Parameterized Algorithms

Exercise 5 – Sheet 4 & 5, Kernel Lower Bounds

Elly, Jean-Pierre, Wendy

Sheet 4 – Makespan Scheduling

Given m machines, n jobs with processing times $p_1, p_2, \ldots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?

Sheet 4 – Makespan Scheduling

Given m machines, n jobs with processing times $p_1, p_2, \dots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?

Metatheorem
A parameterized problem with parameter k that can be formulated as an ILP with $f(k)$ many variables is in FPT.

Sheet 4 – Makespan Scheduling

Given m machines, n jobs with processing times $p_1, p_2, \ldots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?

Metatheorem
A parameterized problem with parameter k that can be formulated as an ILP with $f(k)$ many variables is in FPT.

Sheet 4 – Makespan Scheduling

Given m machines, n jobs with processing times $p_1, p_2, \ldots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?

Observation

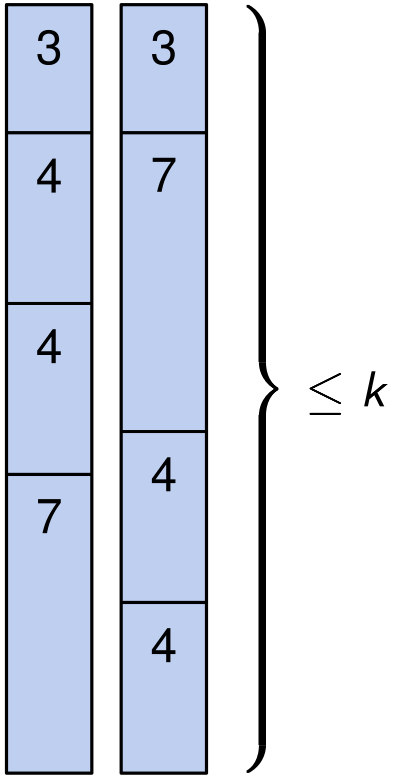
- Only processing time matters, not job itself nor ordering for fixed machine
- The number of possible schedules for one machine only depends on k

  → set S

Metatheorem
A parameterized problem with parameter k that can be formulated as an ILP with $f(k)$ many variables is in FPT.

Sheet 4 – Makespan Scheduling

Given m machines, n jobs with processing times $p_1, p_2, \ldots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?



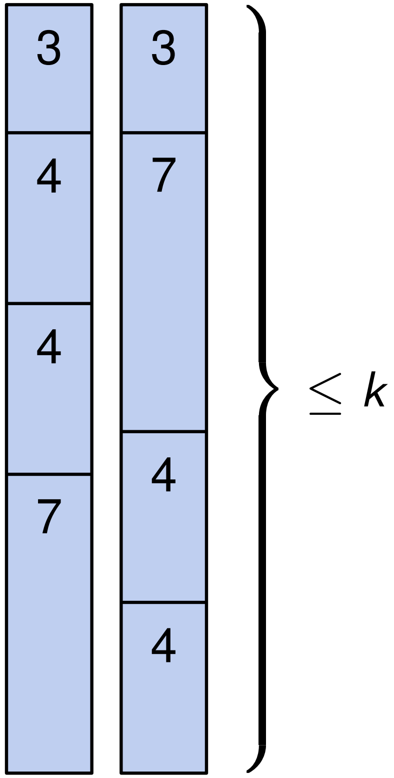
Observation

- Only processing time matters, not job itself nor ordering for fixed machine
- The number of possible schedules for one machine only depends on k (Partition number $\leq k^k$)

→ set S

Metatheorem
A parameterized problem with parameter k that can be formulated as an ILP with $f(k)$ many variables is in FPT.

Sheet 4 – MAKESPAN SCHEDULING



Given m machines, n jobs with processing times $p_1, p_2, \ldots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?

Observation

- Only processing time matters, not job itself nor ordering for fixed machine
- The number of possible schedules for one machine only depends on k (Partition number $\leq k^k$)

→ set S

Constants:

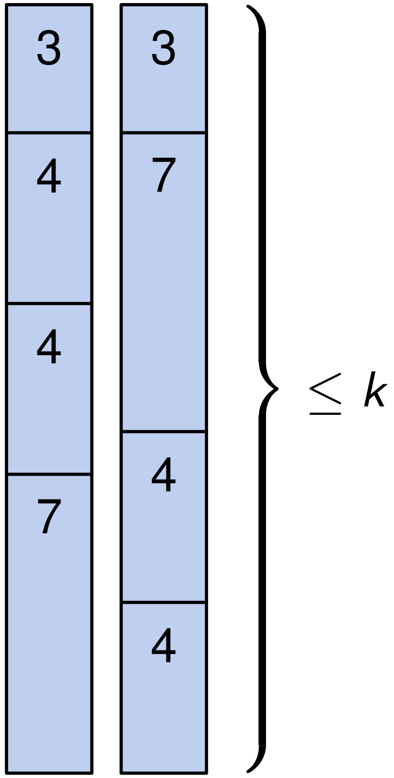
- c_ℓ: number of jobs of length ℓ
- $d_{s,\ell}$: number of jobs with length ℓ in schedule s

Variables:

- $x_s \geq 0$: number of machines with schedule s

Metatheorem
A parameterized problem with parameter k that can be formulated as an ILP with $f(k)$ many variables is in FPT.

Sheet 4 – Makespan Scheduling



Given m machines, n jobs with processing times $p_1, p_2, \ldots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?

Observation

- Only processing time matters, not job itself nor ordering for fixed machine
- The number of possible schedules for one machine only depends on k (Partition number $\leq k^k$)

→ set S

Constants:

- c_ℓ: number of jobs of length ℓ
- $d_{s,\ell}$: number of jobs with length ℓ in schedule s

Variables:

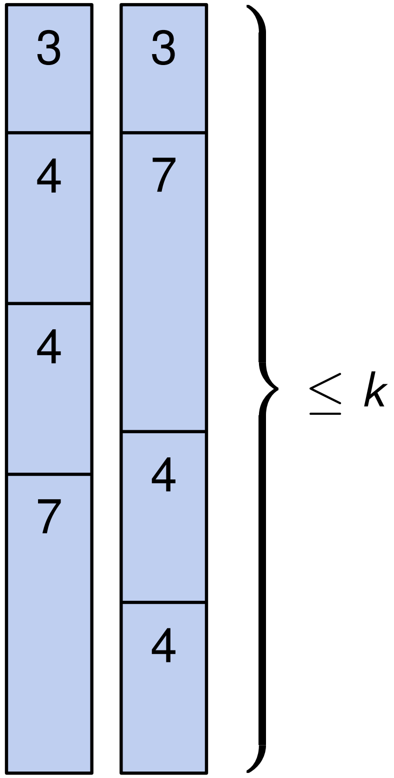
- $x_s \geq 0$: number of machines with schedule s

There are as many assigend schedules as there are machines

Every job gets processed

Sheet 4 – Makespan Scheduling

Given m machines, n jobs with processing times $p_1, p_2, \ldots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?



Observation

- Only processing time matters, not job itself nor ordering for fixed machine
- The number of possible schedules for one machine only depends on k (Partition number $\leq k^k$)

→ set S

Constants:

- c_ℓ: number of jobs of length ℓ
- $d_{s,\ell}$: number of jobs with length ℓ in schedule s

Variables:

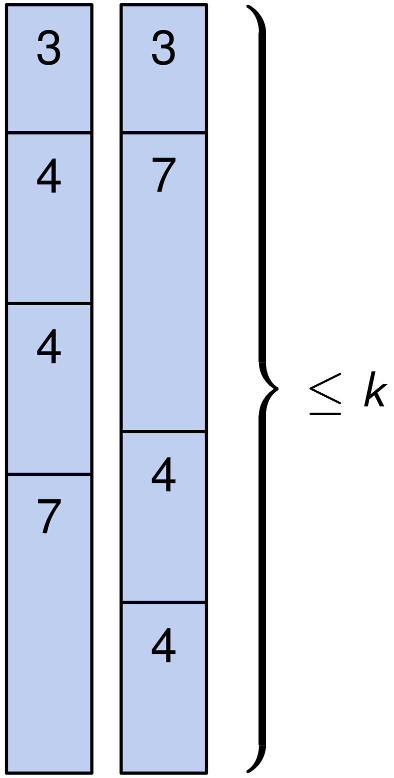
- $x_s \geq 0$: number of machines with schedule s

There are as many assigend schedules as there are machines

$$\sum_{s \in S} x_s = m$$

Every job gets processed

Sheet 4 – Makespan Scheduling



Given m machines, n jobs with processing times $p_1, p_2, \ldots, p_n \in \mathbb{N}$ and maximum processing time k. Is there an assignments of jobs to machines s.t. no machine requires more than k time to process its jobs?

Observation

- Only processing time matters, not job itself nor ordering for fixed machine
- The number of possible schedules for one machine only depends on k (Partition number $\leq k^k$)

→ set S

Constants:

- c_ℓ: number of jobs of length ℓ
- $d_{s,\ell}$: number of jobs with length ℓ in schedule s

Variables:

- $x_s \geq 0$: number of machines with schedule s

There are as many assigend schedules as there are machines

$$\sum_{s \in S} x_s = m$$

Every job gets processed

$$\sum_{s \in S} d_{s,\ell} \cdot x_s = c_\ell \quad \text{for all } \ell \in \{1, \ldots, k\}$$

Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

Case 1: $m > 2k$

Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

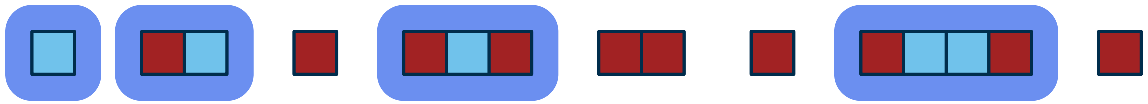
Case 1: $m > 2k$

- Take any arbitrary assignment

Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables



Case 1: $m > 2k$

- Take any arbitrary assignment

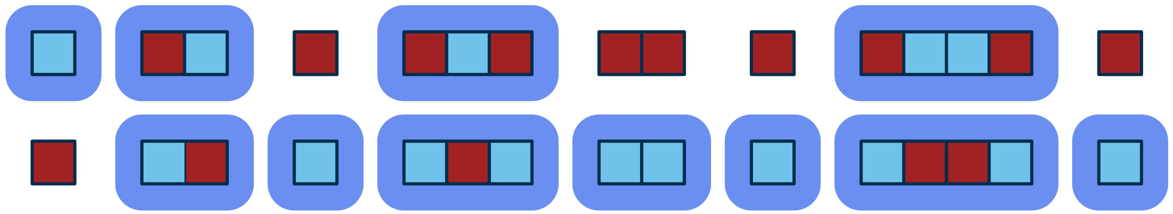
Sheet 4 – Max Sat

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

Case 1: $m > 2k$

- Take any arbitrary assignment

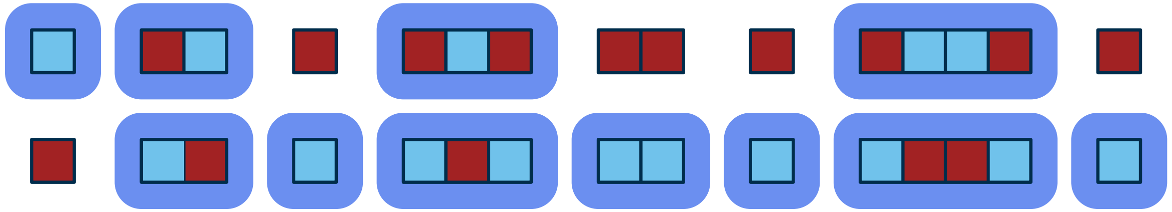


Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

Case 1: $m > 2k$

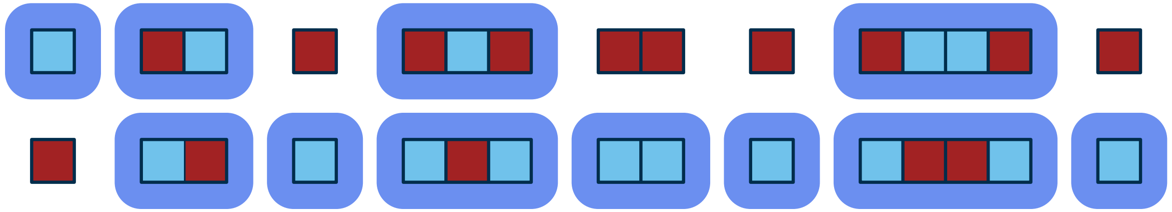


- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables



Case 1: $m > 2k$

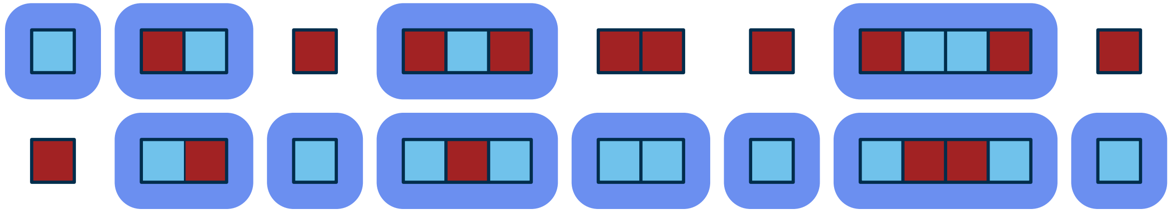
- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

Case 2: $n > k$

Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

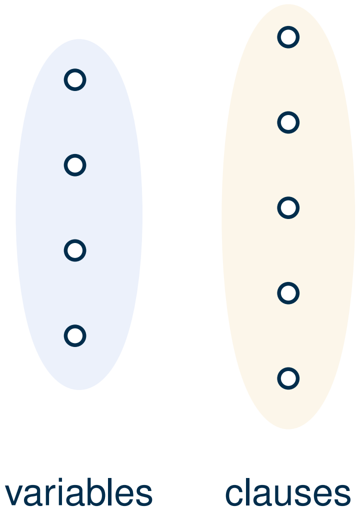
(a): Kernel with $2k$ clauses and k variables



Case 1: $m > 2k$

- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

Case 2: $n > k$

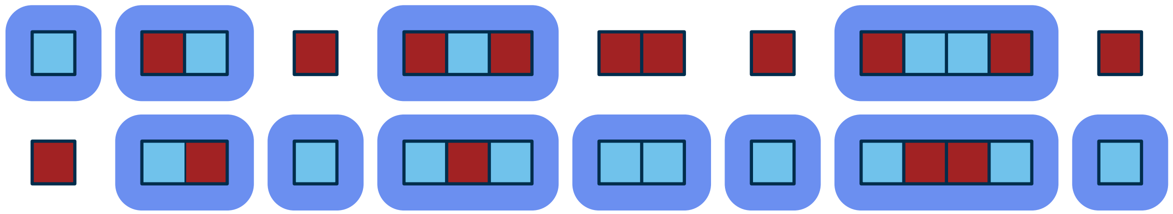


Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

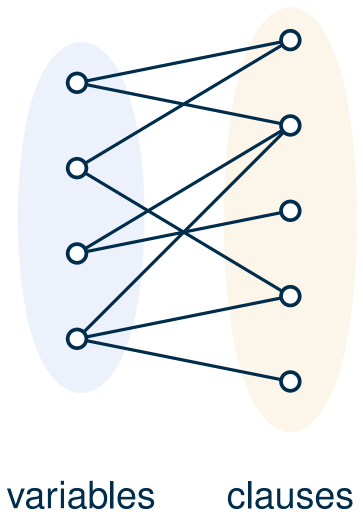
(a): Kernel with $2k$ clauses and k variables

