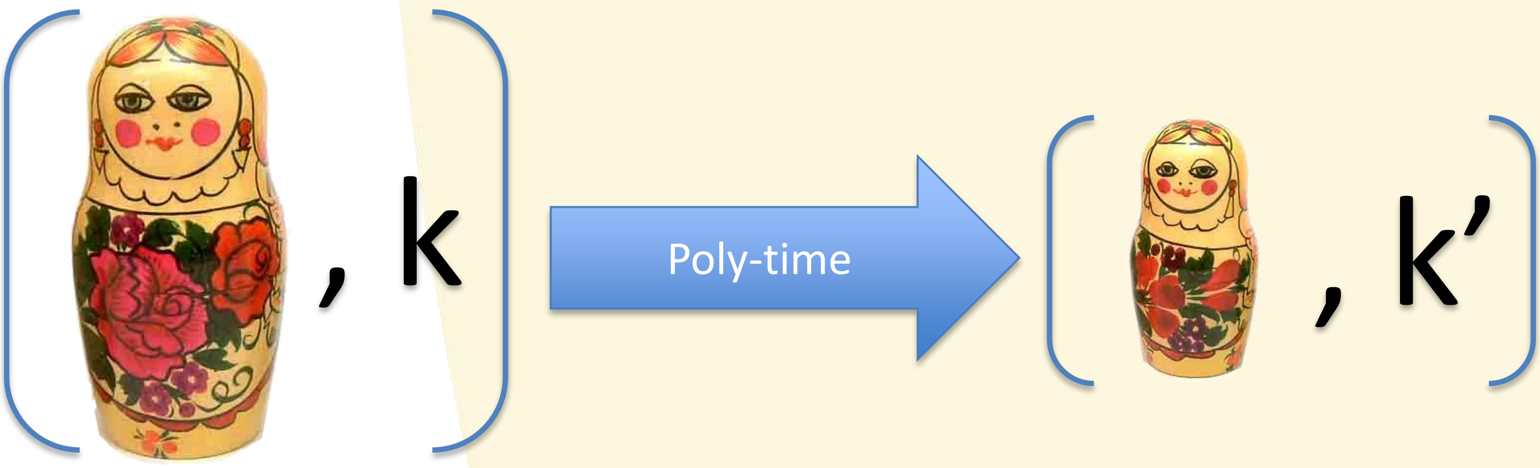
Kernelization

- A **kernel of size f(k)** for a parameterized problem Q is a polynomial-time algorithm that transforms (x,k) into (x',k'),



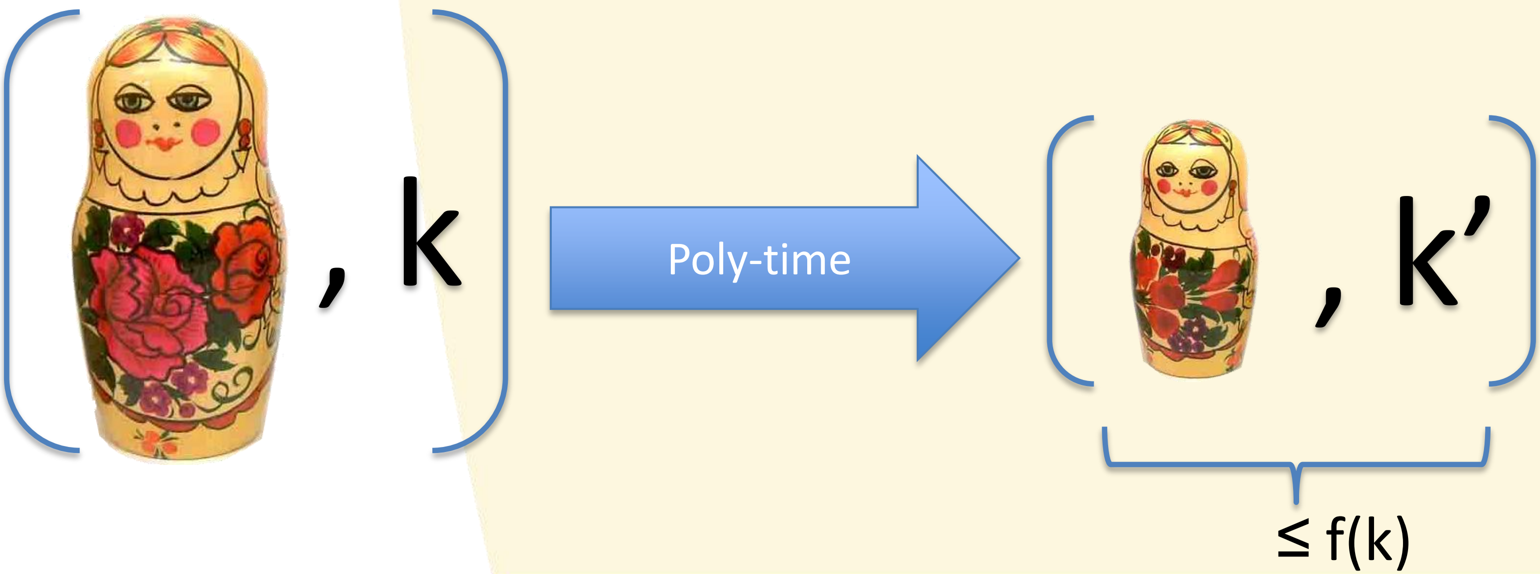
Kernelization

- A **kernel of size f(k)** for a parameterized problem $\mathbb{Q}$ is a polynomial-time algorithm that transforms (x,k) into (x’,k’),
  - such that (x,k) in $\mathbb{Q}$ iff (x’,k’) in $\mathbb{Q}$,



Kernelization

- A **kernel of size f(k)** for a parameterized problem $\mathbb{Q}$ is a polynomial-time algorithm that transforms (x,k) into (x’,k’),
  - such that (x,k) in $\mathbb{Q}$ iff (x’,k’) in $\mathbb{Q}$,
  - and |x’|+k’ is bounded by f(k).



New characterization of FPT

- For any parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:

New characterization of FPT

- For any parameterized problem $\mathbb{Q} \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:

  1. Problem $\mathbb{Q}$ is contained in strongly uniform **FPT**.

New characterization of FPT

- For any parameterized problem $\mathbb{Q} \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:

1. Problem $\mathbb{Q}$ is contained in strongly uniform **FPT**.
2. Problem $\mathbb{Q}$ is decidable and admits a **kernel** of computable size.

New characterization of FPT

- For any parameterized problem $\mathbb{Q} \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:

1. Problem $\mathbb{Q}$ is contained in strongly uniform **FPT**.
2. Problem $\mathbb{Q}$ is decidable and admits a **kernel** of computable size.
3. Problem $\mathbb{Q}$ is decidable and there is a polynomial-time **quasi-order** $\preceq$ on $\Sigma^* \times \mathbb{N}$ and a computable function f: $\mathbb{N} \rightarrow \mathbb{N}$ such that:
   - The set $\mathbb{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
   - For every $(x,k) \notin \mathbb{Q}$, there is an **obstruction** $(x',k') \notin \mathbb{Q}$ of size at most f(k) with $(x',k') \preceq (x,k)$.

Universiteit Utrecht

New characterization of FPT

- For any parameterized problem $\mathbb{Q} \subseteq \Sigma^* \times \mathbb{N}$, the following statements are equivalent:

1. Problem $\mathbb{Q}$ is contained in strongly uniform **FPT**.
2. Problem $\mathbb{Q}$ is decidable and admits a **kernel** of computable size.
3. Problem $\mathbb{Q}$ is decidable and there is a polynomial-time **quasi-order** $\preceq$ on $\Sigma^* \times \mathbb{N}$ and a computable function f: $\mathbb{N} \rightarrow \mathbb{N}$ such that:
   - The set $\mathbb{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
   - For every (x,k) $\notin \mathbb{Q}$, there is an **obstruction** (x',k') $\notin \mathbb{Q}$ of size at most f(k) with (x',k') $\preceq$ (x,k).

New characterization of FPT

3. Problem $\mathbb{Q}$ is decidable and there is a polynomial-time **quasi-order** $\preceq$ on $\Sigma^* \times \mathbb{N}$ and a computable function f: $\mathbb{N} \rightarrow \mathbb{N}$ such that:
   - The set $\mathbb{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
   - For every (x,k) $\notin \mathbb{Q}$, there is an **obstruction** (x',k') $\notin \mathbb{Q}$ of size at most f(k) with (x',k') $\preceq$ (x,k).

New characterization of FPT

- Implies that for every k, there is a **finite** obstruction set OBS(k) containing instances of size ≤ f(k):
  - (x,k) in $\mathbb{Q}$ iff no element of OBS(k) precedes it.

3. Problem $\mathbb{Q}$ is decidable and there is a polynomial-time **quasi-order** $\preceq$ on $\Sigma^* \times \mathbb{N}$ and a computable function f: $\mathbb{N} \rightarrow \mathbb{N}$ such that:
   - The set $\mathbb{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
   - For every (x,k) $\notin \mathbb{Q}$, there is an **obstruction** (x',k') $\notin \mathbb{Q}$ of size at most f(k) with (x',k') $\preceq$ (x,k).

New characterization of FPT

- Implies that for every k, there is a **finite** obstruction set OBS(k) containing instances of size ≤ f(k):
  - (x,k) in $\mathbb{Q}$ iff no element of OBS(k) precedes it.
- The obstruction-testing method that lies at the origins of FPT is not *just one way* of obtaining FPT algorithms:
  - **all of FPT** can be obtained this way.

3. Problem $\mathbb{Q}$ is decidable and there is a polynomial-time **quasi-order** $\preceq$ on $\Sigma^* \times \mathbb{N}$ and a computable function f: $\mathbb{N} \rightarrow \mathbb{N}$ such that:
   - The set $\mathbb{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
   - For every (x,k) $\notin \mathbb{Q}$, there is an **obstruction** (x',k') $\notin \mathbb{Q}$ of size at most f(k) with (x',k') $\preceq$ (x,k).

KERNEL SIZE VS. OBSTRUCTION SIZE

Small kernels yield small obstructions

- Problem $\mathcal{Q}$ is decidable and admits a kernel of size $\mathcal{O}(\mathbf{f(k)})$

Small kernels yield small obstructions

- Problem $\mathcal{Q}$ is decidable and admits a kernel of size $\mathcal{O}(\mathbf{f(k)})$

implies

Small kernels yield small obstructions

- Problem $\mathbb{Q}$ is decidable and admits a kernel of size $\mathcal{O}(\mathbf{f(k)})$

implies

- Problem $\mathbb{Q}$ is decidable and there is a polynomial-time quasi-order $\preceq$ on $\Sigma^* \times \mathbb{N}$ such that:
  - The set $\mathbb{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
  - For every $(x,k) \notin \mathbb{Q}$, there is an obstruction $(x',k') \notin \mathbb{Q}$ of size $\mathcal{O}(\mathbf{f(k)})$ with $(x',k') \preceq (x,k)$.

Small kernels yield small obstructions

- Problem $\mathbb{Q}$ is decidable and admits a kernel of size $\mathcal{O}$**(f(k))**

implies

- Problem $\mathbb{Q}$ is decidable and there is a polynomial-time quasi-order $\preceq$ on $\Sigma^* \times \mathbb{N}$ such that:
  - The set $\mathbb{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
  - For every $(x,k) \notin \mathbb{Q}$, there is an obstruction $(x',k') \notin \mathbb{Q}$ of size $\mathcal{O}$**(f(k))** with $(x',k') \preceq (x,k)$.