Case 1: $m > 2k$



- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

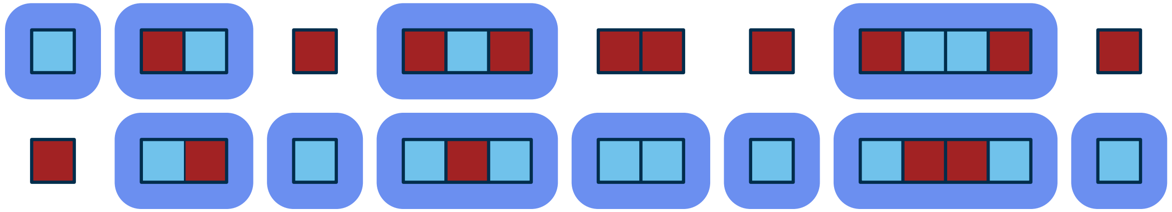
Case 2: $n > k$



Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

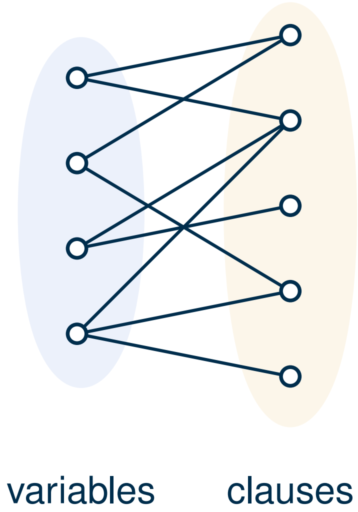
(a): Kernel with $2k$ clauses and k variables

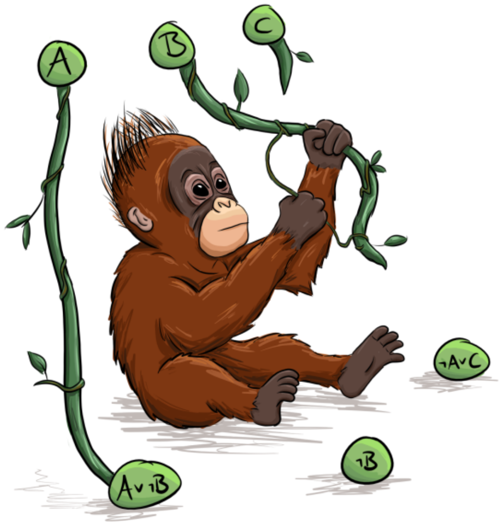


Case 1: $m > 2k$

- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

Case 2: $n > k$



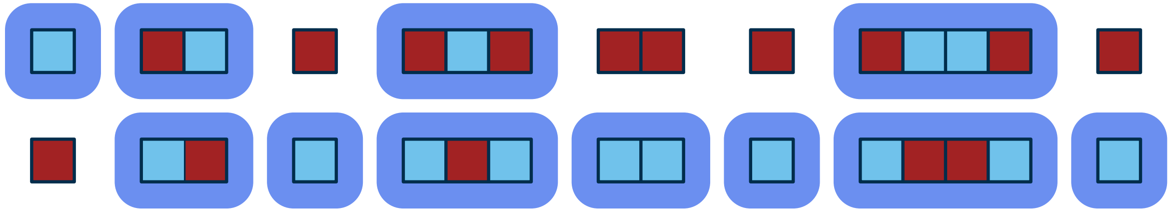


Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

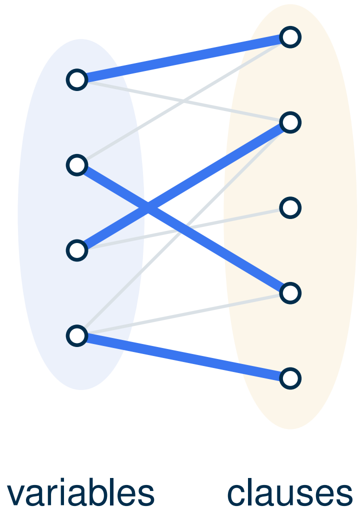
Case 1: $m > 2k$

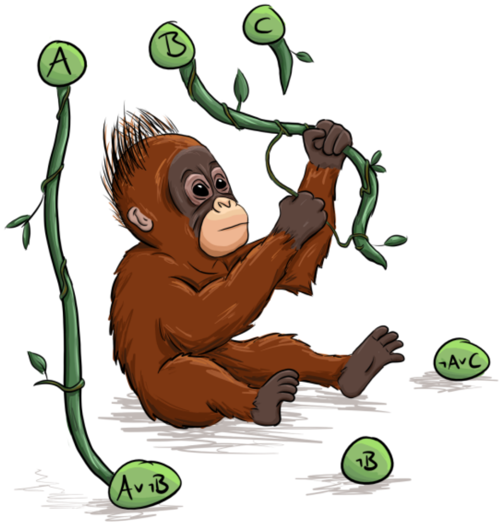


- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

Case 2: $n > k$

- If all variable vertices can be matched: yes instance





Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

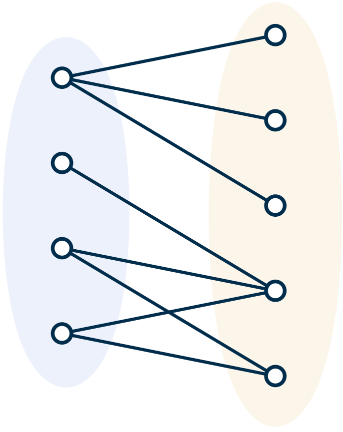
Case 1: $m > 2k$

Hall's Theorem: Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. There exists a matching in G that covers all vertices of V_1 if and only if $|X| \leq |N(X)|$ for every subset $X \subseteq V_1$.
Otherwise, an inclusion-minimal set $X \subseteq V_1$ with $|X| > |N(X)|$ can be efficiently found.

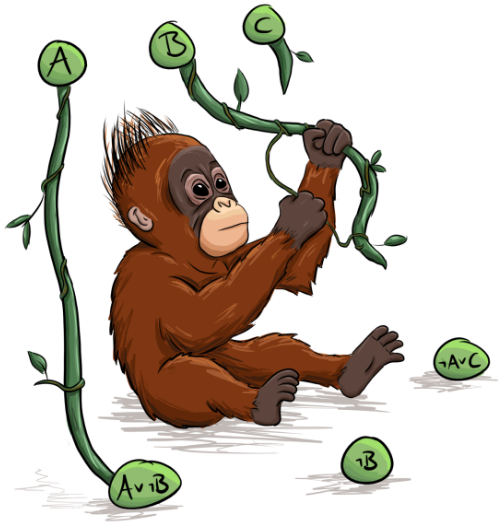
- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

Case 2: $n > k$

- If all variable vertices can be matched: yes instance
- Else:



variables clauses



Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

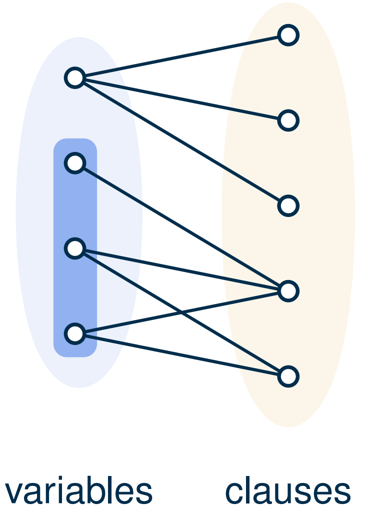
Case 1: $m > 2k$

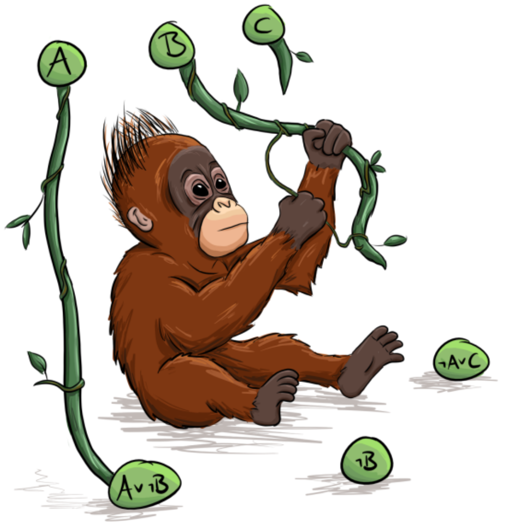
Hall's Theorem: Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. There exists a matching in G that covers all vertices of V_1 if and only if $|X| \leq |N(X)|$ for every subset $X \subseteq V_1$.
Otherwise, an inclusion-minimal set $X \subseteq V_1$ with $|X| > |N(X)|$ can be efficiently found.

- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

Case 2: $n > k$

- If all variable vertices can be matched: yes instance
- Else: we can find minimal set of variables X with $|X| > |N(X)|$





Sheet 4 – Max Sat

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

Case 1: $m > 2k$

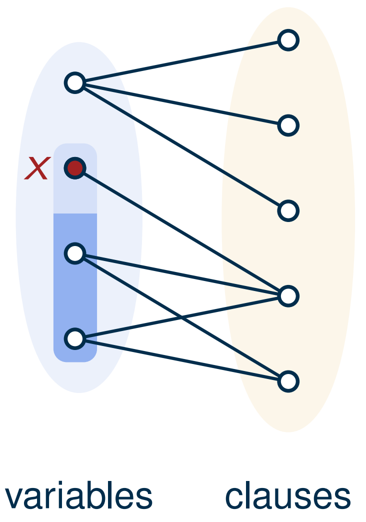
Hall's Theorem: Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. There exists a matching in G that covers all vertices of V_1 if and only if $|X| \leq |N(X)|$ for every subset $X \subseteq V_1$.
Otherwise, an inclusion-minimal set $X \subseteq V_1$ with $|X| > |N(X)|$ can be efficiently found.

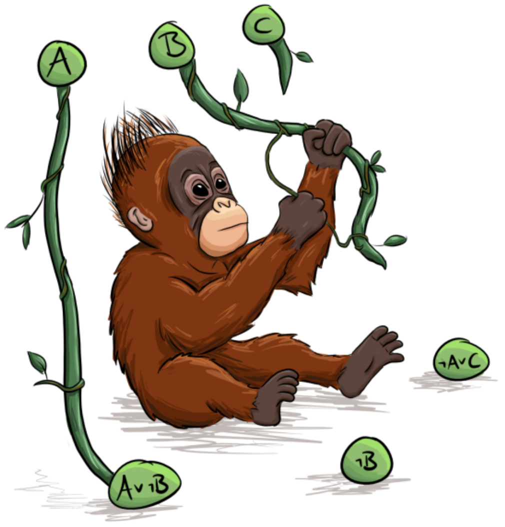
- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

Case 2: $n > k$

- If all variable vertices can be matched: yes instance
- Else: we can find minimal set of variables X with $|X| > |N(X)|$

minimality of X $\Longrightarrow$ for $x \in X$ all subsets $X' \subseteq X \setminus x$ satisfy $|X'| \leq |N(X')|$





Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

Case 1: $m > 2k$

Hall's Theorem: Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. There exists a matching in G that covers all vertices of V_1 if and only if $|X| \leq |N(X)|$ for every subset $X \subseteq V_1$.
Otherwise, an inclusion-minimal set $X \subseteq V_1$ with $|X| > |N(X)|$ can be efficiently found.

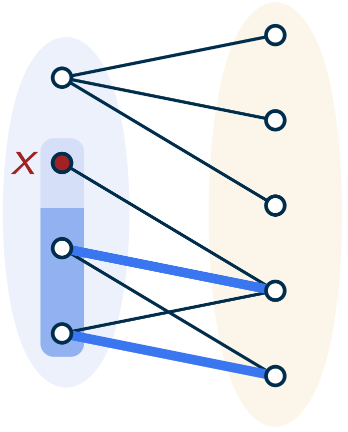
- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

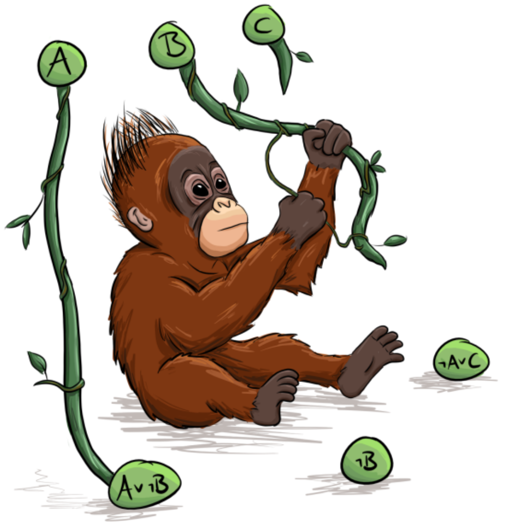
Case 2: $n > k$

- If all variable vertices can be matched: yes instance
- Else: we can find minimal set of variables X with $|X| > |N(X)|$

$\overset{\text{minimality of } X}{\Longrightarrow}$ for $x \in X$ all subsets $X' \subseteq X \setminus x$ satisfy $|X'| \leq |N(X')|$

$\overset{\text{Hall}}{\Longrightarrow}$ all vertices in $X \setminus x$ can be matched





Sheet 4 – MAX SAT

Given a logical formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least k clauses are satisfied?

(a): Kernel with $2k$ clauses and k variables

Case 1: $m > 2k$

Hall's Theorem: Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. There exists a matching in G that covers all vertices of V_1 if and only if $|X| \leq |N(X)|$ for every subset $X \subseteq V_1$.
Otherwise, an inclusion-minimal set $X \subseteq V_1$ with $|X| > |N(X)|$ can be efficiently found.

- Take any arbitrary assignment
- This or its complement satifies at least half of the clauses! $\Rightarrow$ yes instance

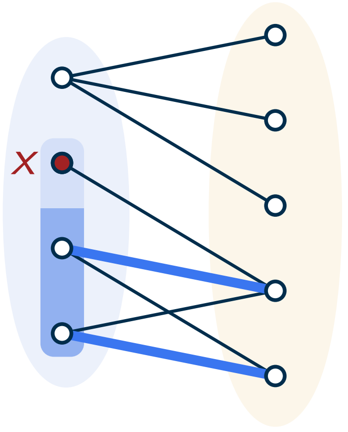
Case 2: $n > k$

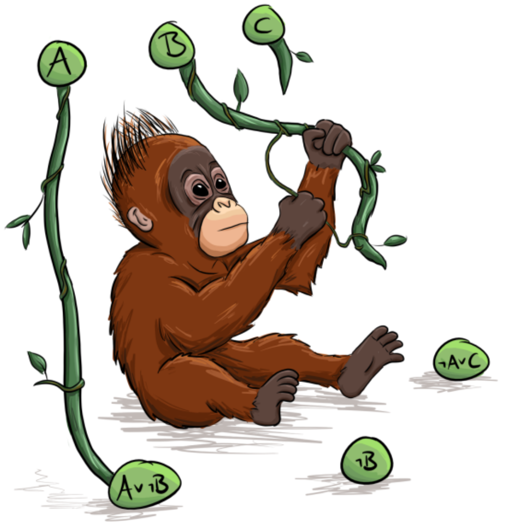
- If all variable vertices can be matched: yes instance
- Else: we can find minimal set of variables X with $|X| > |N(X)|$

$\overset{\text{minimality of } X}{\Longrightarrow}$ for $x \in X$ all subsets $X' \subseteq X \setminus x$ satisfy $|X'| \leq |N(X')|$

$\overset{\text{Hall}}{\Longrightarrow}$ all vertices in $X \setminus x$ can be matched

- delete X and $N(X)$, reduce parameter by $|N(X)|$





Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

Sheet 4 – Max Sat Above $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?

Sheet 4 – Max Sat Above $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?

  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged

Sheet 4 – Max Sat Above $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged
  } apply exhaustively

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged
  } apply exhaustively
- let $u := \#$unary clauses, $g := \#$larger clauses

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged

  } apply exhaustively

- let $u := \#$unary clauses, $g := \#$larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)

Sheet 4 – Max Sat Above $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged

  } apply exhaustively

- let $u := \#$unary clauses, $g := \#$larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

Sheet 4 – Max Sat Above $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged

  } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)
- consider number X of satisfied clauses in uniformly distributed variable assignment

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged

  } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)
- consider number X of satisfied clauses in uniformly distributed variable assignment

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

$E[X]$

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged
  } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)
- consider number X of satisfied clauses in uniformly distributed variable assignment

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right]$$

X_i : is clause i satisfied?

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged

  } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

- consider number X of satisfied clauses in uniformly distributed variable assignment

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right] = \sum_{i=1}^{m} E[X_i]$$

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)
- consider number X of satisfied clauses in uniformly distributed variable assignment

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right] = \sum_{i=1}^{m} E\left[X_i\right] \geq \frac{u}{2} + \frac{3}{4}g$$

$P(\text{“unary clause is satisfied”}) = \frac{1}{2}$

$P(\text{“larger clause is satisfied”}) \geq \frac{3}{4}$

Sheet 4 – Max Sat Above $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged

  } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

- consider number X of satisfied clauses in uniformly distributed variable assignment

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right] = \sum_{i=1}^{m} E\left[X_i\right] \geq \frac{u}{2} + \frac{3}{4}g = \frac{m}{2} + \frac{g}{4}$$

↑ $u + g = m$

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged
  } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

- consider number X of satisfied clauses in uniformly distributed variable assignment

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right] = \sum_{i=1}^{m} E\left[X_i\right] \geq \frac{u}{2} + \frac{3}{4}g = \frac{m}{2} + \frac{g}{4}$$

$\Rightarrow \frac{m}{2} + \frac{g}{4}$ clauses are satisfiable

(↑ $u + g = m$)

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged
  } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

- consider number X of satisfied clauses in uniformly distributed variable assignment

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right] = \sum_{i=1}^{m} E[X_i] \geq \frac{u}{2} + \frac{3}{4}g = \frac{m}{2} + \frac{g}{4}$$

$\Rightarrow \frac{m}{2} + \frac{g}{4}$ clauses are satisfiable

$u + g = m$

- if $\frac{g}{4} \geq k$: yes instance; else: $g < 4k$

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged

  } apply exhaustively

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)
- consider number X of satisfied clauses in uniformly distributed variable assignment

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right] = \sum_{i=1}^{m} E[X_i] \geq \frac{u}{2} + \frac{3}{4}g = \frac{m}{2} + \frac{g}{4}$$

$\uparrow$ $u + g = m$

$\Rightarrow \frac{m}{2} + \frac{g}{4}$ clauses are satisfiable

- if $\frac{g}{4} \geq k$: yes instance; else: $g < 4k$
- $m = u + g < \frac{m}{2} + k + 4k = \frac{m}{2} + 5k \Rightarrow m < 10k$

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged } apply exhaustively
- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)
- consider number X of satisfied clauses in uniformly distributed variable assignment

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right] = \sum_{i=1}^{m} E[X_i] \geq \frac{u}{2} + \frac{3}{4}g = \frac{m}{2} + \frac{g}{4} \qquad \Rightarrow \frac{m}{2} + \frac{g}{4} \text{ clauses are satisfiable}$$

($u + g = m$)

- if $\frac{g}{4} \geq k$: yes instance; else: $g < 4k$
- $m = u + g < \frac{m}{2} + k + 4k = \frac{m}{2} + 5k \Rightarrow m < 10k$
- new question: are $\frac{m}{2} + k < 6k$ clauses satisfiable?

Sheet 4 – MAX SAT ABOVE $\frac{m}{2}$

Given formula φ in CNF with n variables and m clauses, can we assign the variables s.t. at least $\frac{m}{2} + k$ clauses are satisfied?

Part (b) Find FPT Algo

- **Reduction rule 1:** Are there two unary clauses $\{v\}$ und $\{\neg v\}$ with same variable v?
  each assignment satisfies exactly one of them $\rightarrow$ delete the two clauses, parameter unchanged
  } apply exhaustively

Probabilistic method: $E[X] \geq k \Rightarrow Pr[X \geq k] > 0$

- let $u :=$ #unary clauses, $g :=$ #larger clauses
- after rule 1: $u < \frac{m}{2} + k$ (or yes instance)
- consider number X of satisfied clauses in uniformly distributed variable assignment

$$E[X] = E\left[\sum_{i=1}^{m} X_i\right] = \sum_{i=1}^{m} E[X_i] \geq \frac{u}{2} + \frac{3}{4}g = \frac{m}{2} + \frac{g}{4}$$

$\Rightarrow \frac{m}{2} + \frac{g}{4}$ clauses are satisfiable

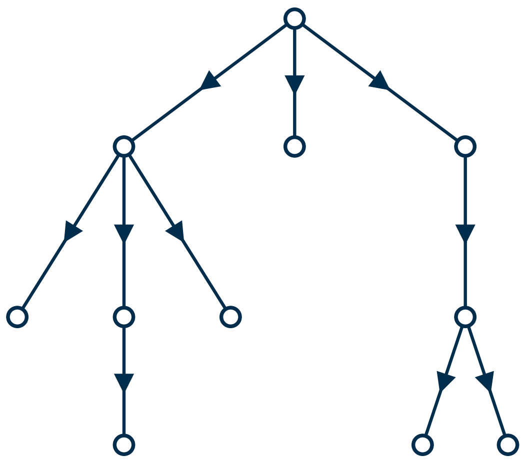
($u + g = m$)

- if $\frac{g}{4} \geq k$: yes instance; else: $g < 4k$
- $m = u + g < \frac{m}{2} + k + 4k = \frac{m}{2} + 5k \Rightarrow m < 10k$
- new question: are $\frac{m}{2} + k < 6k$ clauses satisfiable? $\Rightarrow$ use part (a)

Sheet 4 – WEIGHTED VERTEX COVER ON TREES

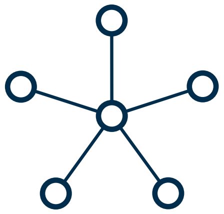
Given a vertex-weighted undirected tree T. Formulate a dynamic program that finds the minimum weight of a vertex cover in T.

- finding the recursion formulas worked generally well
- don't forget the stuff around it:
  - root the tree at arbitrary vertex
    (otherwise there are no parents or children)
  - traverse the tree bottom-up (BFS)
  - argue runtime $\mathcal{O}(n)$



Sheet 5 – Color Coding

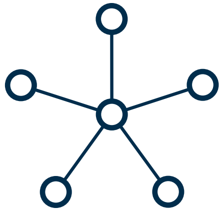
5-Stars:



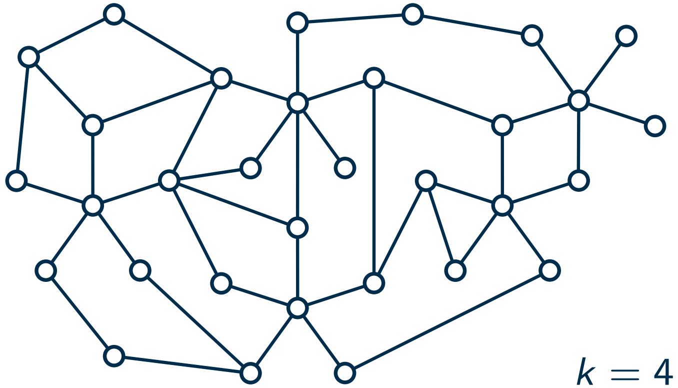
Given a graph G and parameter k, does G contain at least k vertex-disjoint induced 5-stars?

Sheet 5 – Color Coding

5-Stars:



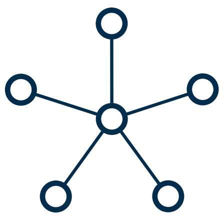
Given a graph G and parameter k, does G contain at least k vertex-disjoint induced 5-stars?



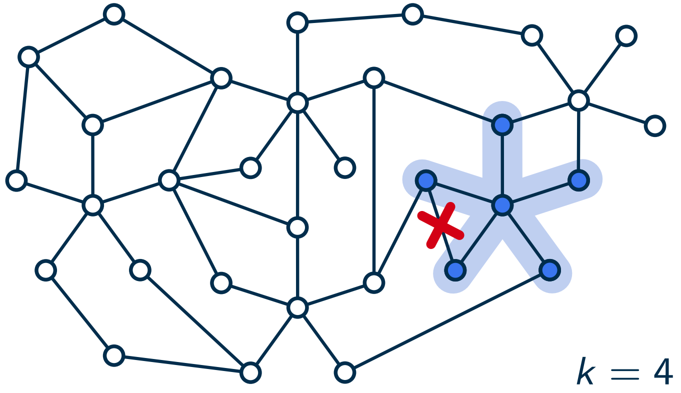
$k = 4$

Sheet 5 – Color Coding

5-Stars:

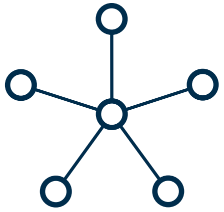


Given a graph G and parameter k, does G contain at least k vertex-disjoint induced 5-stars?

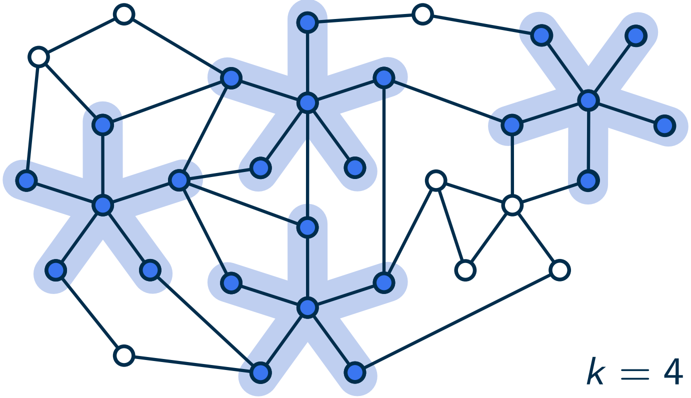


Sheet 5 – Color Coding

5-Stars:

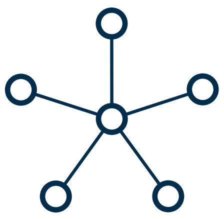


Given a graph G and parameter k, does G contain at least k vertex-disjoint induced 5-stars?

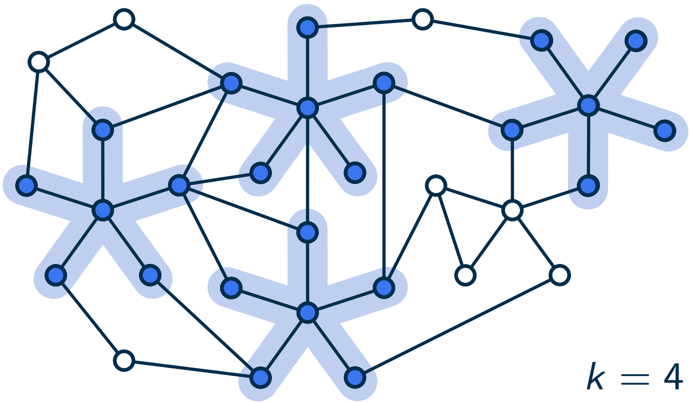


Sheet 5 – Color Coding

5-Stars:



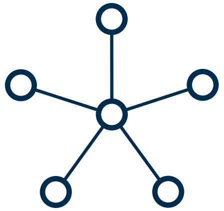
Given a graph G and parameter k, does G contain at least k vertex-disjoint induced 5-stars?



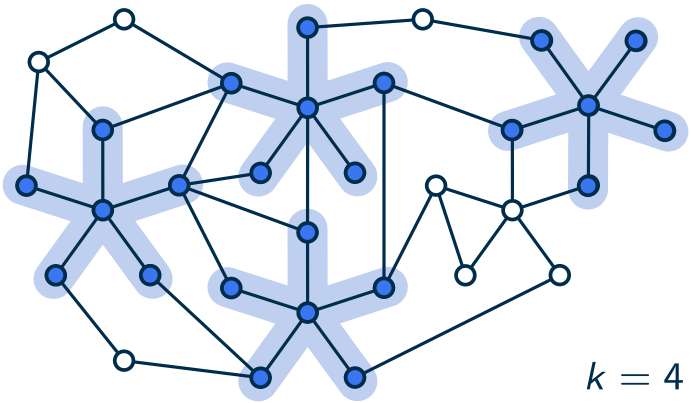
How does a colorful version of the problem look like?

Sheet 5 – Color Coding

5-Stars:



Given a graph G and parameter k, does G contain at least k vertex-disjoint induced 5-stars?

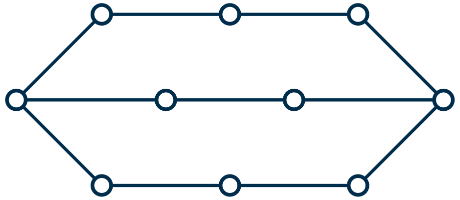


$k = 4$

How does a colorful version of the problem look like?

Longest Cycle:

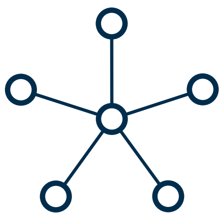
Given a graph G and parameter k, does G contain a cycle of length at least k?



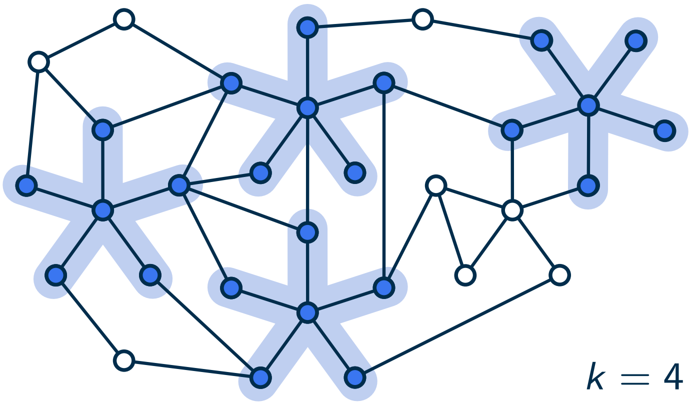
$k = 5$

Sheet 5 – Color Coding

5-Stars:



Given a graph G and parameter k, does G contain at least k vertex-disjoint induced 5-stars?

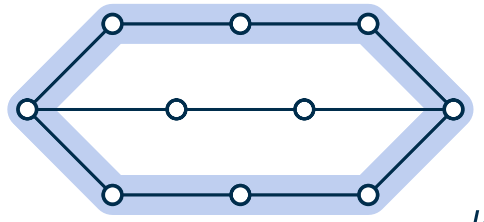


$k = 4$

How does a colorful version of the problem look like?

Longest Cycle:

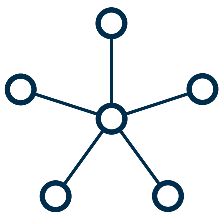
Given a graph G and parameter k,
does G contain a cycle of length <u>at least</u> k?