Parameterized problems with polynomial kernels are characterized by obstructions of polynomial size.

Small kernels yield small obstructions

- Problem $\mathbb{Q}$ is decidable and admits a kernel of size **$\mathcal{O}(f(k))$**

implies

- Problem $\mathbb{Q}$ is decidable and there is a polynomial-time quasi-order $\preceq$ on $\Sigma^* \times \mathbb{N}$ such that:
  - The set $\mathbb{Q}$ is a lower ideal of $\Sigma^* \times \mathbb{N}$ under $\preceq$.
  - For every (x,
    (x',k') $\notin \mathbb{Q}$ of

Reverse is false, assuming NP $\not\subset$ coNP/poly.
(Kratsch & Walhström, 2011)

Parameterized problems with polynomial kernels are characterized by obstructions of polynomial size.

Obstruction size vs. kernel size

Polynomial bounds

Obstruction size vs. kernel size

Polynomial bounds

k-Vertex Cover

- Best known kernel has 2k – o(k) vertices [Lampis'11]
- Largest graph that is minor-minimal with vertex cover size k has 2k vertices
  - Vertex Cover obstructions have been studied since 1964 [α-critical graphs: Erdős, Hajnal & Moon]

Obstruction size vs. kernel size

Polynomial bounds

k-VERTEX COVER

- Best known kernel has 2k – o(k) vertices [Lampis'11]
- Largest graph that is minor-minimal with vertex cover size k has 2k vertices
  - Vertex Cover obstructions have been studied since 1964 [α-critical graphs: Erdős, Hajnal & Moon]

k-$\mathcal{F}$-MINOR-FREE DELETION (when $\mathcal{F}$ contains a planar graph)

- Polynomial kernel [Fomin et al.'12]
- Minor-minimal obstructions have polynomial size

Obstruction size vs. kernel size

Polynomial bounds

k-VERTEX COVER

- Best known kernel has 2k – o(k) vertices [Lampis'11]
- Largest graph that is minor-minimal with vertex cover size k has 2k vertices
  - Vertex Cover obstructions have been studied since 1964 [α-critical graphs: Erdős, Hajnal & Moon]

k-$\mathcal{F}$-MINOR-FREE DELETION (when $\mathcal{F}$ contains a planar graph)

- Polynomial kernel [Fomin et al.'12]
- Minor-minimal obstructions have polynomial size

TREEWIDTH parameterized by Vertex Cover

- $\mathcal{O}(vc^3)$-vertex kernel [Bodlaender et al.'11]
- Minor-minimal obstructions have $|V| \leq \mathcal{O}(vc^3)$.

Obstruction size vs. kernel size

Polynomial bounds

k-Vertex Cover

- Best known kernel has 2k – o(k) vertices [Lampis'11]
- Largest graph that is minor-minimal with vertex cover size k has 2k vertices
  - Vertex Cover obstructions have been studied since 1964 [α-critical graphs: Erdős, Hajnal & Moon]

k-$\mathcal{F}$-Minor-Free Deletion (when $\mathcal{F}$ contains a planar graph)

- Polynomial kernel [Fomin et al.'12]
- Minor-minimal obstructions have polynomial size

Treewidth parameterized by Vertex Cover

- $\mathcal{O}(\text{vc}^3)$-vertex kernel [Bodlaender et al.'11]
- Minor-minimal obstructions have $|V| \leq \mathcal{O}(\text{vc}^3)$.

q-Coloring parameterized by Vertex Cover

- $\mathcal{O}(\text{vc}^q)$-vertex kernel [J+Kratsch'11]
- Vertex-minimal no-instances have $\text{vc}^{\Theta(q)}$ vertices.

Obstruction size vs. kernel size

Superpolynomial bounds

Obstruction size vs. kernel size

Superpolynomial bounds

3-COLORING parameterized by Feedback Vertex Set

- No polynomial kernel unless NP ⊆ coNP/poly. [J+Kratsch'11]
- Size of vertex-minimal NO-instances is unbounded in FVS number.

Obstruction size vs. kernel size

Superpolynomial bounds

3-Coloring parameterized by Feedback Vertex Set

- No polynomial kernel unless NP ⊆ coNP/poly. [J+Kratsch'11]
- Size of vertex-minimal NO-instances is unbounded in FVS number.

***k*-Ramsey**

- No polynomial kernel unless NP ⊆ coNP/poly. [Kratsch'12]
- Lower bound construction is based on a Turán-like host graph whose size is superpolynomial in its parameter.

Obstruction size vs. kernel size

Superpolynomial bounds

3-COLORING parameterized by Feedback Vertex Set

- No polynomial kernel unless NP ⊆ coNP/poly. [J+Kratsch'11]
- Size of vertex-minimal NO-instances is unbounded in FVS number.

***k*-RAMSEY**

- No polynomial kernel unless NP ⊆ coNP/poly. [Kratsch'12]
- Lower bound construction is based on a Turán-like host graph whose size is superpolynomial in its parameter.

***k*-PATHWIDTH**

- No polynomial kernel unless NP ⊆ coNP/poly. [BodlaenderDFH'09]
- Minor-minimal obstructions with $\Omega(3^k)$ vertices.

EXPLOITING OBSTRUCTIONS FOR LOWER-BOUNDS ON KERNEL SIZES

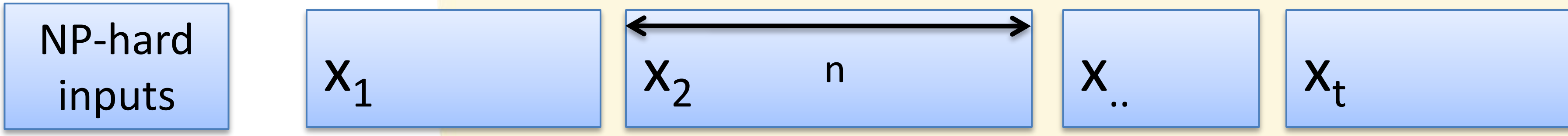
Composition algorithms

Composition algorithms

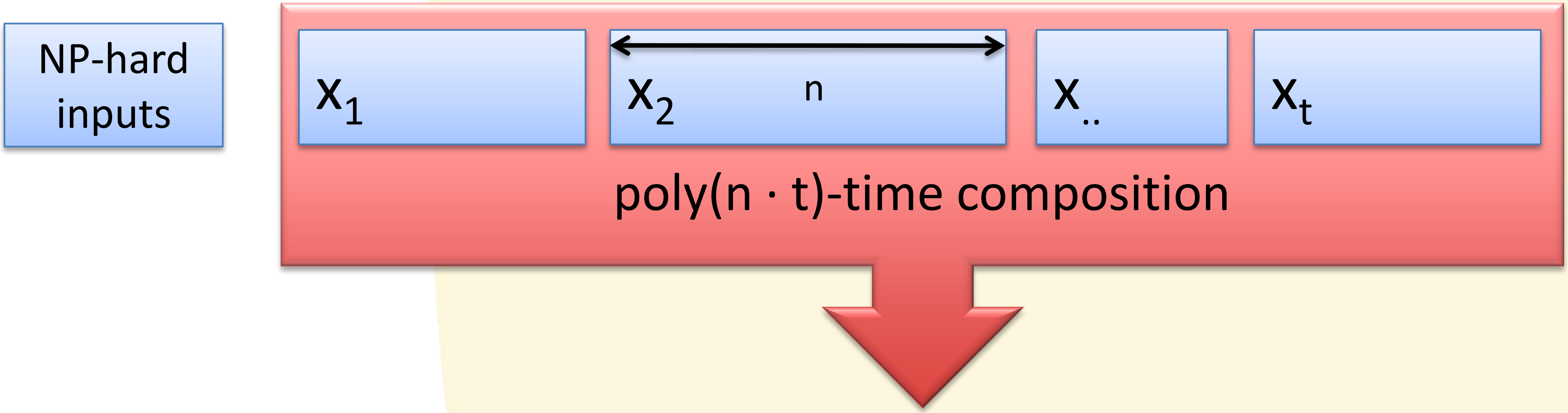
NP-hard inputs

x_1 | x_2 | $x_{..}$ | x_t

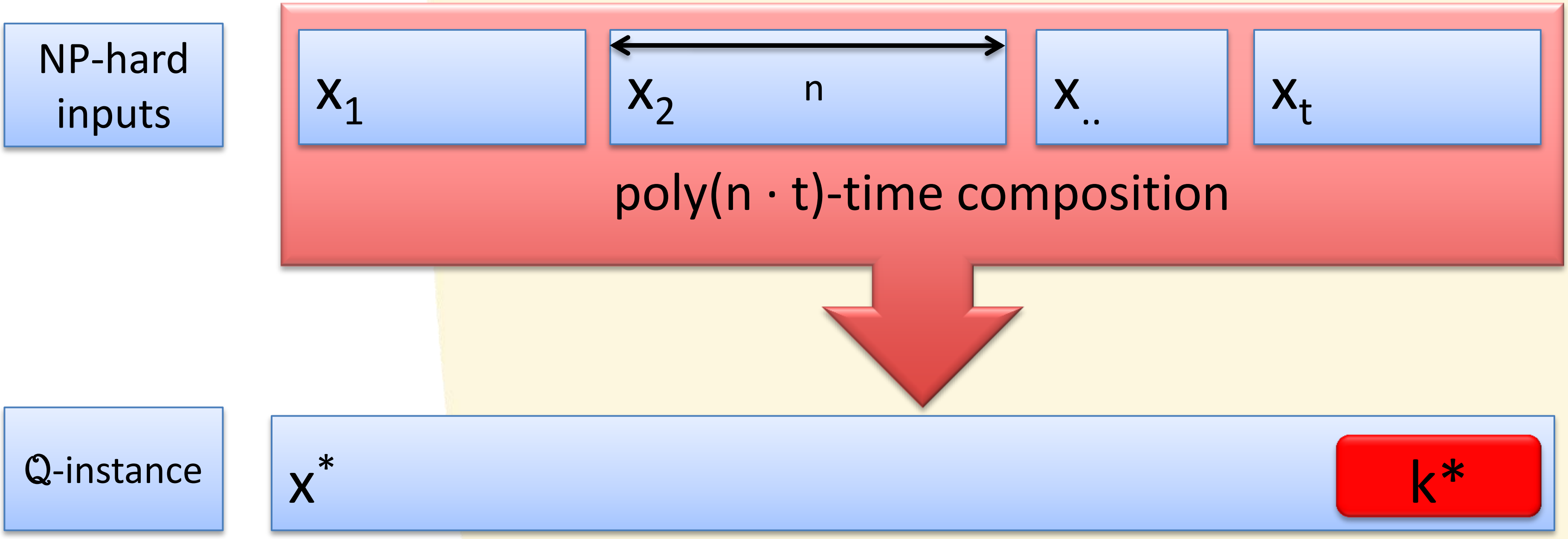
Composition algorithms



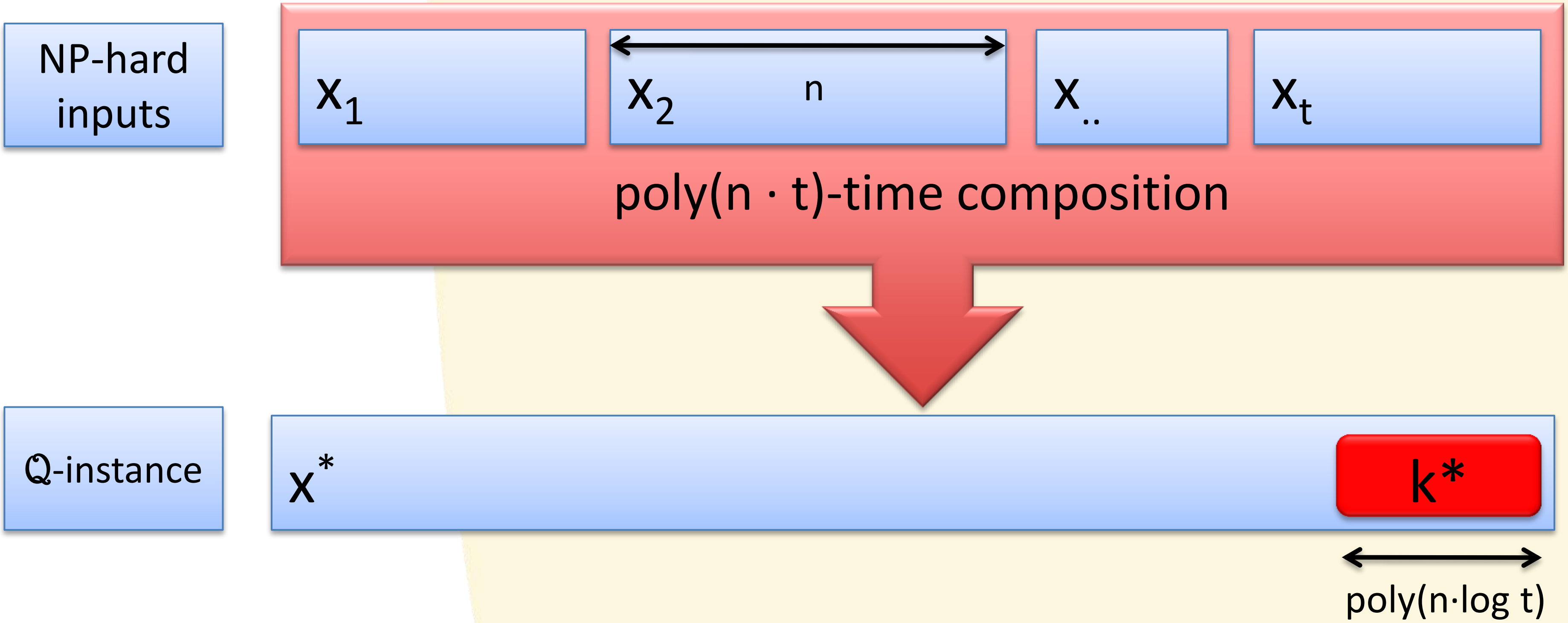
Composition algorithms



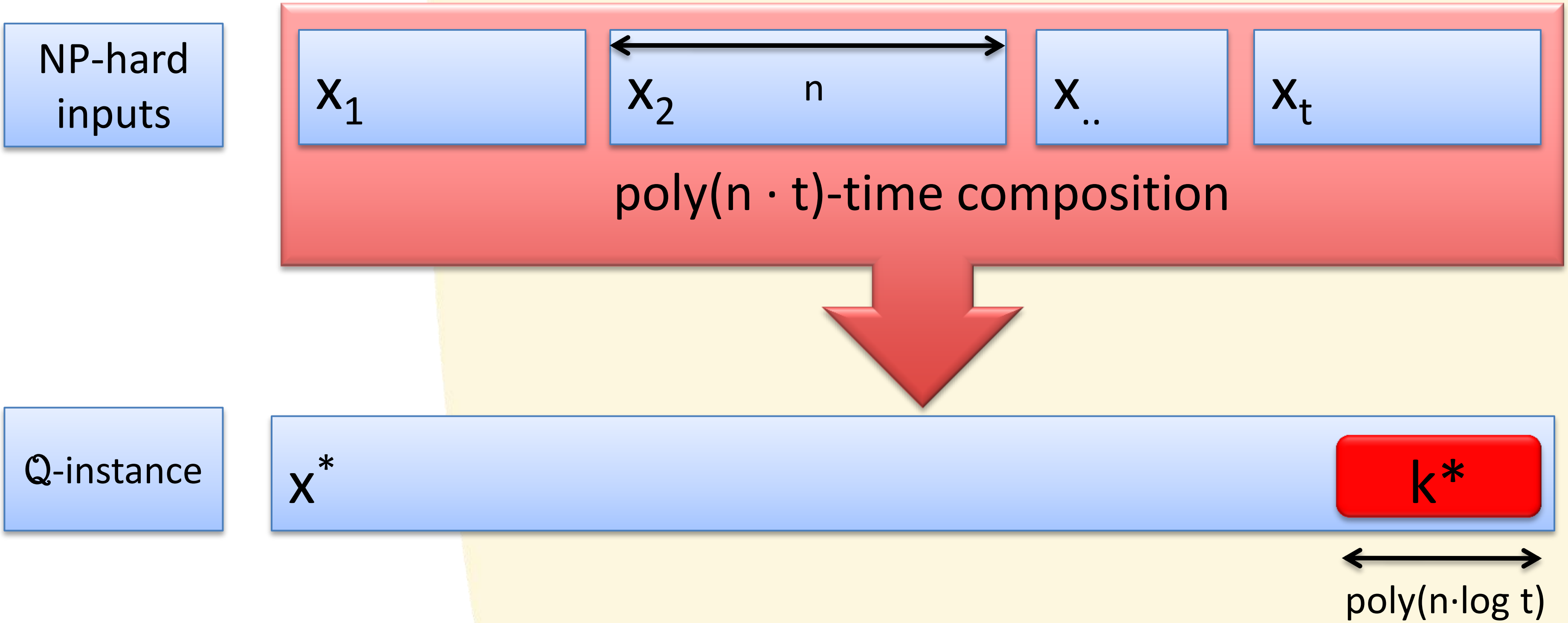
Composition algorithms



Composition algorithms



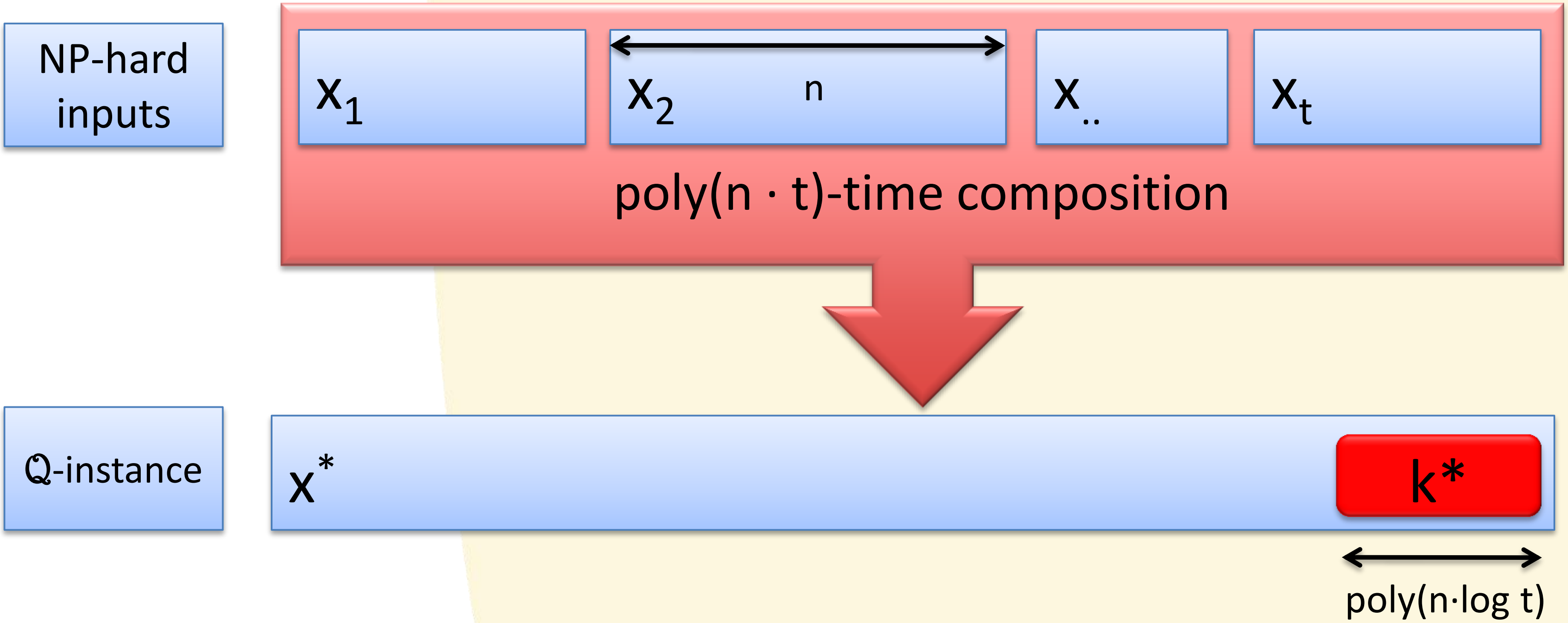
Composition algorithms



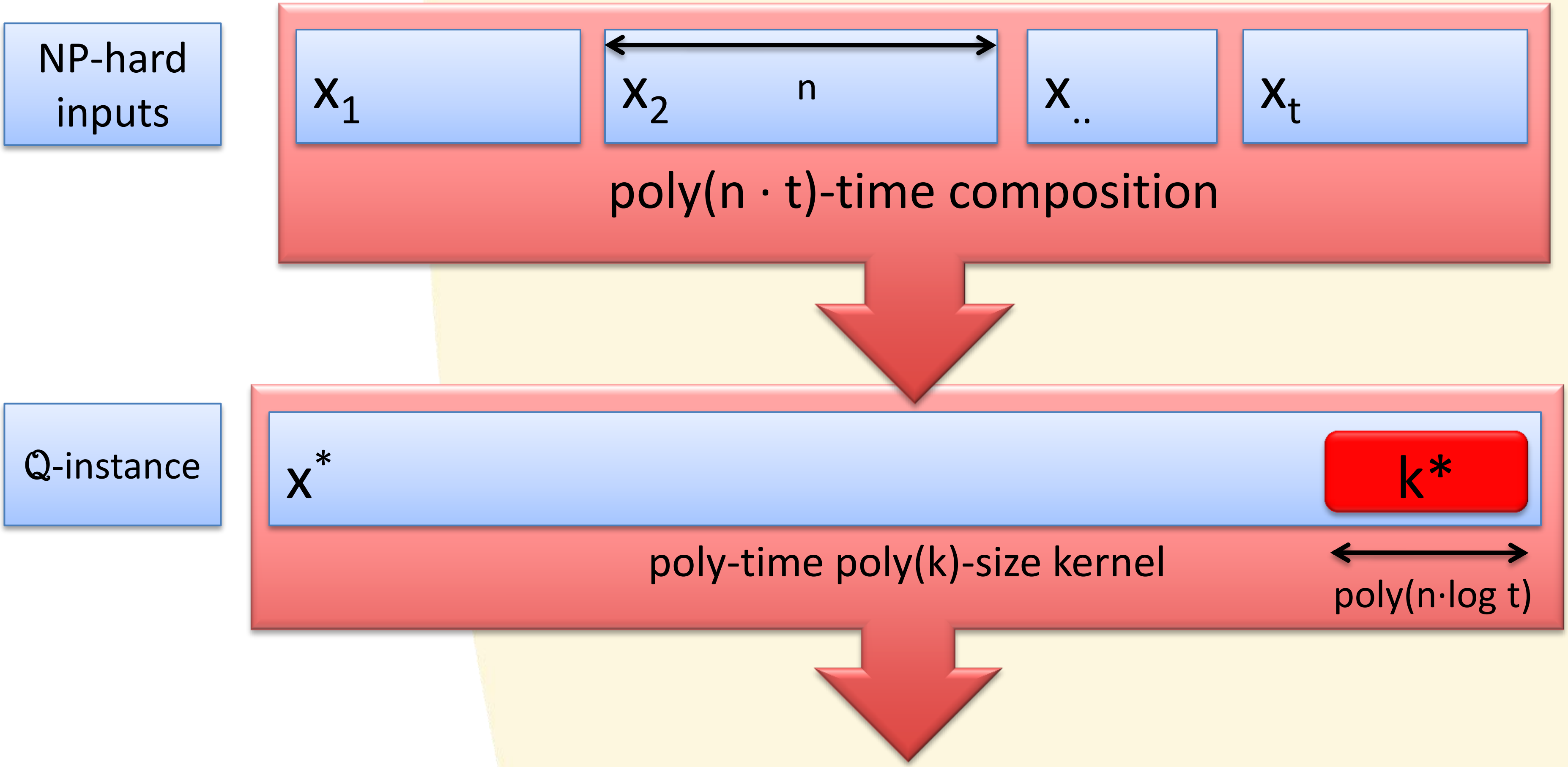
AND-Cross-composition: (x*,k*) ∈ Q iff all inputs are YES