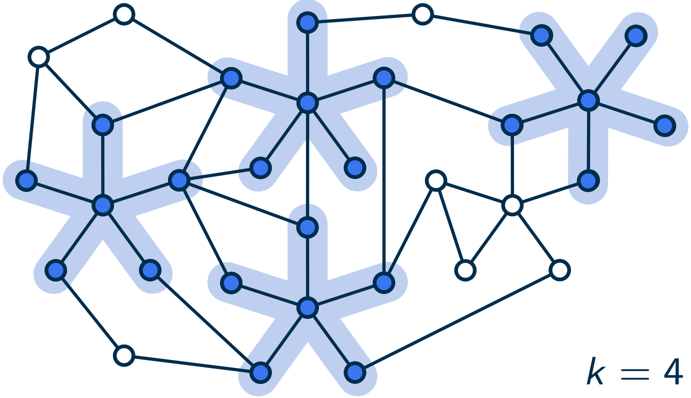
$k = 5$

Sheet 5 – Color Coding

5-Stars:



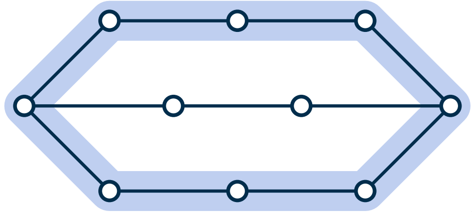
Given a graph G and parameter k, does G contain at least k vertex-disjoint induced 5-stars?



How does a colorful version of the problem look like?

Longest Cycle:

Given a graph G and parameter k, does G contain a cycle of length <u>at least</u> k?



How can we solve the problem for *exactly* k?

Hint 1: Xvh froru frglqj dojr vlplodu wr wkh rqh iru orqjhvw sdwk wr vroyh hAdfw sureohp

Hint 2: Wklqn derxw hgjh frqwudfwlrq (krz grhv wklv fkdqjh wkh ohqjwk ri fBfohv?)

Sheet 5 - Lower Bounds for Kernels

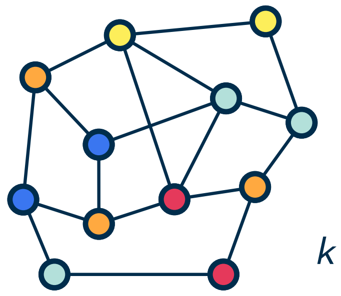
Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless $\mathrm{NP} \subseteq \mathrm{coNP/\,poly}$

Sheet 5 - Lower Bounds for Kernels

Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless $\mathrm{NP} \subseteq \mathrm{coNP/poly}$

Colorful Graph Motif:

Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?



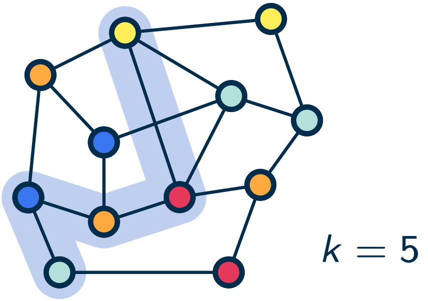
$k = 5$

Sheet 5 - Lower Bounds for Kernels

Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless $\mathrm{NP} \subseteq \mathrm{coNP/poly}$

Colorful Graph Motif:

Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

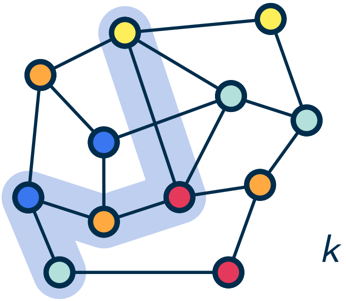


Sheet 5 - Lower Bounds for Kernels

Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless NP $\subseteq$ coNP/ poly

Colorful Graph Motif:

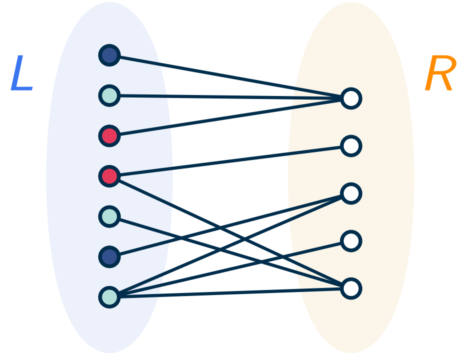
Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?



$k = 5$

Colorful Left-Right Dominating Set:

Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors.



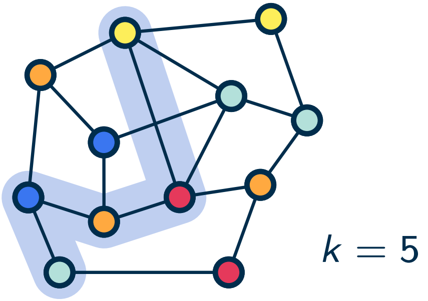
$\ell = 3$

Sheet 5 - Lower Bounds for Kernels

Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless $\mathrm{NP} \subseteq \mathrm{coNP/poly}$

Colorful Graph Motif:

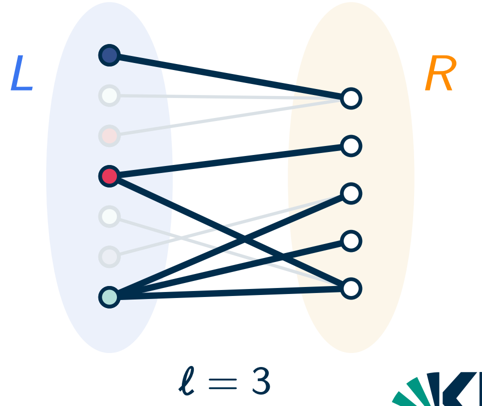
Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?



Colorful Left-Right Dominating Set: (parameter: $k = |R| + \ell$)

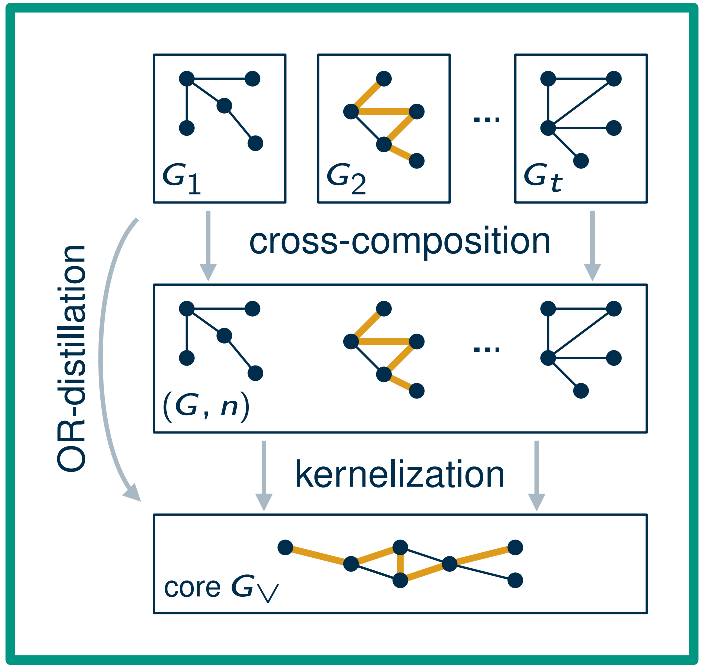
Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

this is basically SET COVER



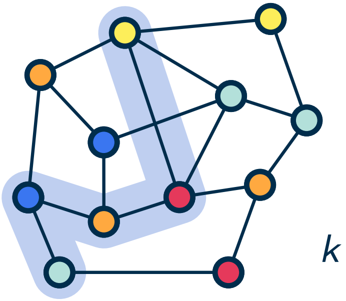
Sheet 5 - Lower Bounds for Kernels

Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless NP $\subseteq$ coNP/ poly



Colorful Graph Motif:

Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?



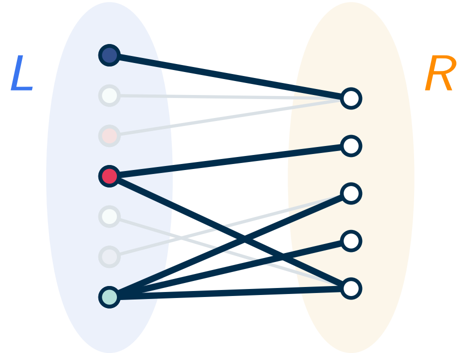
$k = 5$

Theorem (Reminder): If there is an OR-distillation of L into R, then $L \in$ coNP/poly

Colorful Left-Right Dominating Set: (parameter: $k = |R| + \ell$)

Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

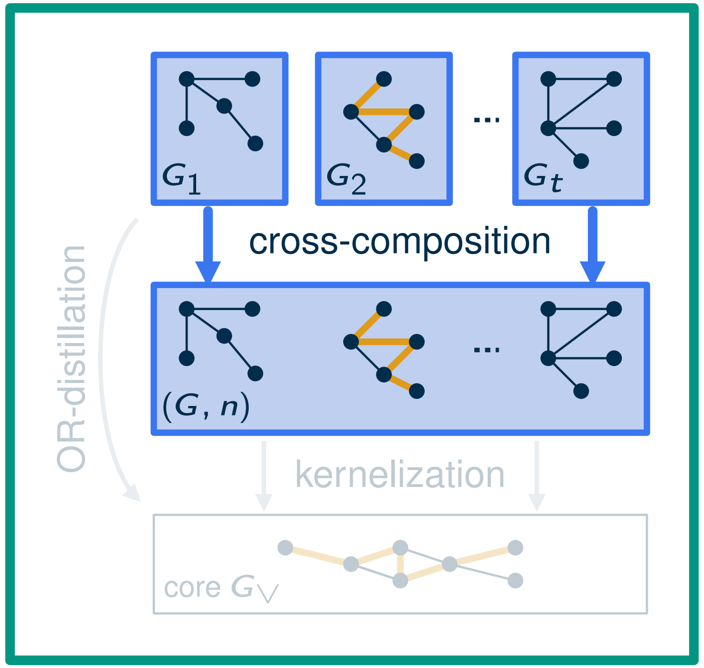
this is basically SET COVER



$\ell = 3$

Sheet 5 - Lower Bounds for Kernels

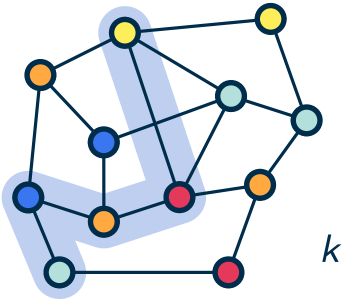
Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless NP $\subseteq$ coNP/ poly



Theorem (Reminder): If there is an OR-distillation of L into R, then $L \in$ coNP/poly

Colorful Graph Motif:

Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

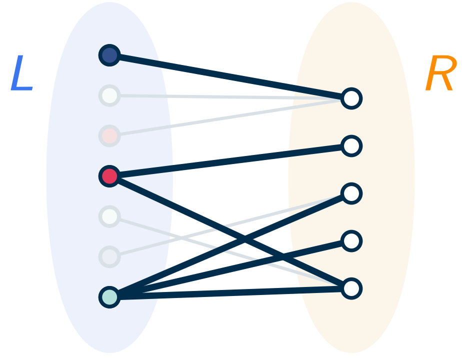


$k = 5$

Colorful Left-Right Dominating Set: (parameter: $k = |R| + \ell$)

Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors.
Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

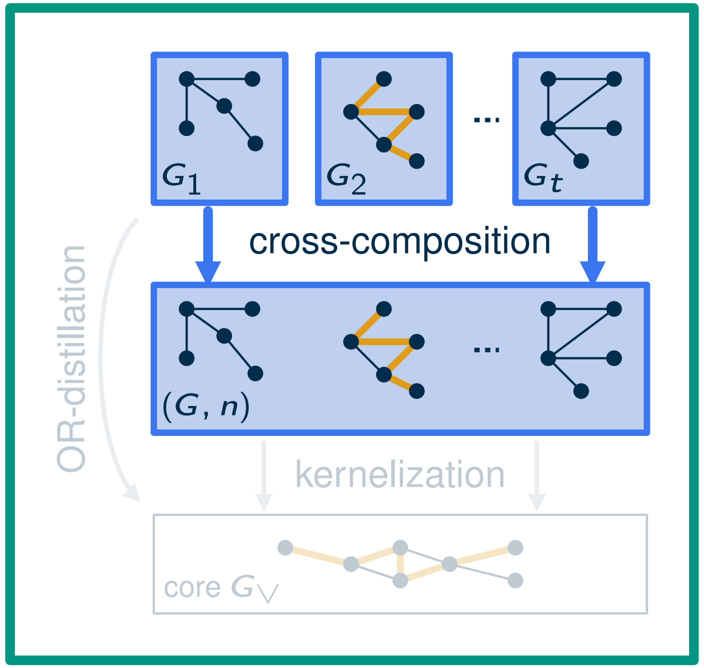
this is basically SET COVER



$\ell = 3$

Sheet 5 - Lower Bounds for Kernels

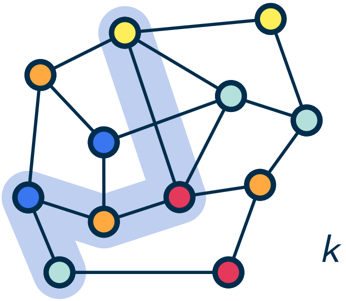
Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless NP $\subseteq$ coNP/ poly



Theorem (Reminder): If there is an OR-distillation of L into R, then $L \in$ coNP/poly

Colorful Graph Motif:

Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

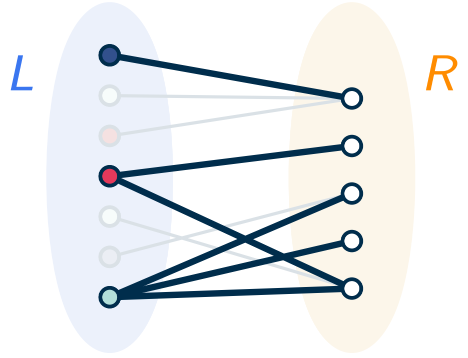


$k = 5$

Colorful Left-Right Dominating Set: (parameter: $k = |R| + \ell$)

Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

this is basically SET COVER

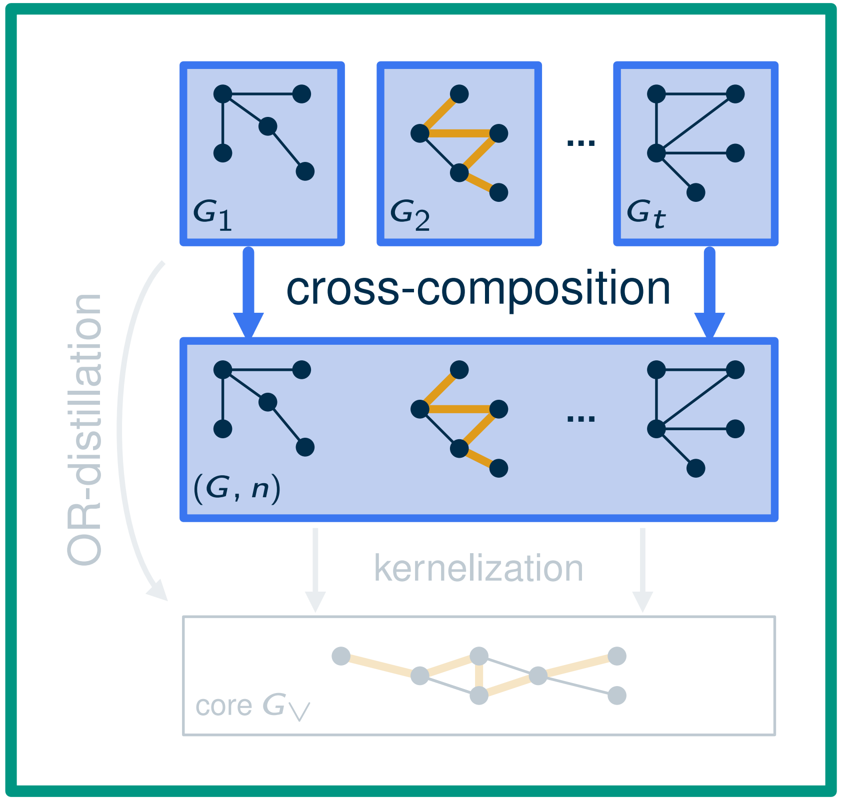


$\ell = 3$

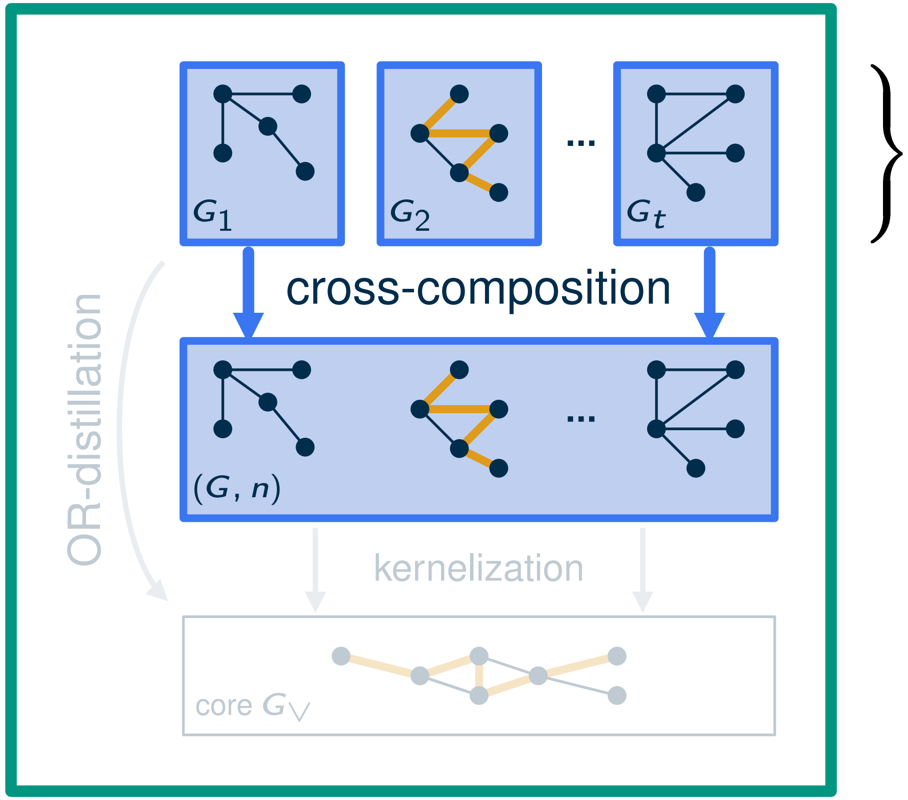
Hint 1: gr qrw gxsolfdwh wkh uljkw vlghv ri wkh elsduwlwlrqv.

Hint 2: Wkh qxpehu ri yhuwlfhv lq Brxu lqvwdqfh vhohfwru vkrxog ghshqg erwk rq wkh qxpehu ri gljlwv lq wkh elqduB uhsuhvhqwdwlrq ri w dqg wkh qxpehu ri froruv.

Recap: Cross Composition

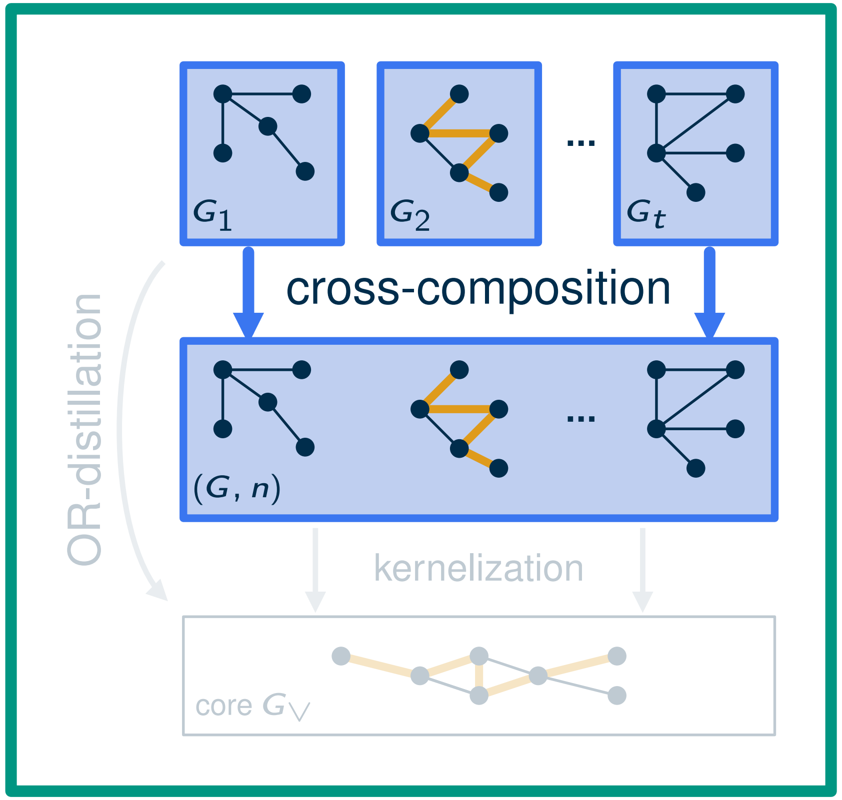


Recap: Cross Composition



t many "similar" instances x_1, x_2, . . ., x_t of problem P

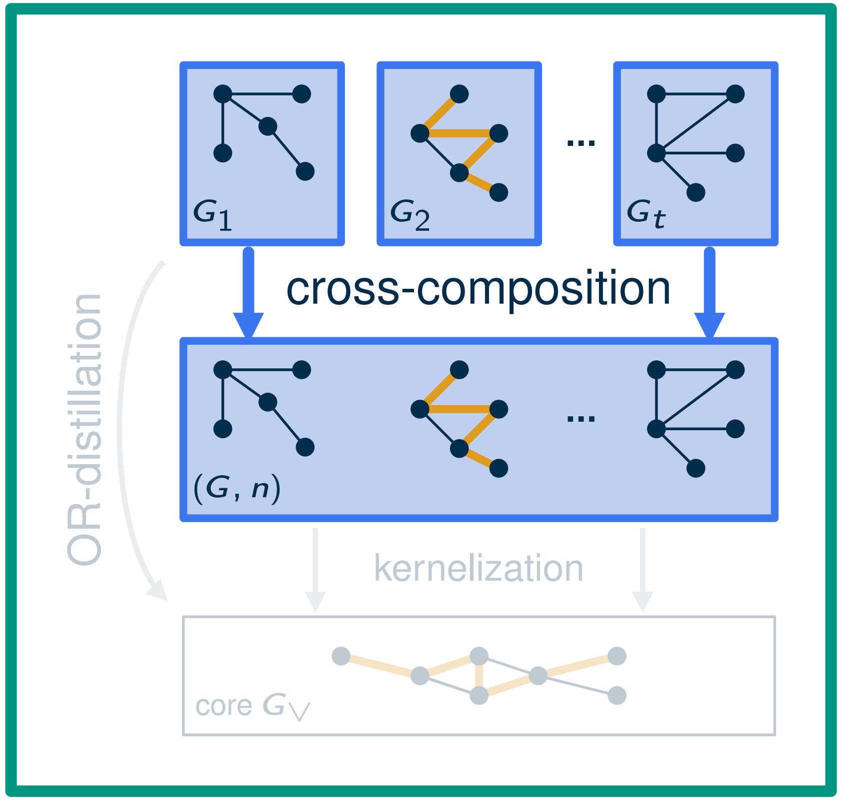
Recap: Cross Composition



t many "similar" instances x_1, x_2, ..., x_t of problem P

does not have to be parameterized

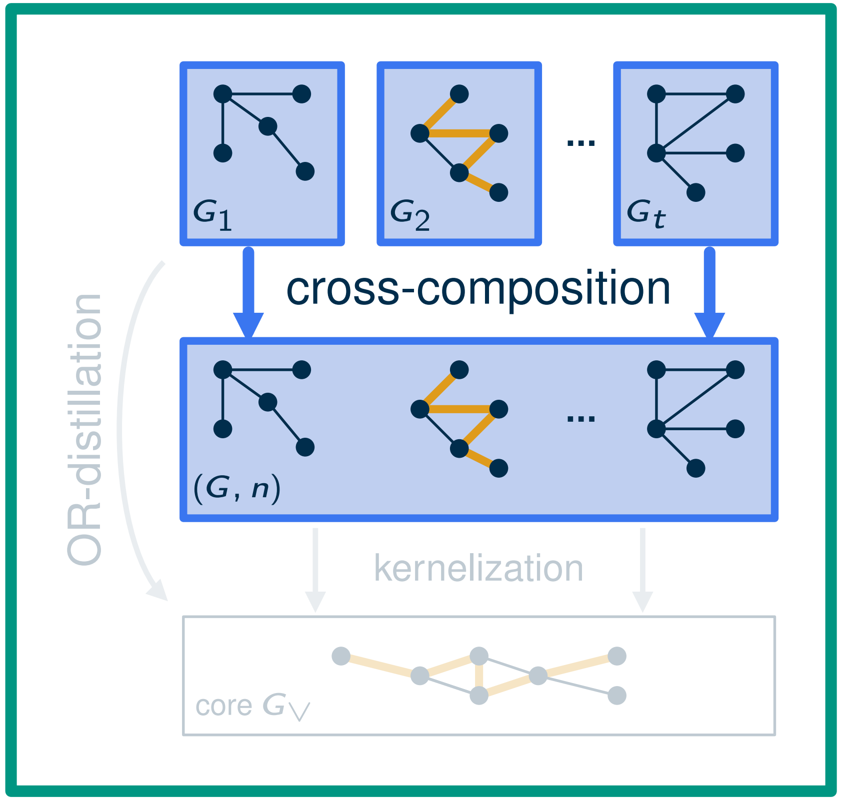
Recap: Cross Composition



t many "similar" instances x_1, x_2, ..., x_t of problem P

does not have to be parameterized

Recap: Cross Composition



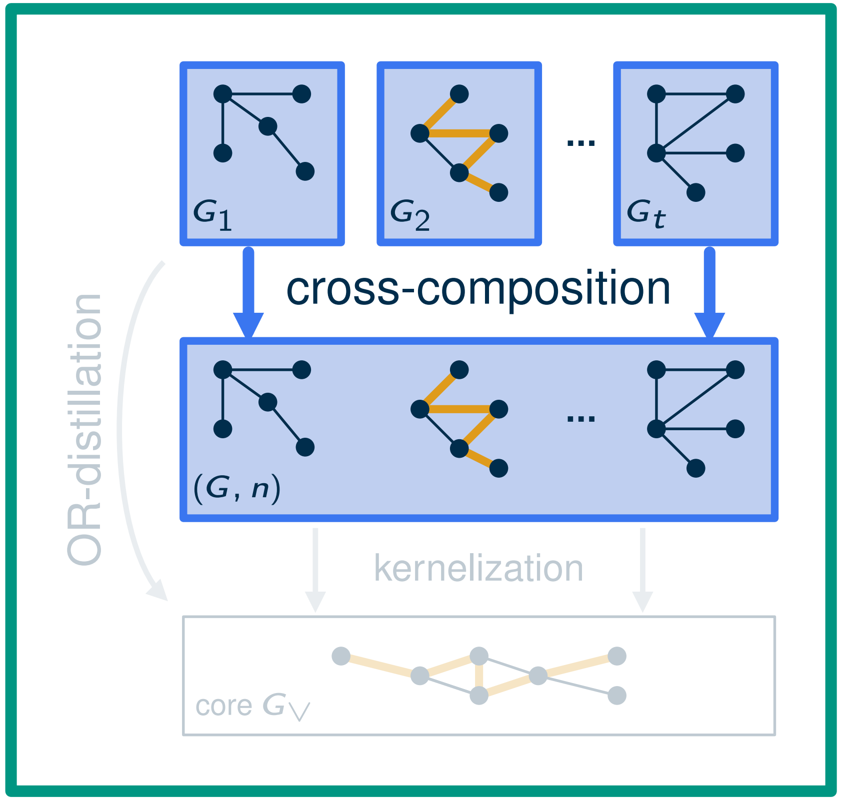
equivalent under some existing equivalence relation

- only polynomially many equivalence classes
- equivalence can be checked in polynomial time

does not have to be parameterized

t many "similar" instances x_1, x_2, ..., x_t of problem P

Recap: Cross Composition



often useful: same "size"

number of vertices, colors, letters in alphabet, . . .

equivalence relation

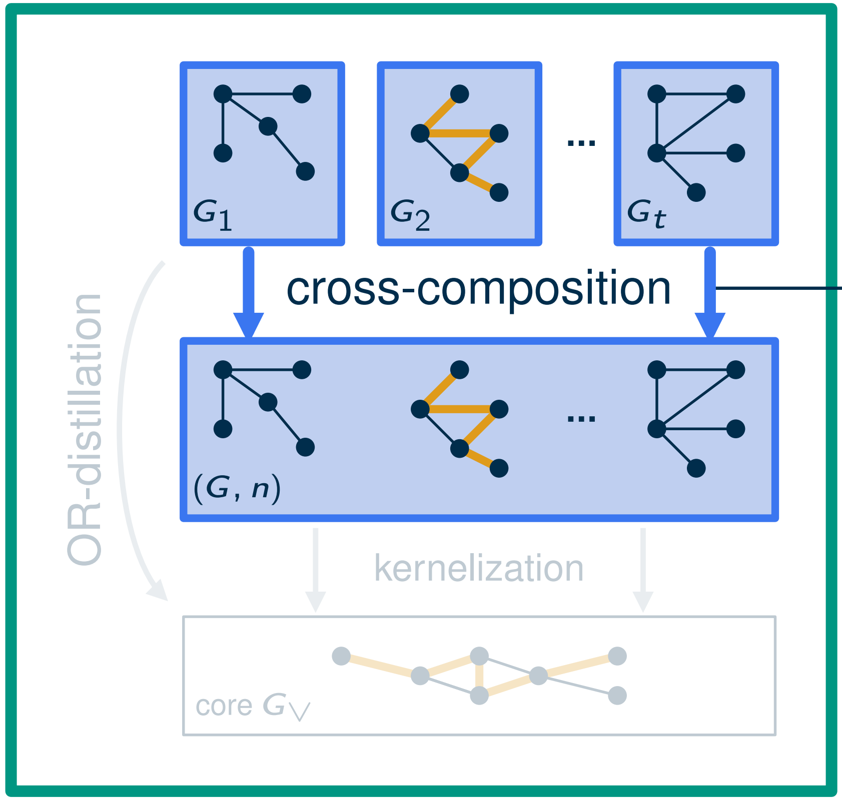
ivalence classes

d in polynomial time

does not have to be parameterized

t many "similar" instances x_1 , x_2 , . . ., x_t of problem P

Recap: Cross Composition



often useful: same "size"

number of vertices, colors, letters in alphabet, . . .