OR-Cross-composition: (x*,k*) ∈ Q iff some input is YES

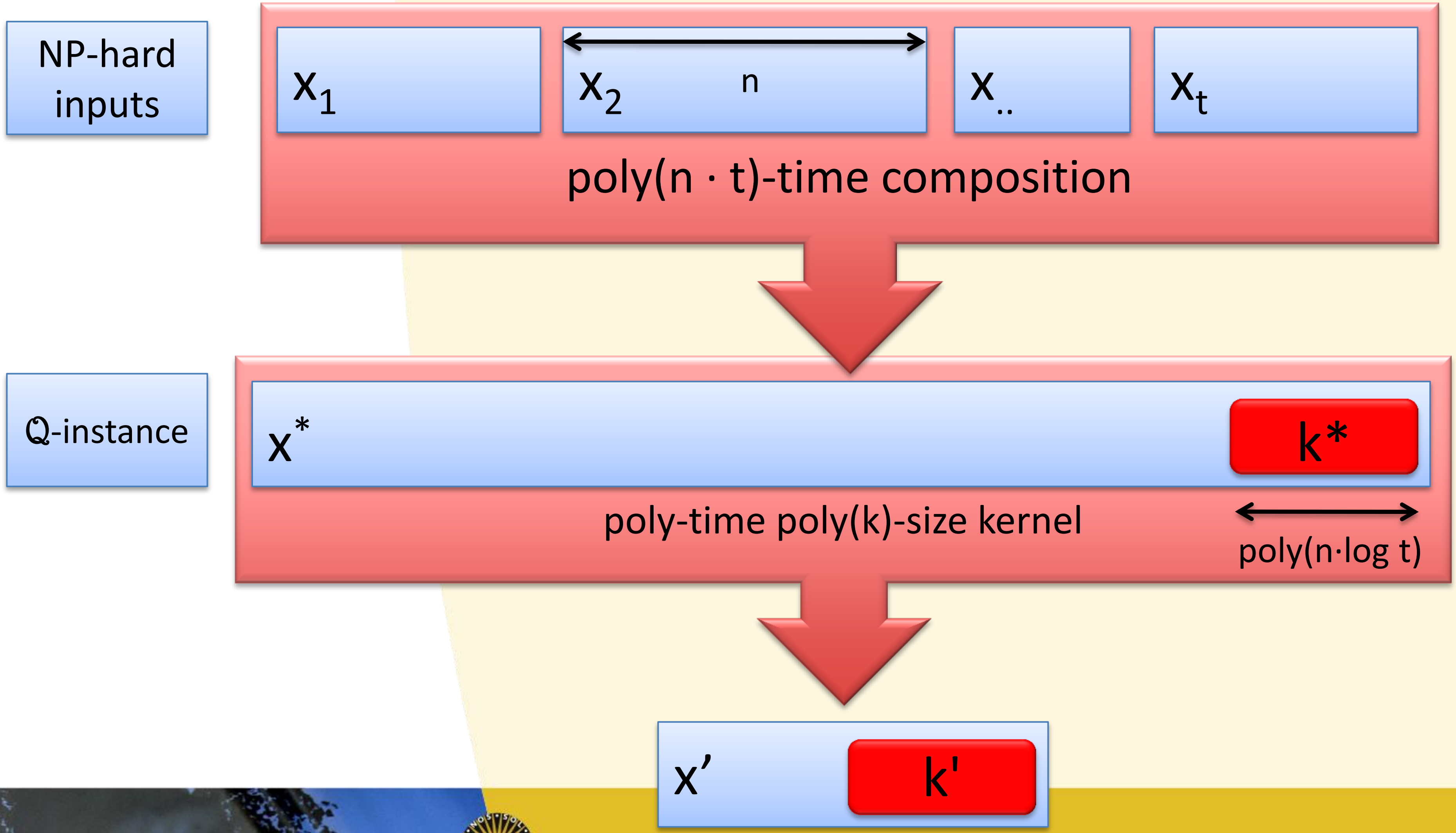
Composition algorithms



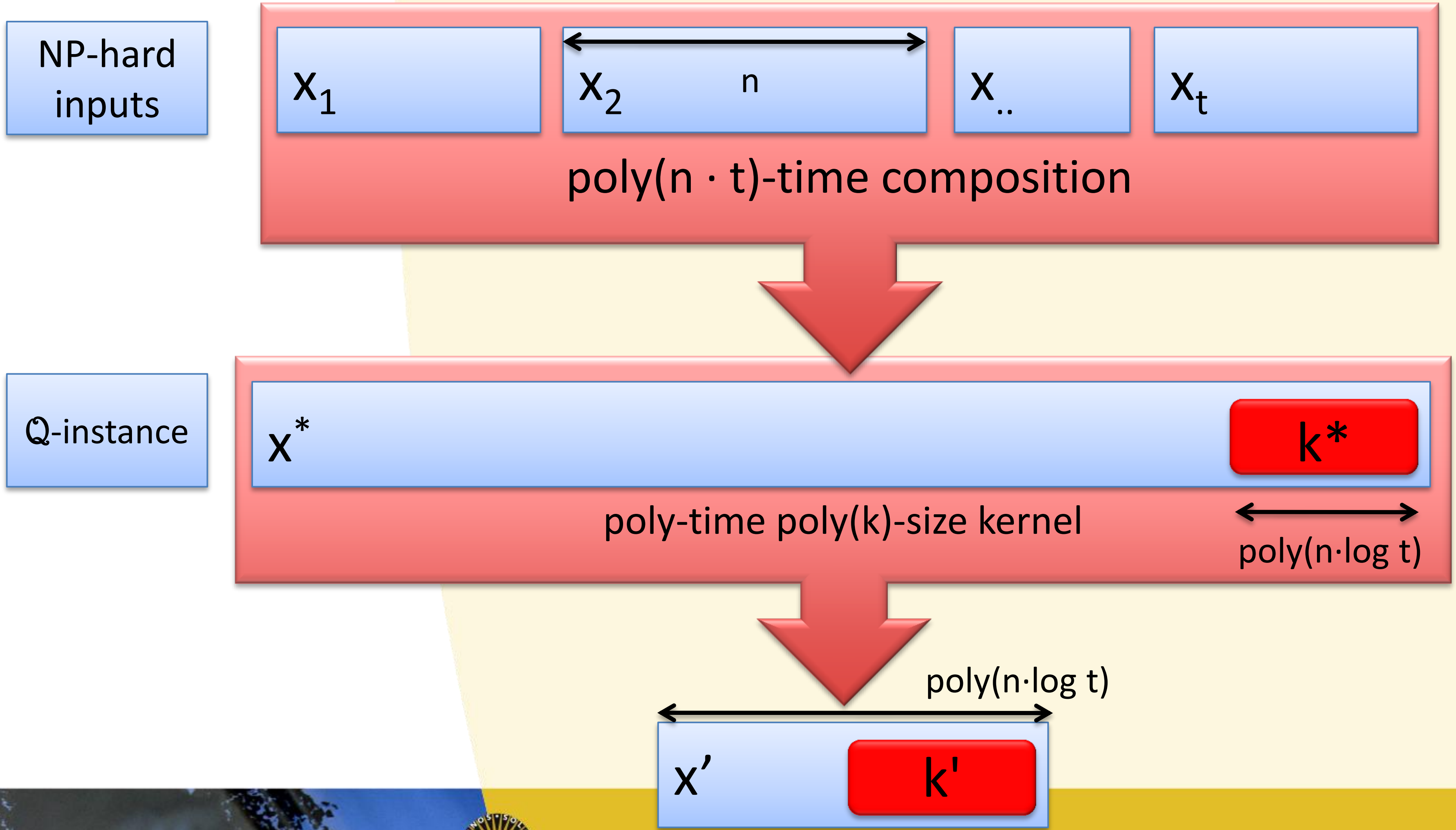
Composition algorithms



Composition algorithms



Composition algorithms



The *k*-Pathwidth problem

- The pathwidth of a graph measures how "path-like" it is
  - Pathwidth does not increase when taking minors

The *k*-Pathwidth problem

- The pathwidth of a graph measures how “path-like” it is
  - Pathwidth does not increase when taking minors

- *k*-PATHWIDTH
  Input: A graph G, an integer k.
  Parameter: k.
  Question: Is the pathwidth of G at most k?

The *k*-Pathwidth problem

- The pathwidth of a graph measures how “path-like” it is
  - Pathwidth does not increase when taking minors
- *k*-PATHWIDTH
  Input: A graph G, an integer k.
  Parameter: k.
  Question: Is the pathwidth of G at most k?
- Disjoint union acts as AND for question of “pathwidth ≤ k?”:
  - $\textsc{pw}(G_1 \uplus G_2 \uplus \ldots \uplus G_t) \leq k \Leftrightarrow \forall i: \textsc{pw}(G_i) \leq k$.

The *k*-Pathwidth problem

- The pathwidth of a graph measures how "path-like" it is
  - Pathwidth does not increase when taking minors
- *k*-PATHWIDTH
  Input: A graph G, an integer k.
  Parameter: k.
  Question: Is the pathwidth of G at most k?
- Disjoint union acts as AND for question of "pathwidth ≤ k?":
  - $\mathrm{PW}(G_1 \uplus G_2 \uplus \ldots \uplus G_t) \leq k \Leftrightarrow \forall i: \mathrm{PW}(G_i) \leq k$.
- Trivial AND-composition for *k*-PATHWIDTH:
  - Take disjoint union of t PATHWIDTH-instances.
    - Ensure same value of k by padding.
  - Output parameter value is k ≤ n.