… equivalence relation

… ivalence classes

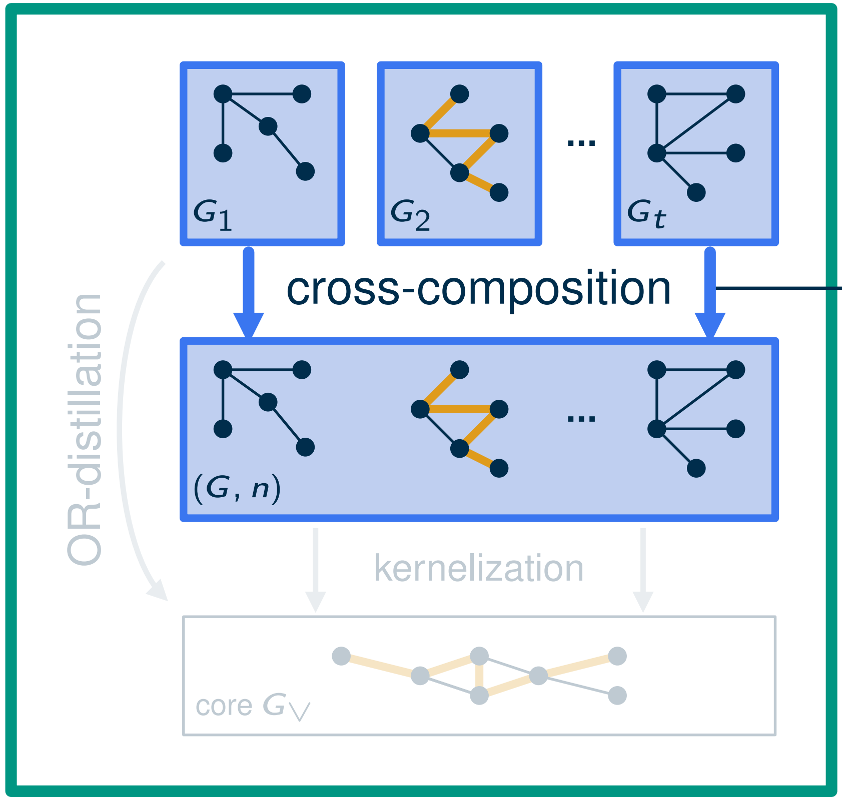
… d in polynomial time

does not have to be parameterized

t many "similar" instances x_1, x_2, . . ., x_t of problem P

runtime polynomial in $\sum_{i=1}^{t} |x_i|$

Recap: Cross Composition



often useful: same “size”

number of vertices, colors, letters in alphabet, . . .

equivalence relation

ivalence classes

d in polynomial time

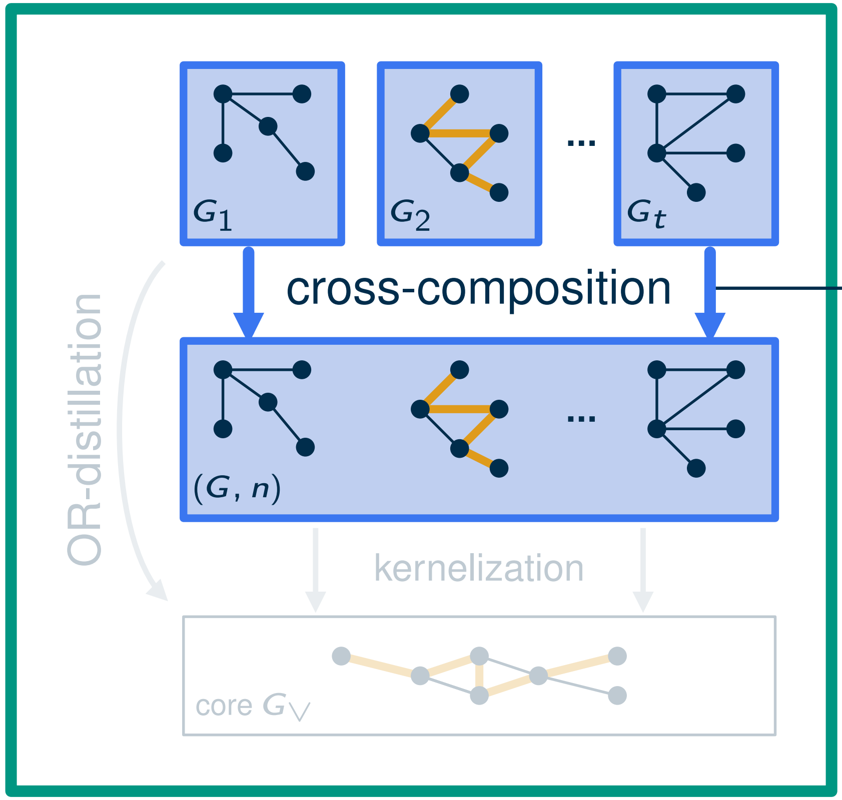
does not have to be parameterized

t many “similar” instances x_1, x_2, . . ., x_t of problem P

runtime polynomial in $\sum_{i=1}^{t} |x_i|$

one instance of parameterized problem Q such that:

Recap: Cross Composition



often useful: same "size"

number of vertices, colors, letters in alphabet, . . .

equivalence relation

ivalence classes

d in polynomial time

does not have to be parameterized

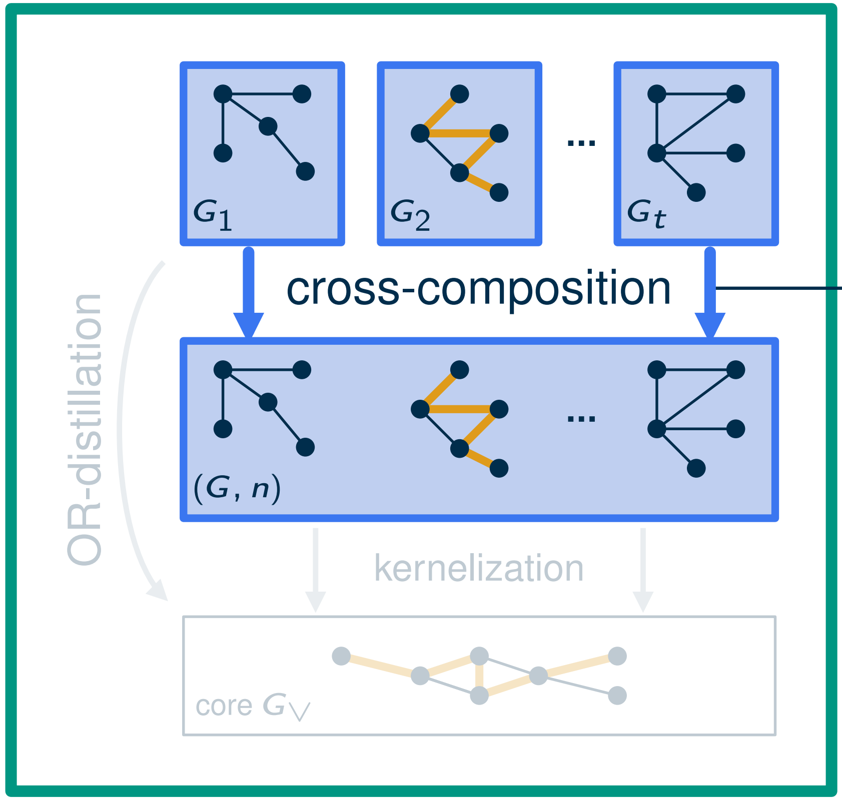
t many "similar" instances x_1 , x_2 , . . . , x_t of problem P

runtime polynomial in $\sum_{i=1}^{t} |x_i|$

one instance of parameterized problem Q such that:

- it has a solution if and only if at least one input instance x_i has a solution ← $\vee$

Recap: Cross Composition



often useful: same "size"

number of vertices, colors, letters in alphabet, . . .

equivalence relation

ivalence classes

d in polynomial time

does not have to be parameterized

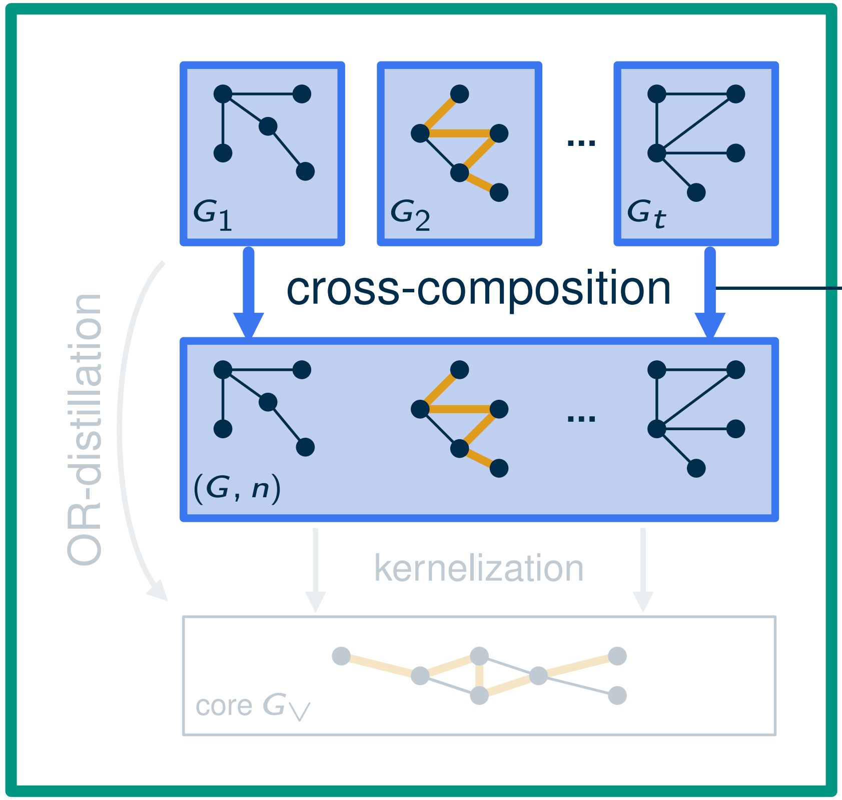
t many "similar" instances x_1, x_2, . . ., x_t of problem P

runtime polynomial in $\sum_{i=1}^{t} |x_i|$

one instance of parameterized problem Q such that:

- it has a solution if and only if at least one input instance x_i has a solution ← $\bigvee$
- parameter k is polynomial in $\max_{i \in [t]} |x_i| + \log t$

Recap: Cross Composition



often useful: same "size"
number of vertices, colors, letters in alphabet, . . .

…equivalence relation
…ivalence classes
…d in polynomial time

does not have to be parameterized

t many "similar" instances x_1, x_2, . . . , x_t of problem P

runtime polynomial in $\sum_{i=1}^{t} |x_i|$

one instance of parameterized problem Q such that:

- it has a solution if and only if at least one input instance x_i has a solution ← $\bigvee$
- parameter k is polynomial in $\max_{i \in [t]} |x_i| + \log t$

Do we always have to do this to show a lower bound for a kernel?

Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)

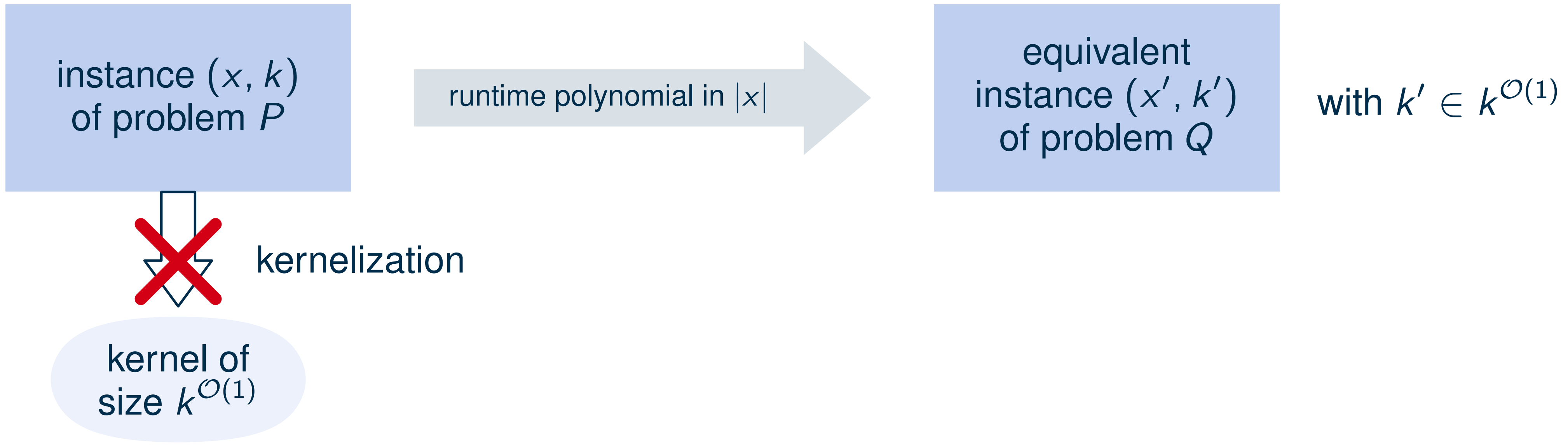
Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)

instance (x, k) of problem P → runtime polynomial in $|x|$ → equivalent instance (x', k') of problem Q with $k' \in k^{\mathcal{O}(1)}$

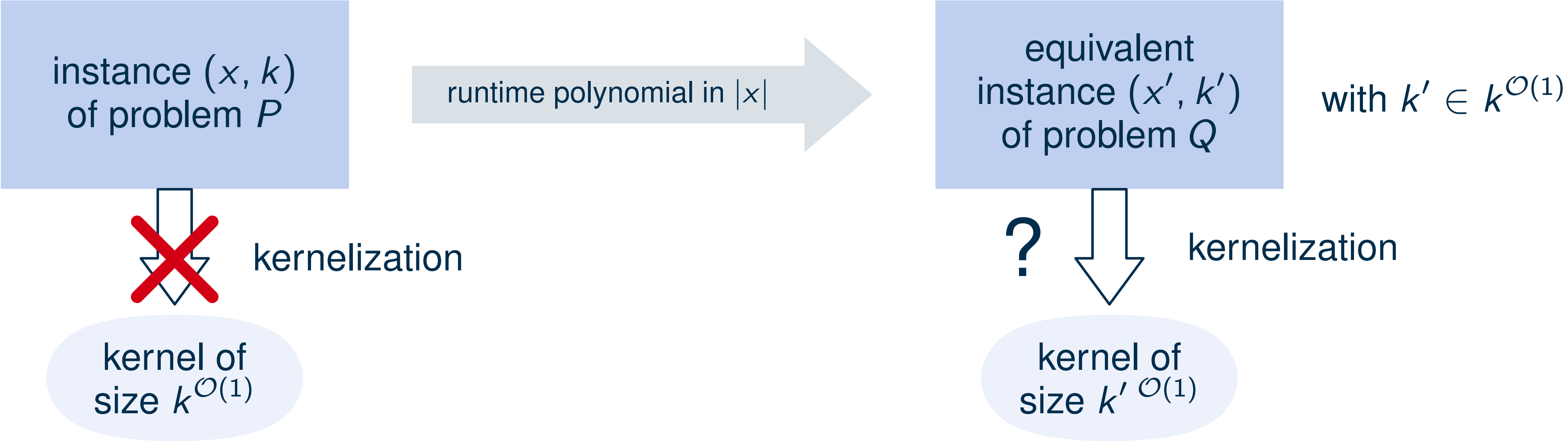
Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)