The *k*-Pathwidth problem

- The pathwidth of a graph measures how “path-like” it is
  - Pathwidth does not increase when taking minors

- *k*-PATHWIDTH
  Input: A graph G, an integer k.
  Parameter: k.
  Question: Is the pathwidth of G at most k?

- Disjoint union acts as AND for question of “pathwidth ≤ k?”:
  - $\text{PW}(G_1 \uplus G_2 \uplus \ldots \uplus G_t) \leq k \Leftrightarrow \forall i: \text{PW}(G_i) \leq k$.

- Trivial AND-composition for *k*-PATHWIDTH:
  - Take disjoint union of t PATHWIDTH-instances.
    - Ensure same value of k by padding.
  - Output parameter value is k ≤ n.

k-PATHWIDTH is AND-compositional and does not admit a polynomial kernel unless NP ⊆ coNP/poly. [BodlaenderDFH’09,Drucker’12]

OR-Cross-composition

OR-Cross-composition

- The pathwidth measure naturally behaves like an **AND**-gate

OR-Cross-composition

- The pathwidth measure naturally behaves like an **AND**-gate
- By exploiting minimal obstructions to PW≤k with $\Omega(3^k)$ vertices, we create an **OR**-Cross-composition of:
  - $t=3^s$ instances of PW-IMPROVEMENT (G_1,k), … , (G_t,k)
  - into one *k*-PATHWIDTH instance (G^*,k^*) with $k^* \leq \mathcal{O}(n \cdot \log t)$,
  - such that PW$(G^*) \leq k^*$ iff **some** input i is YES.

OR-Cross-composition

- The pathwidth measure naturally behaves like an **AND**-gate
- By exploiting minimal obstructions to PW≤k with $\Omega(3^k)$ vertices, we create an **OR**-Cross-composition of:
  - $t=3^s$ instances of PW-IMPROVEMENT (G_1,k), … , (G_t,k)
  - into one *k*-PATHWIDTH instance (G^*,k^*) with $k^* \leq O(n \cdot \log t)$,
  - such that PW$(G^*) \leq k^*$ iff **some** input i is YES.
- PATHWIDTH IMPROVEMENT

  Input: An integer k, and a graph G of pathwidth ≤ k-1.

  Question: Is the pathwidth of G at most k-2?

OR-Cross-composition

- The pathwidth measure naturally behaves like an **AND**-gate
- By exploiting minimal obstructions to Pw≤k with $\Omega(3^k)$ vertices, we create an **OR**-Cross-composition of:
  - $t=3^s$ instances of PW-IMPROVEMENT (G_1,k), … , (G_t,k)
  - into one *k*-PATHWIDTH instance (G^*,k^*) with $k^* \leq O(n \cdot \log t)$,
  - such that $\text{PW}(G^*) \leq k^*$ iff **some** input i is YES.
- PATHWIDTH IMPROVEMENT

  NP-hard.

  Input: An integer k, and a graph G of pathwidth ≤ k-1.

  Question: Is the pathwidth of G at most k-2?

Tree obstructions to Pathwidth

- Kinnersley'92 and TakahashiUK'94 independently proved:

Tree obstructions to Pathwidth

- Kinnersley'92 and TakahashiUK'94 independently proved:

Tree obstructions to Pathwidth

- Kinnersley'92 and TakahashiUK'94 independently proved:

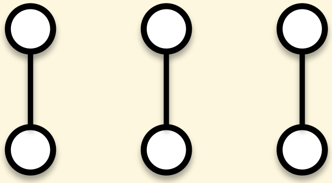
K_2 is the unique minimal obstruction to PW=0

Tree obstructions to Pathwidth

- Kinnersley'92 and TakahashiUK'94 independently proved:

Tree obstructions to Pathwidth

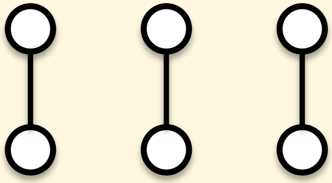
- Kinnersley'92 and TakahashiUK'94 independently proved:



Tree obstructions to Pathwidth

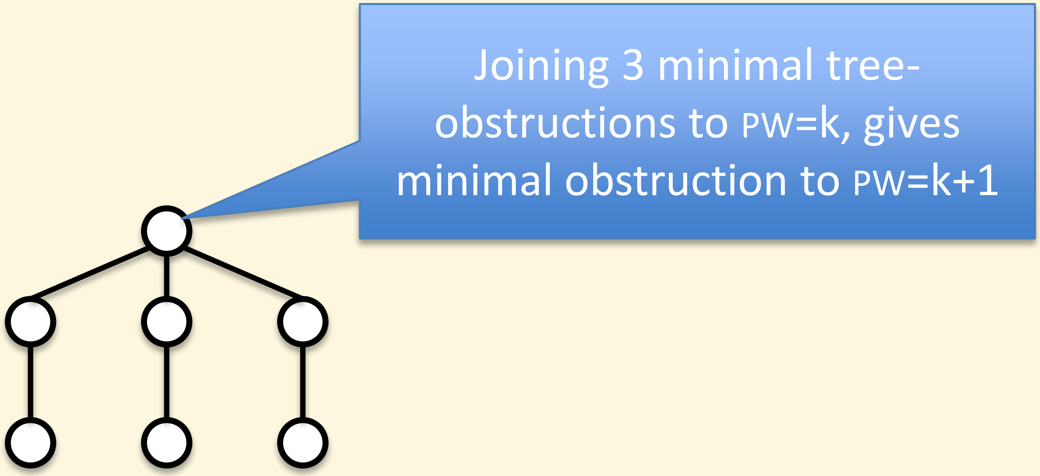
- Kinnersley'92 and TakahashiUK'94 independently proved:

Joining 3 minimal tree-obstructions to PW=k, gives minimal obstruction to PW=k+1



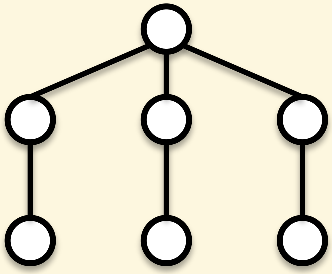
Tree obstructions to Pathwidth

- Kinnersley'92 and TakahashiUK'94 independently proved:



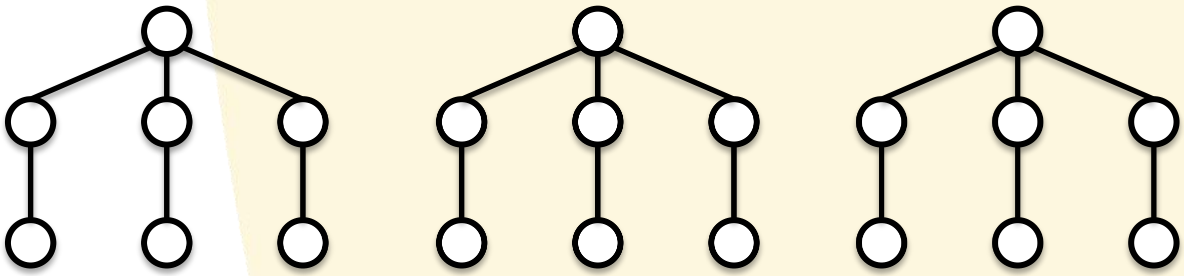
Tree obstructions to Pathwidth

- Kinnersley'92 and TakahashiUK'94 independently proved:



Tree obstructions to Pathwidth

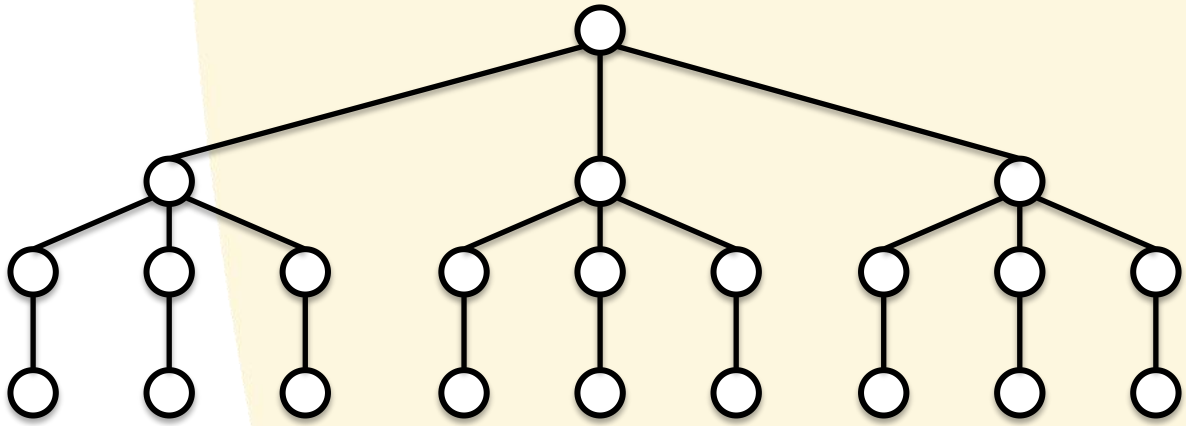
- Kinnersley’92 and TakahashiUK’94 independently proved:



Universiteit Utrecht

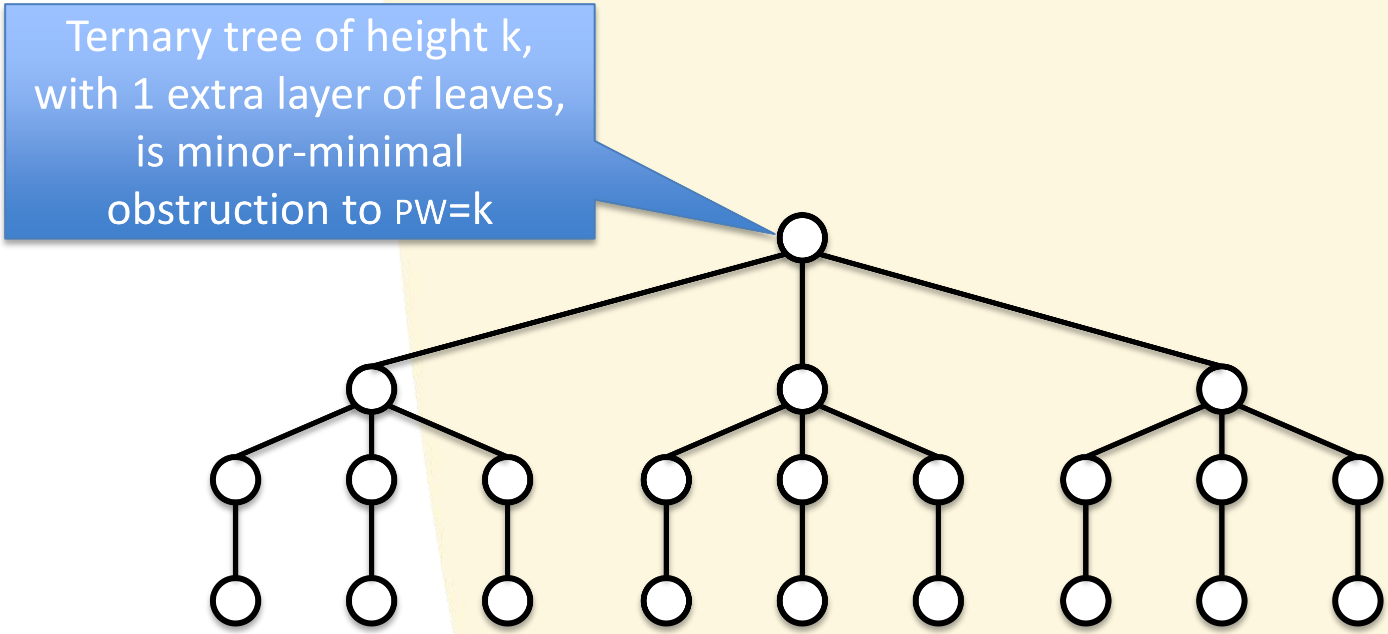
Tree obstructions to Pathwidth

- Kinnersley'92 and TakahashiUK'94 independently proved:



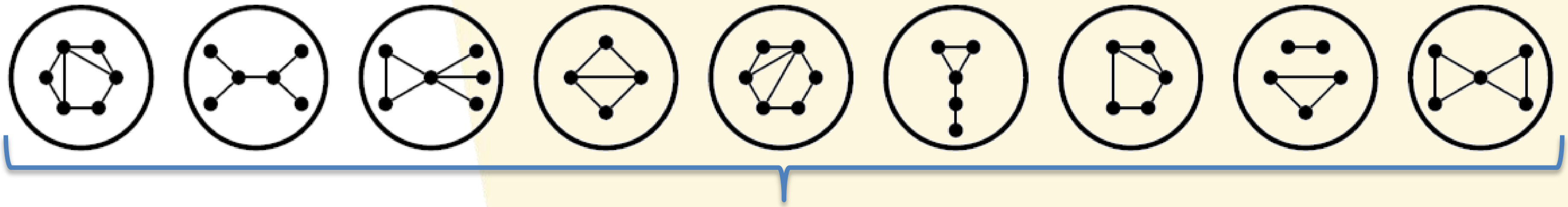
Tree obstructions to Pathwidth

- Kinnersley'92 and TakahashiUK'94 independently proved:



Construction

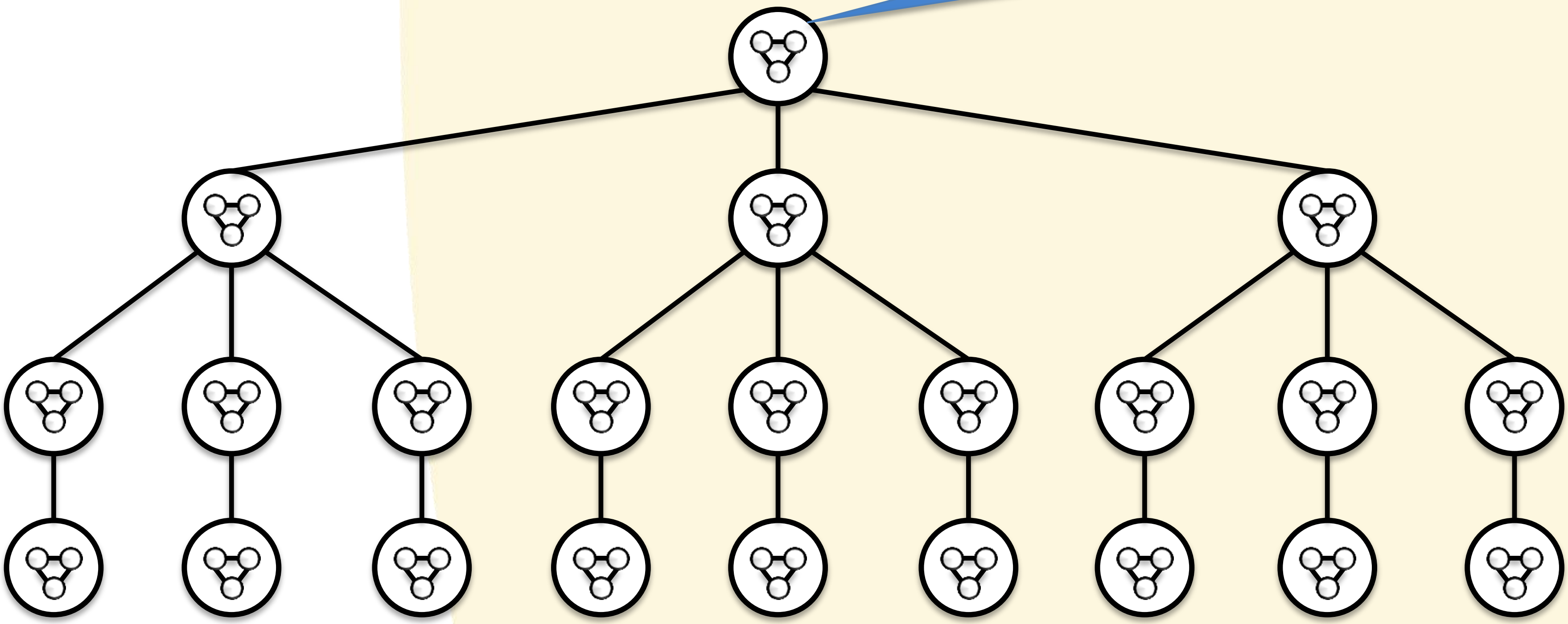
Construction



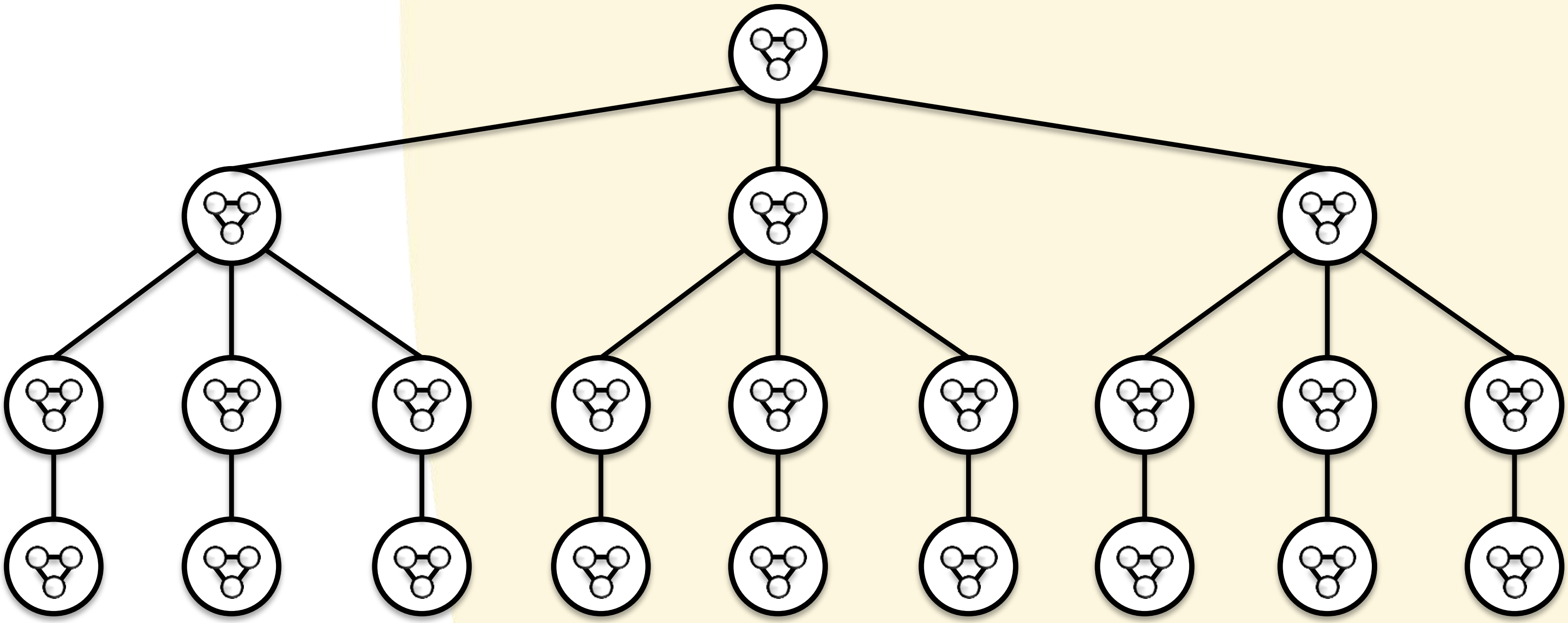
$t=3^s$ instances of PW-IMPROVEMENT with $k=3$
(each asking if $\text{PW}(G_i) \leq k - 2$)

Construction

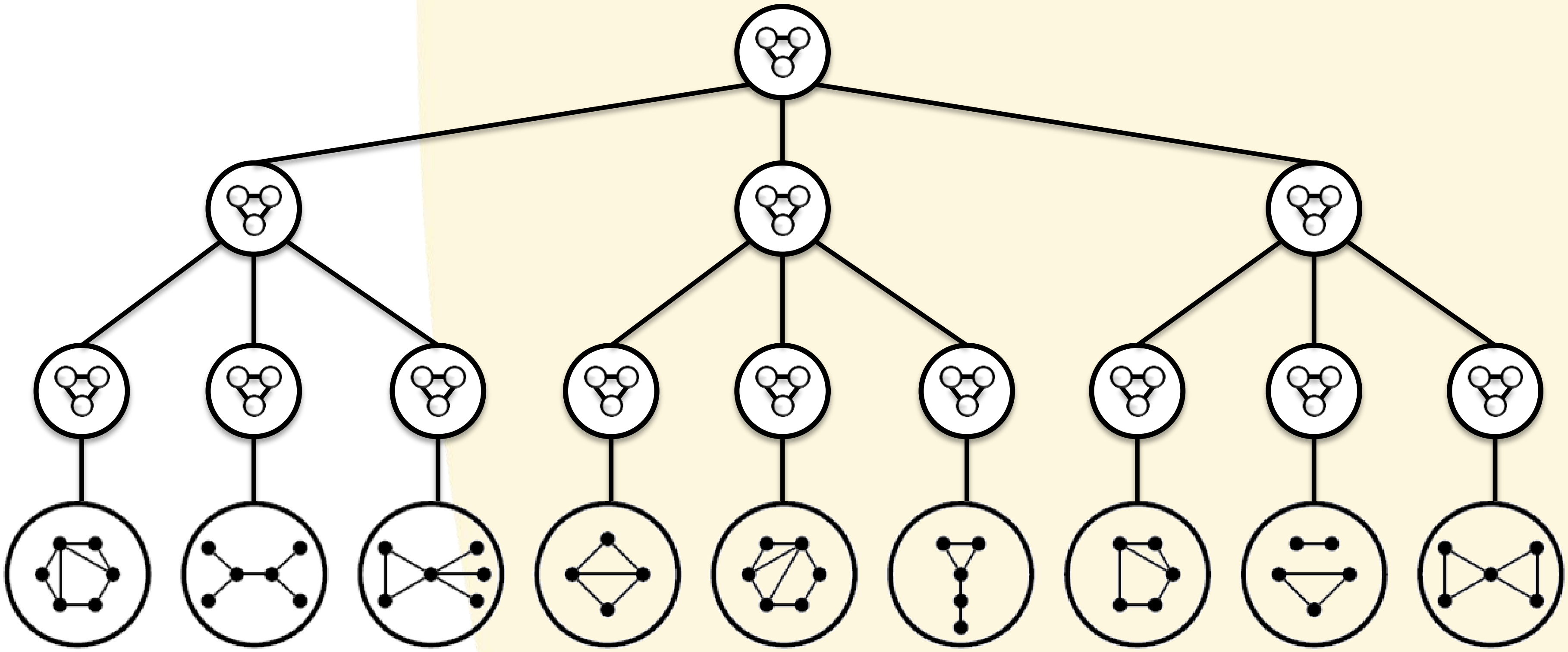
Obstruction with 3^s leaves,
inflated by factor k
Pathwidth is $\mathcal{O}(k \cdot s) \leq \mathcal{O}(n \cdot \log t)$