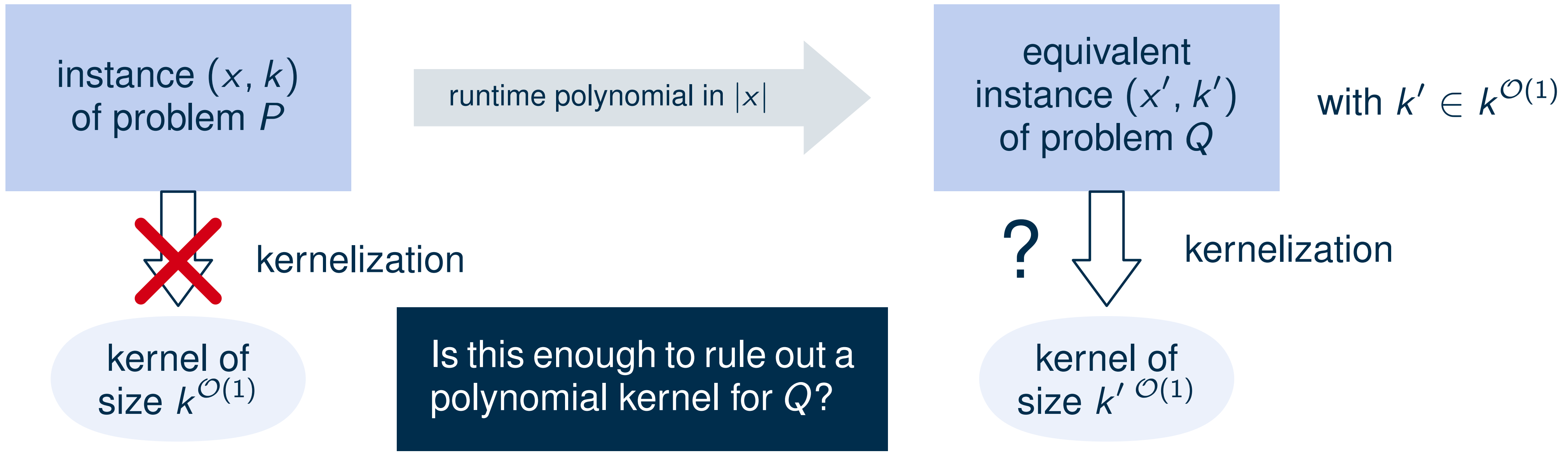
Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)



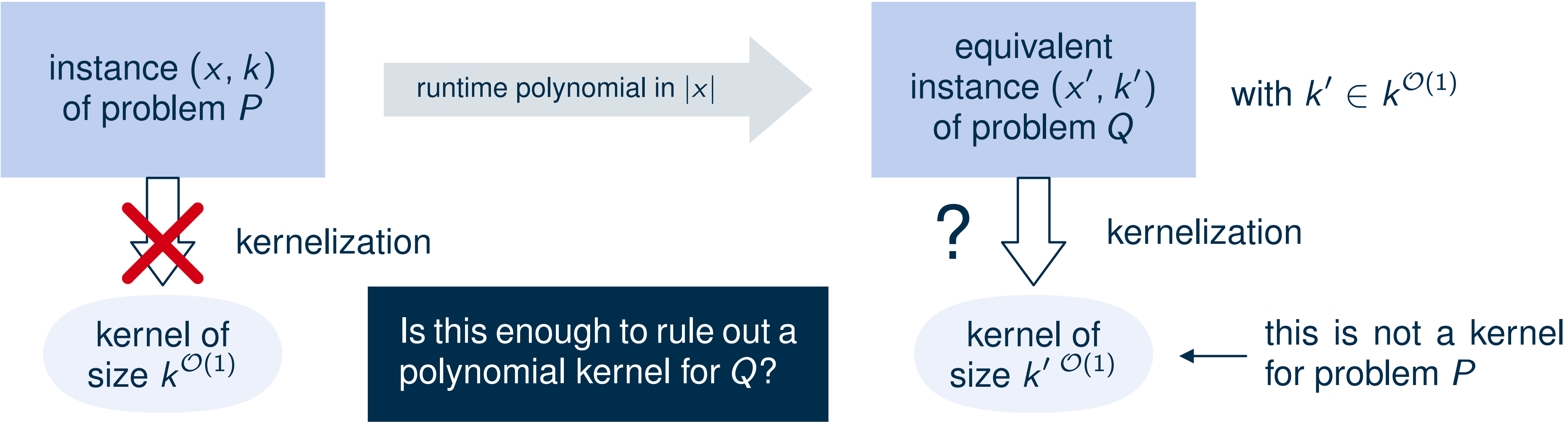
Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)



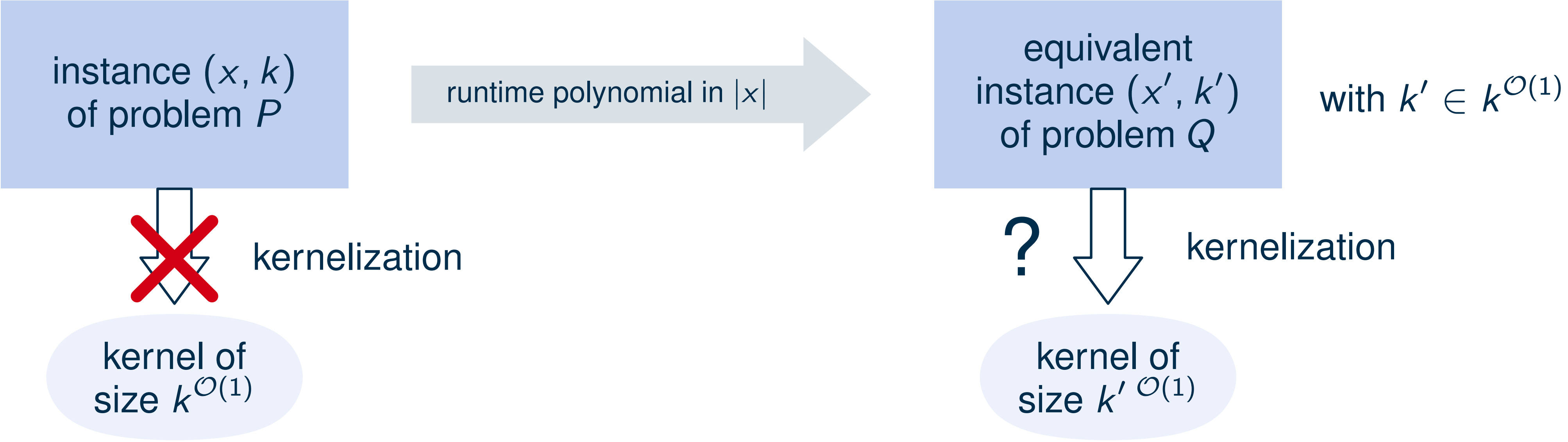
Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)



Polynomial Parameter Transformations (PPTs)

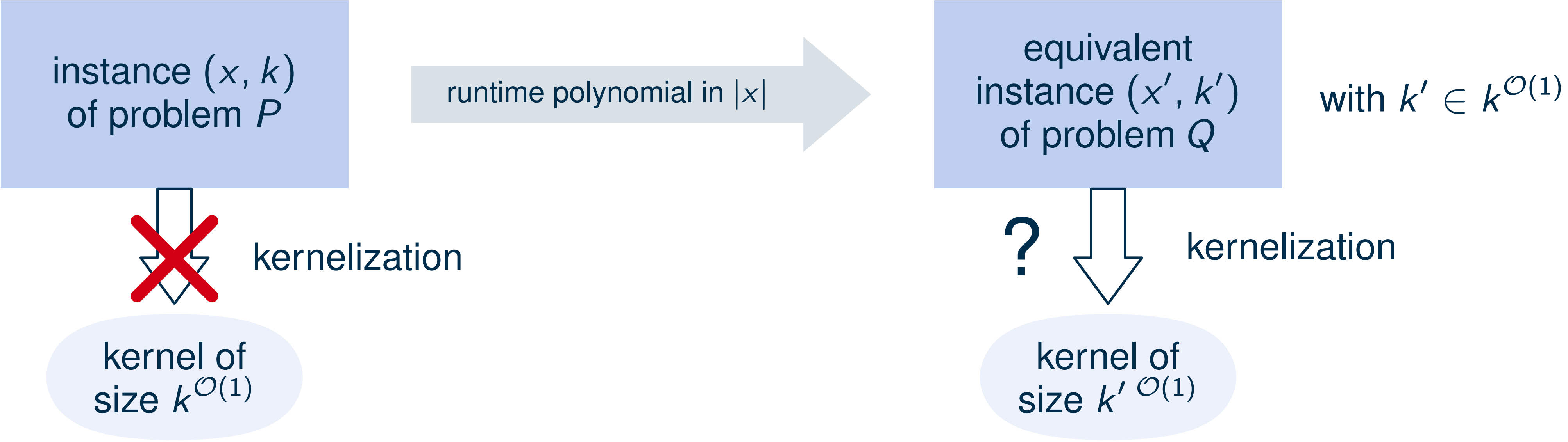
Idea: use problem from which we already know that it has no polynomial kernel (unless...)



For a parameterized problem L, a **kernelization** is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') (the **kernel**) such that $|x'| + k' \leq g(k)$.

Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)



polynomial compression

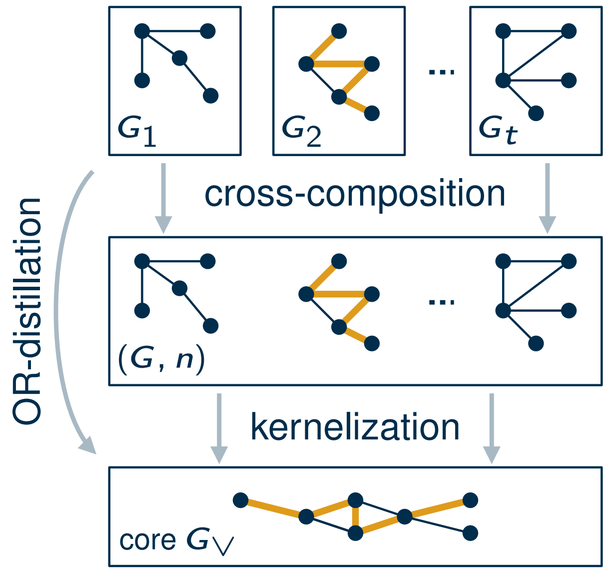
For a parameterized problem L, a ~~kernelization~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the kernel)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

Polynomial Parameter Transformations (PPTs)

Definition
Let L and R be languages. An **OR-distillation** of L into R is an algorithm that takes t similar inputs $x_1, \ldots, x_t$, runs in time $(\sum_{i\in[t]} |x_i|)^{O(1)}$, and produces an output y such that:
$|y| \in \max_{i\in[t]} |x_i|^{O(1)}$ and $y \in R \Leftrightarrow \bigvee_{i\in[t]} x_i \in L$.

Theorem
If there is an OR-distillation of L into R then $L \in \text{coNP}/\text{poly}$.



polynomial compression

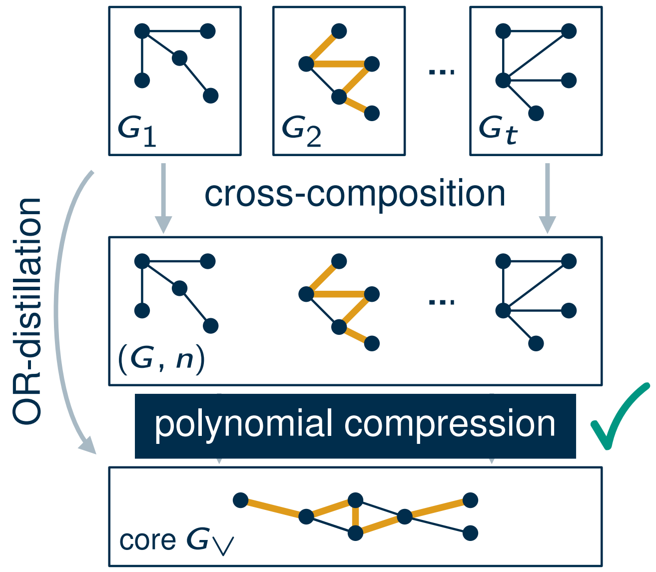
For a parameterized problem L, a ~~kernelization~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the kernel)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

Polynomial Parameter Transformations (PPTs)

Definition
Let L and R be languages. An **OR-distillation** of L into R is an algorithm that takes t similar inputs $x_1, \dots, x_t$, runs in time $(\sum_{i \in [t]} |x_i|)^{O(1)}$, and produces an output y such that:
$|y| \in \max_{i \in [t]} |x_i|^{O(1)}$ and $y \in R \Leftrightarrow \bigvee_{i \in [t]} x_i \in L$.

Theorem
If there is an OR-distillation of L into R then $L \in \mathrm{coNP/poly}$.



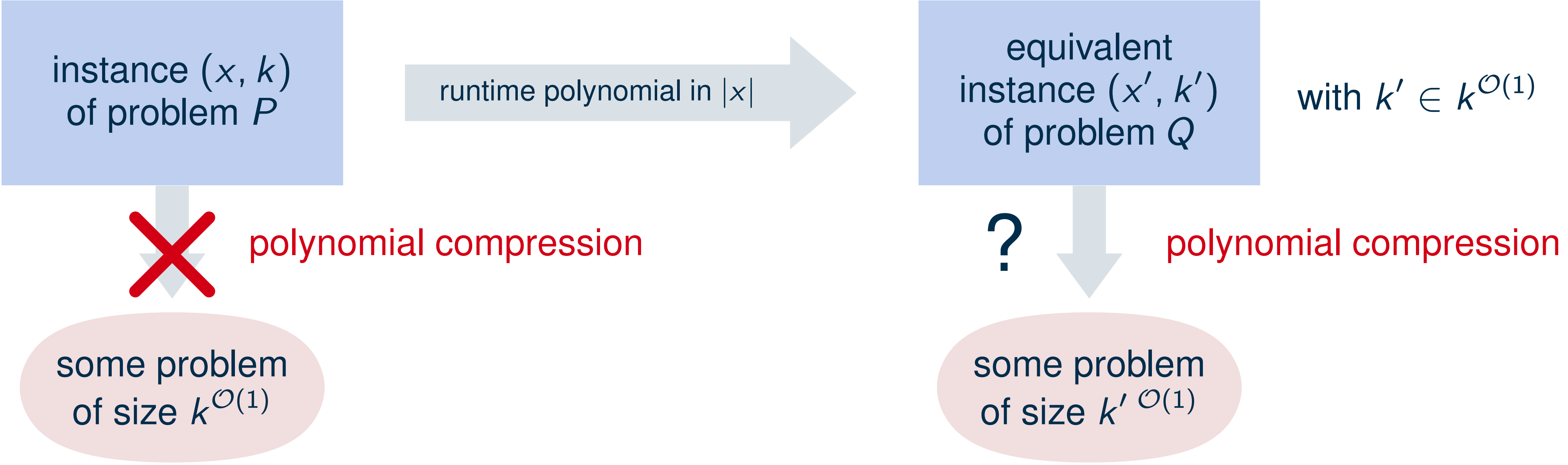
polynomial compression

For a parameterized problem L, a ~~kernelization~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the kernel)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)