Construction

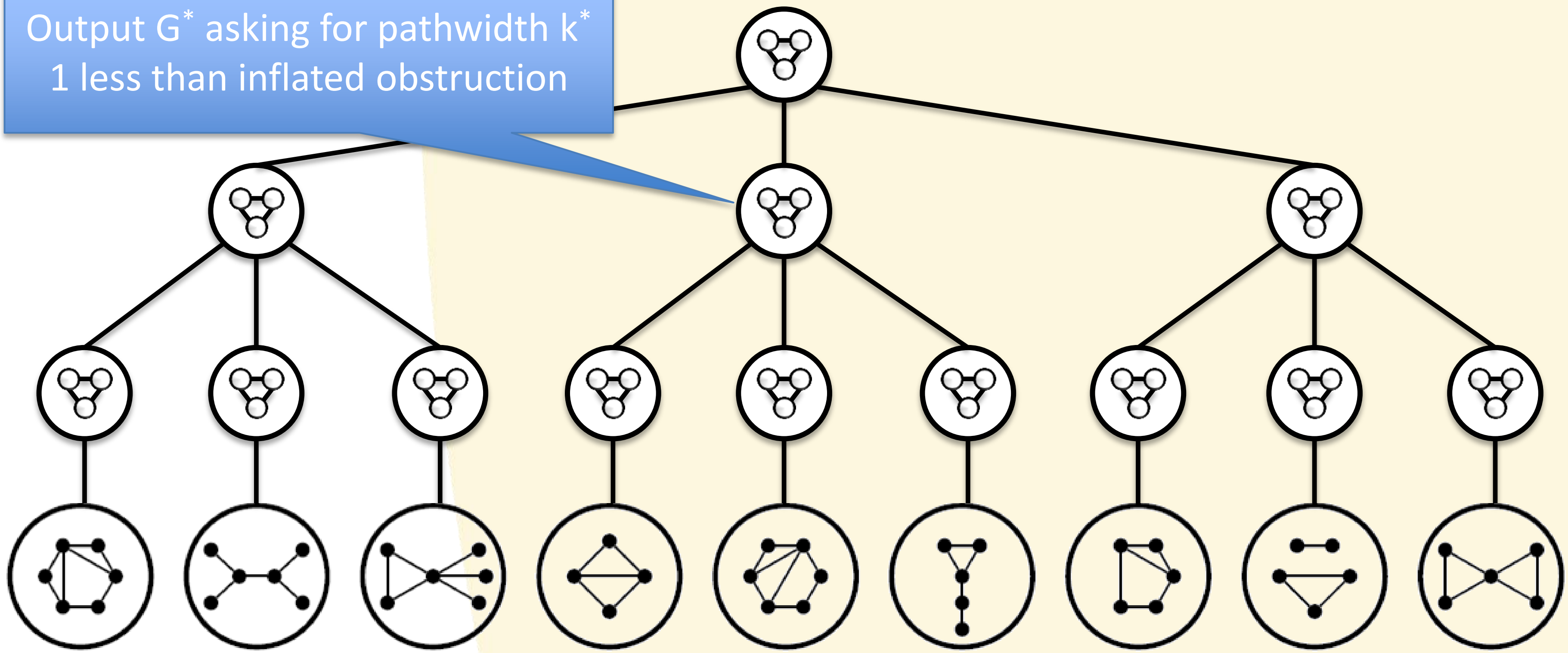


Construction

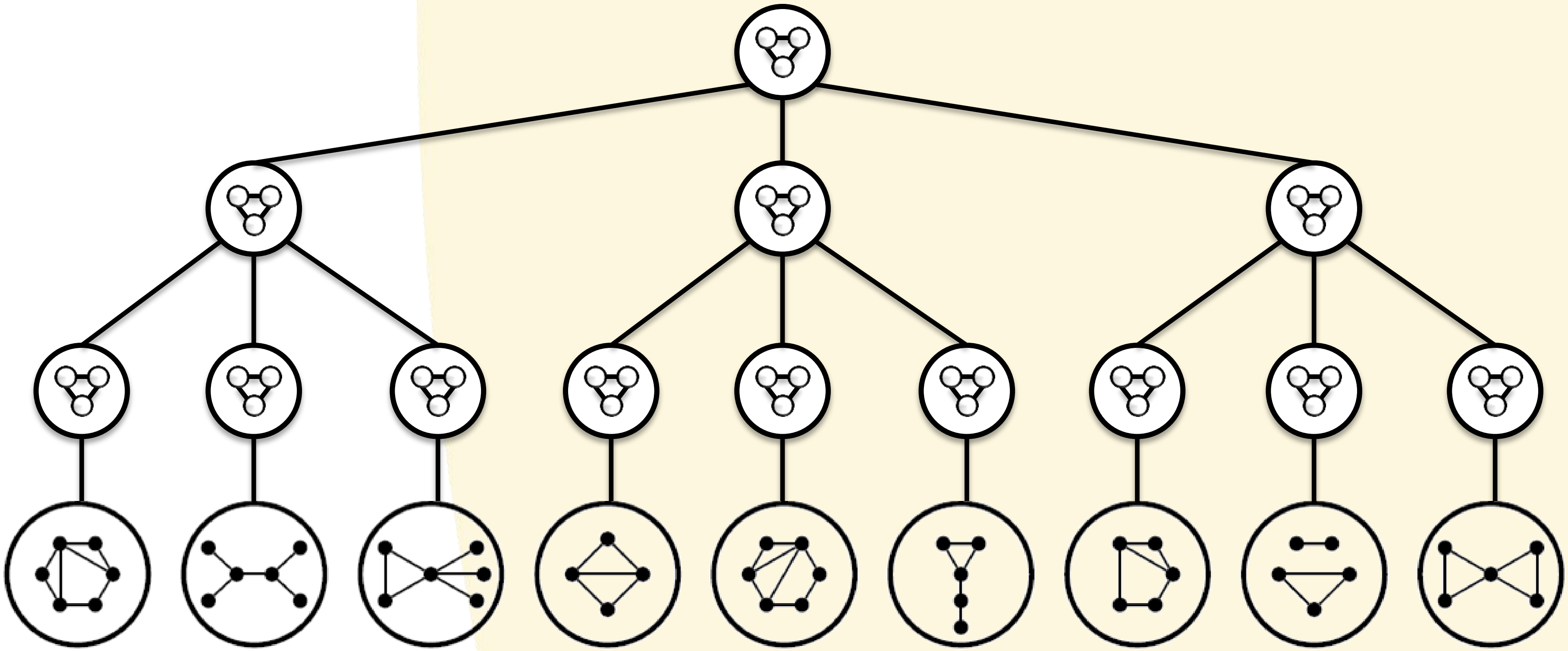


Construction

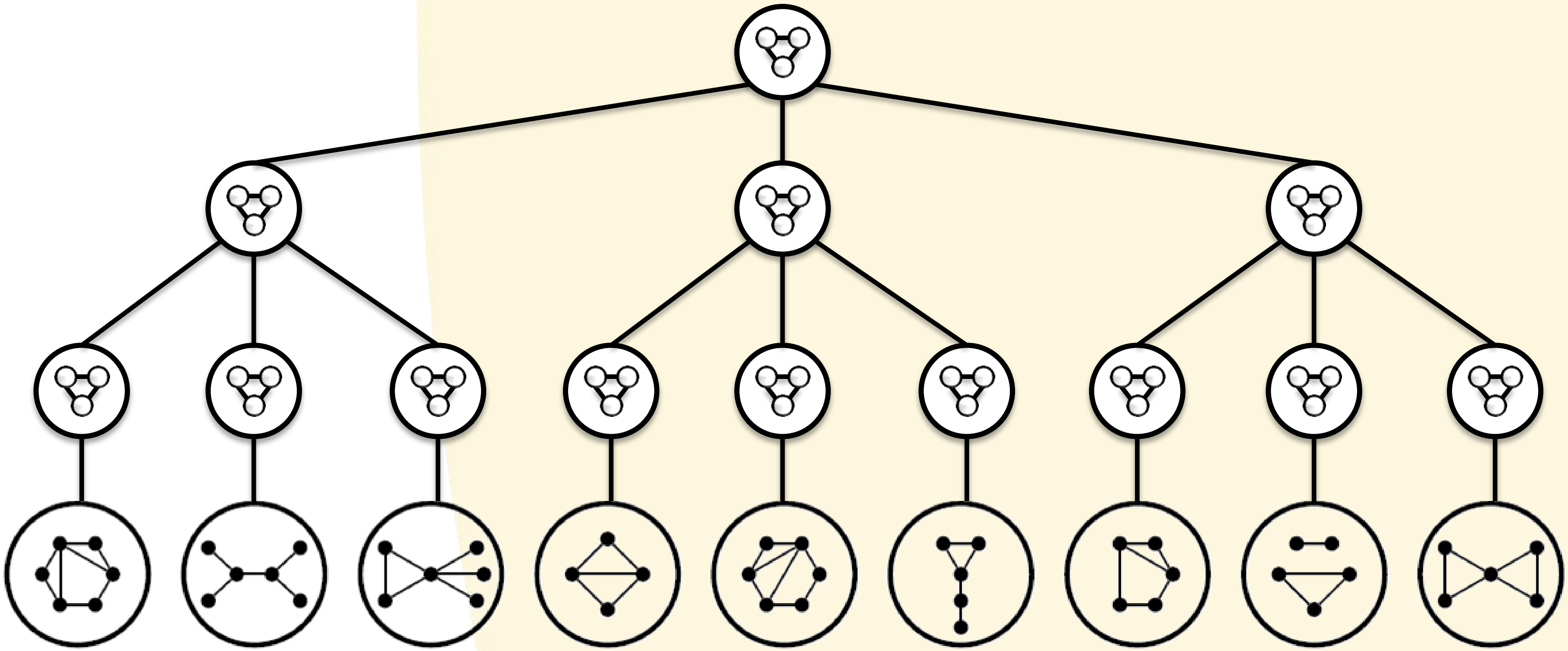
Output G^* asking for pathwidth k^*
1 less than inflated obstruction



Correctness sketch

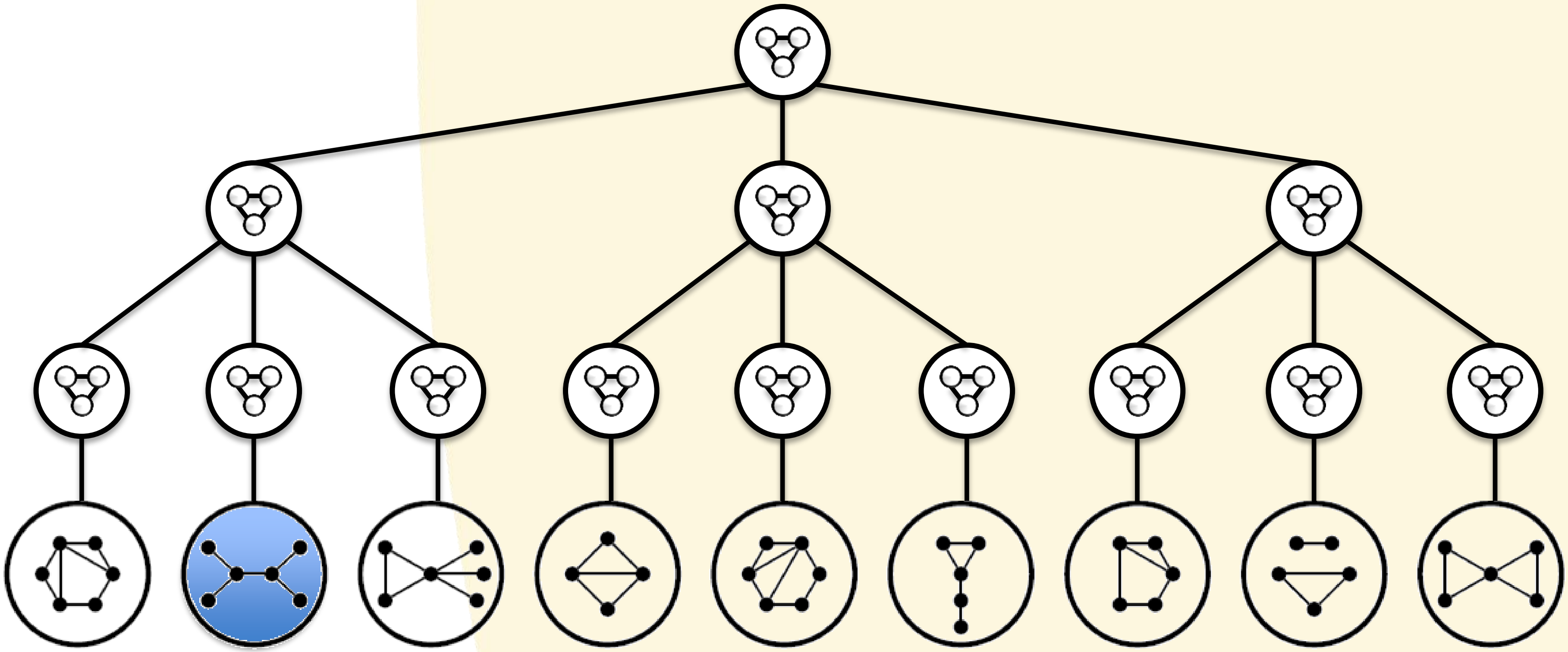


Correctness sketch



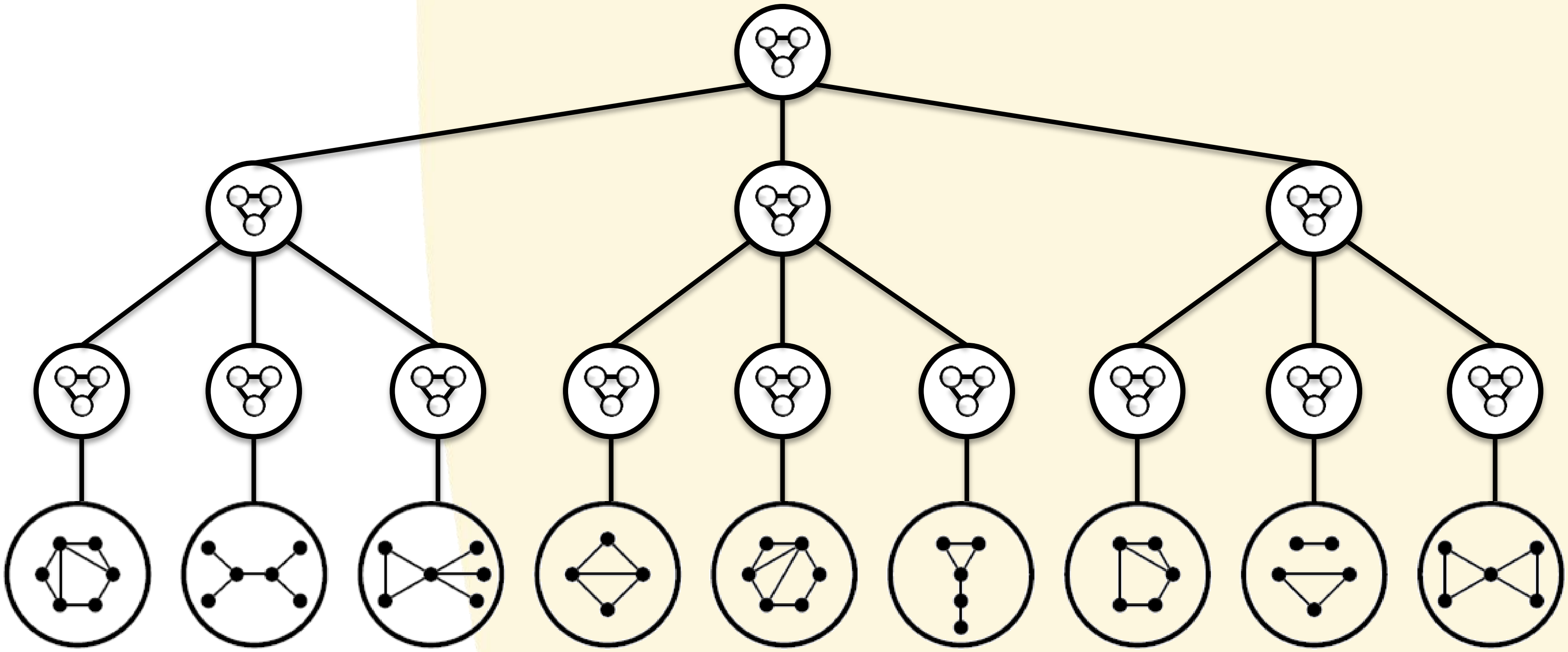
Claim: **some** input i has $\mathrm{PW}(G_i) \leq k\text{-}2 \rightarrow \mathrm{PW}(G^*) < \mathrm{PW}(T^s \Diamond k)$

Correctness sketch

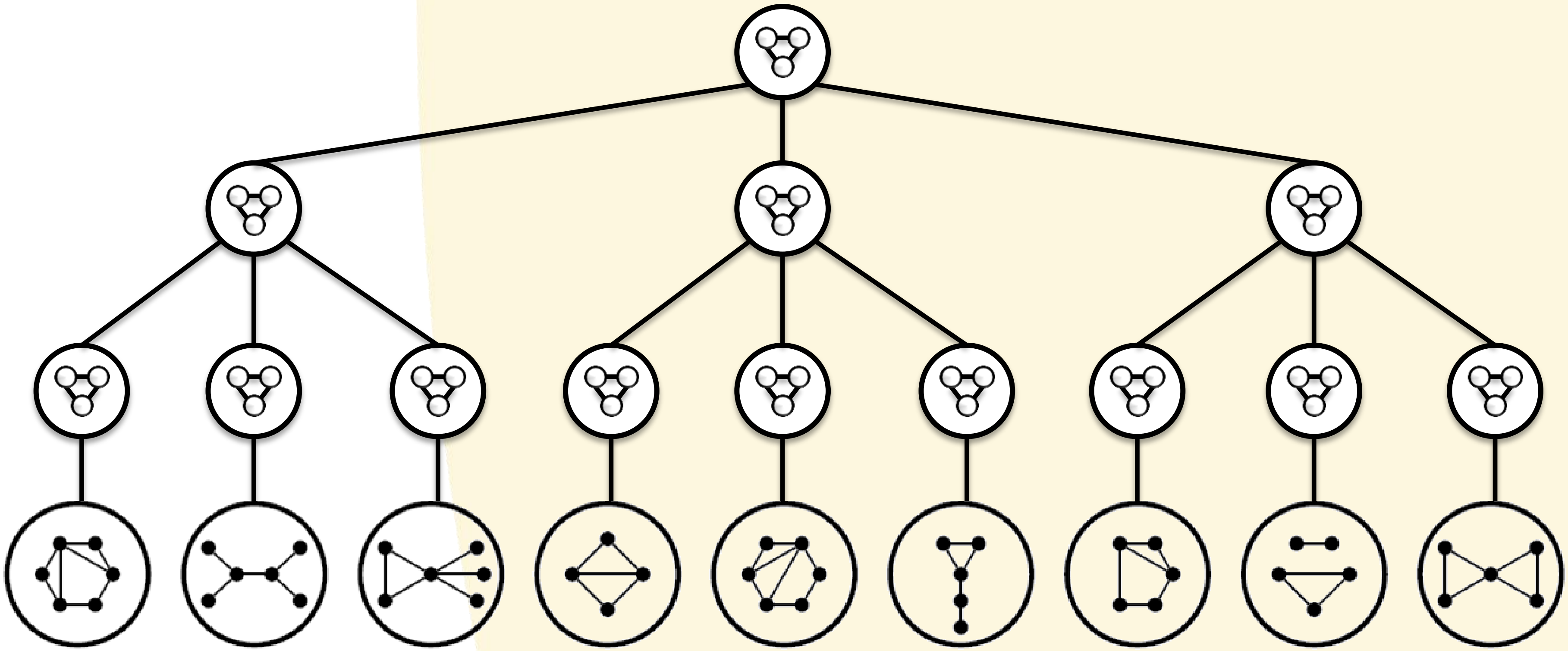


Claim: **some** input i has PW(G_i) ≤ k-2 → PW(G^*) < PW(T^s ◊ k)

Correctness sketch



Correctness sketch



Claim: **all** inputs have PW(G_i) > k-2 ➔ PW(G^*) ≥ PW(T^s ◊ k)

Conclusion

Conclusion

- For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance (x,k) is preceded by an f(k)-size obstruction

Conclusion

- For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance (x,k) is preceded by an f(k)-size obstruction
- Characterization suggests a connection between kernel sizes and obstruction sizes

Conclusion

- For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance (x,k) is preceded by an f(k)-size obstruction
- Characterization suggests a connection between kernel sizes and obstruction sizes
- Large obstructions form the crucial ingredient for OR-cross-composition into *k*-PATHWIDTH

Conclusion

- For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance (x,k) is preceded by an f(k)-size obstruction
- Characterization suggests a connection between kernel sizes and obstruction sizes
- Large obstructions form the crucial ingredient for OR-cross-composition into *k*-PATHWIDTH

- Open problems:

Conclusion

- For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance (x,k) is preceded by an f(k)-size obstruction
- Characterization suggests a connection between kernel sizes and obstruction sizes
- Large obstructions form the crucial ingredient for OR-cross-composition into *k*-PATHWIDTH

- Open problems:

OR-Cross-composition into *k*-TREEWIDTH?

Further relations between kernel and obstruction sizes?

Conclusion

- For each problem in FPT, there is a polynomial-time quasi-order under which each NO-instance (x,k) is preceded by an f(k)-size obstruction
- Characterization suggests a connection between kernel sizes and obstru…
- Large c… …-cross-com…

THANK YOU!

- Open prob…

OR-Cross-composition into *k*-TREEWIDTH?

Further relations between kernel and obstruction sizes?