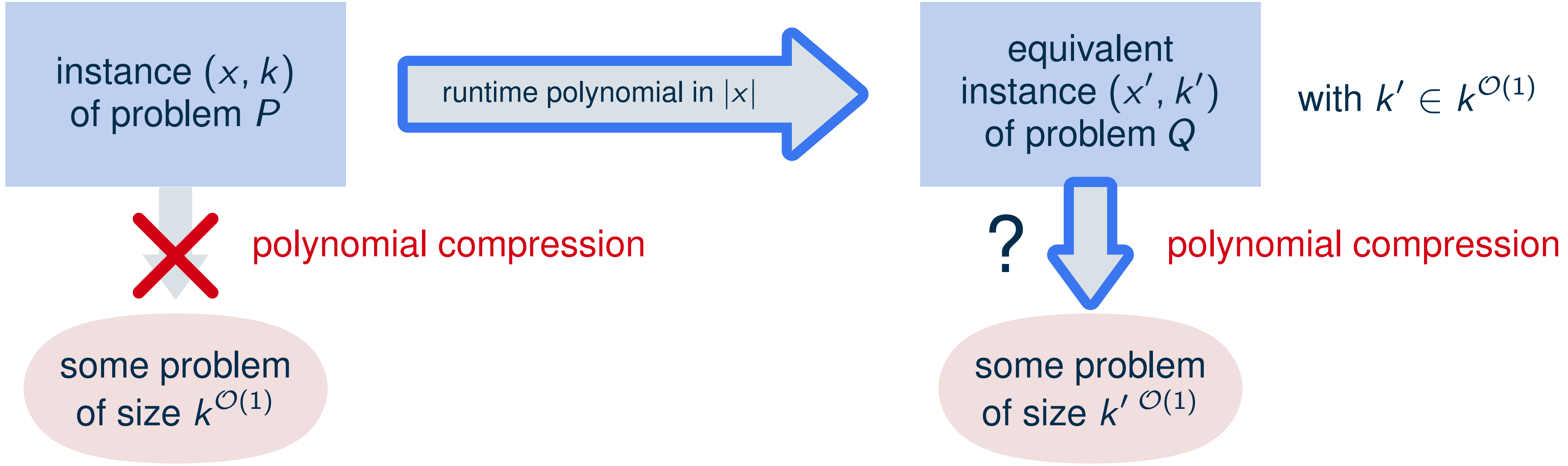
polynomial compression

For a parameterized problem L, a ~~kernelization~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the kernel)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)



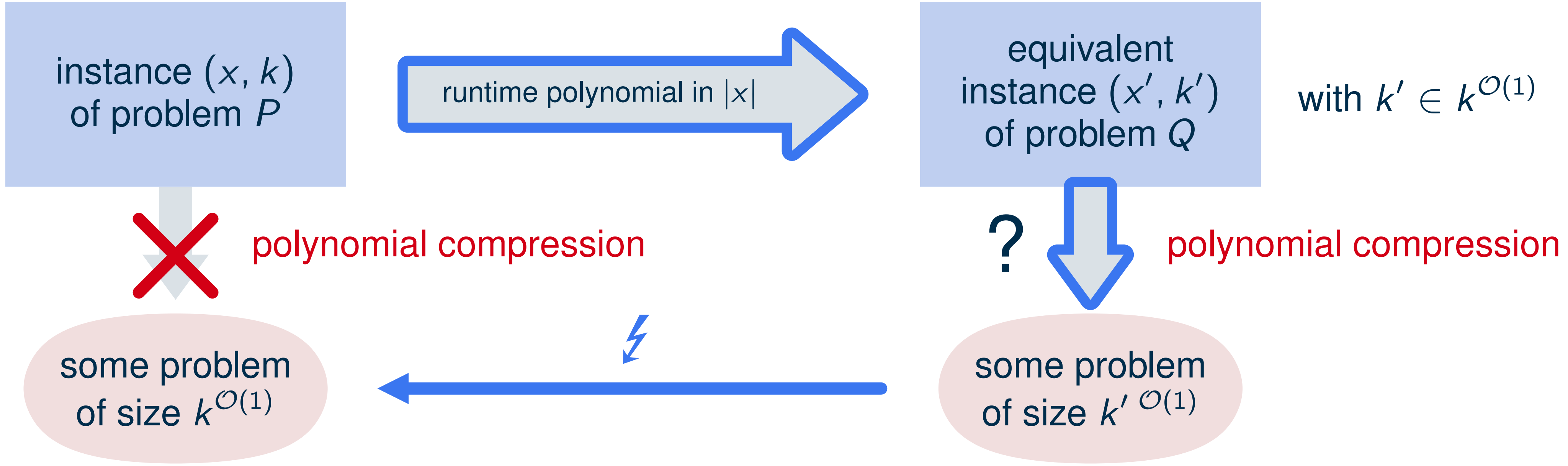
polynomial compression

For a parameterized problem L, a ~~kernelization~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the kernel)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)



polynomial compression

For a parameterized problem L, a ~~**kernelization**~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the **kernel**)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)

instance (x, k) of problem P — runtime polynomial in $|x|$ → equivalent instance (x', k') of problem Q with $k' \in k^{\mathcal{O}(1)}$

instance (x, k) of problem P ✗ polynomial compression → some problem of size $k^{\mathcal{O}(1)}$

equivalent instance (x', k') of problem Q ✗ polynomial compression → some problem of size $k'^{\,\mathcal{O}(1)}$

polynomial compression

For a parameterized problem L, a ~~**kernelization**~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the **kernel**)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)

instance (x, k) of problem P → runtime polynomial in $|x|$ → equivalent instance (x', k') of problem Q with $k' \in k^{\mathcal{O}(1)}$

Polynomial Parameter Transformation (PPT)

If problem P does not admit a polynomial compression and there is a PPT from P to Q, then Q does not admit a polynomial compression.

polynomial compression

For a parameterized problem L, a ~~**kernelization**~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the **kernel**)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

Polynomial Parameter Transformations (PPTs)

Idea: use problem from which we already know that it has no polynomial kernel (unless...)

instance (x, k) of problem P → runtime polynomial in $|x|$ → equivalent instance (x', k') of problem Q with $k' \in k^{\mathcal{O}(1)}$

Polynomial Parameter Transformation (PPT)

If problem P does not admit a polynomial compression and there is a PPT from P to Q, then Q does not admit a polynomial compression.

and if there is no polynomial compression, there is also no polynomial kernel!

polynomial compression

For a parameterized problem L, a ~~**kernelization**~~ is a polynomial-time algorithm that transforms any instance (x, k) into an equivalent instance (x', k') ~~(the **kernel**)~~ such that $|x'| + k' \leq g(k)$.

of a parameterized problem L_2

PPT: Colorful Graph Motif $\rightarrow$ Steiner Tree

instance (x, k)
of problem P

PPT
runtime polynomial in $|x|$

equivalent
instance (x', k')
of problem Q

with $k' \in k^{\mathcal{O}(1)}$

PPT: Colorful Graph Motif $\rightarrow$ Steiner Tree

instance (x, k) of problem P	**PPT** runtime polynomial in $\|x\|$ $\rightarrow$	equivalent instance (x', k') of problem Q	with $k' \in k^{\mathcal{O}(1)}$

STEINER TREE:
Given a graph G with special vertices $T \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?

PPT: Colorful Graph Motif $\rightarrow$ Steiner Tree

instance (x, k) of problem P

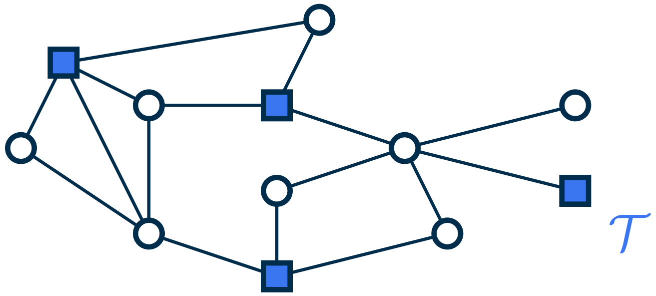
PPT

runtime polynomial in $|x|$

equivalent instance (x', k') of problem Q

with $k' \in k^{\mathcal{O}(1)}$

STEINER TREE:
Given a graph G with special vertices $\mathcal{T} \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?



$k = 6$

PPT: Colorful Graph Motif $\rightarrow$ Steiner Tree

instance (x, k) of problem P

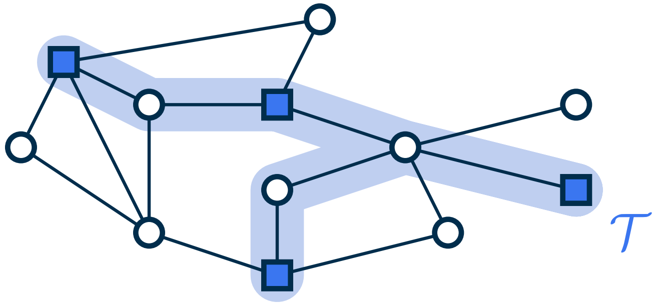
PPT

runtime polynomial in $|x|$

equivalent instance (x', k') of problem Q

with $k' \in k^{\mathcal{O}(1)}$

STEINER TREE:
Given a graph G with special vertices $\mathcal{T} \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?



PPT: Colorful Graph Motif $\to$ Steiner Tree

instance (x, k) of problem P

PPT

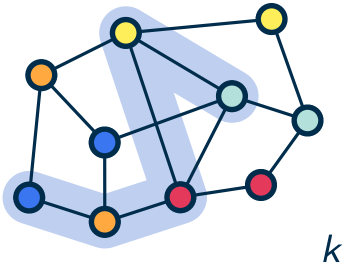
runtime polynomial in $|x|$

equivalent instance (x', k') of problem Q

with $k' \in k^{\mathcal{O}(1)}$

Colorful Graph Motif:
Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

Steiner Tree:
Given a graph G with special vertices $\mathcal{T} \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?



$k = 5$

PPT: Colorful Graph Motif $\rightarrow$ Steiner Tree

instance (x, k) of problem P

PPT

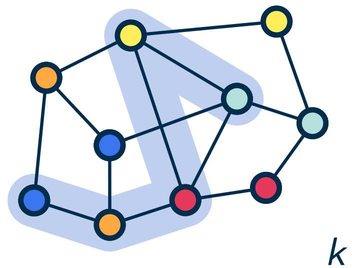
runtime polynomial in $|x|$

equivalent instance (x', k') of problem Q

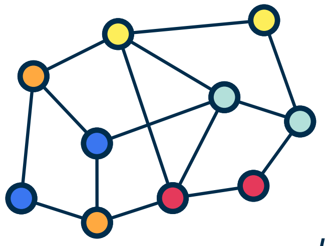
with $k' \in k^{\mathcal{O}(1)}$

Colorful Graph Motif:
Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

Steiner Tree:
Given a graph G with special vertices $\mathcal{T} \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?

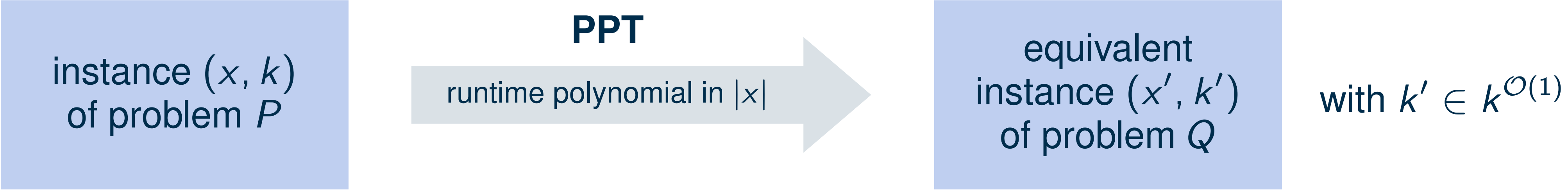


$k = 5$



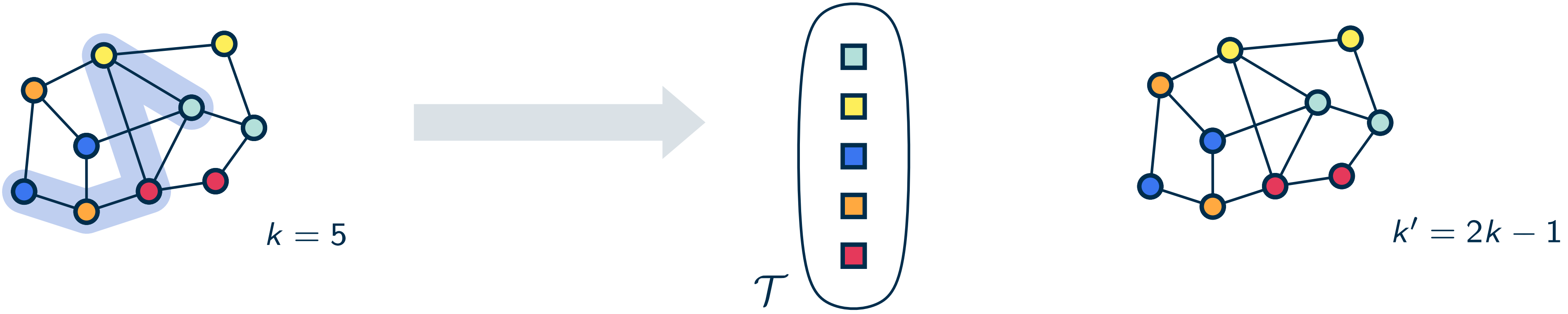
$k' = 2k - 1$

PPT: Colorful Graph Motif → Steiner Tree



COLORFUL GRAPH MOTIF:
Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

STEINER TREE:
Given a graph G with special vertices $\mathcal{T} \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?

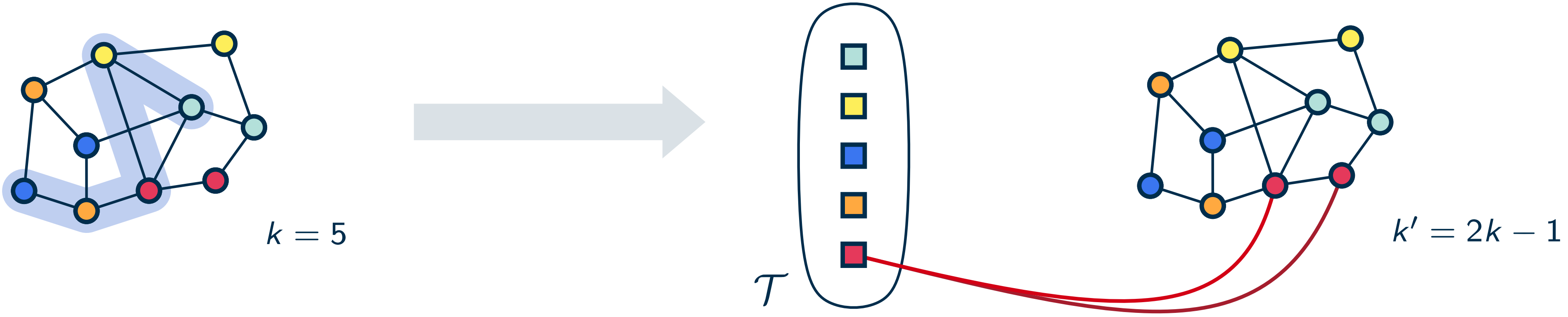


PPT: Colorful Graph Motif $\rightarrow$ Steiner Tree

instance (x, k) of problem P → **PPT** (runtime polynomial in $|x|$) → equivalent instance (x', k') of problem Q with $k' \in k^{\mathcal{O}(1)}$

COLORFUL GRAPH MOTIF:
Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

STEINER TREE:
Given a graph G with special vertices $\mathcal{T} \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?

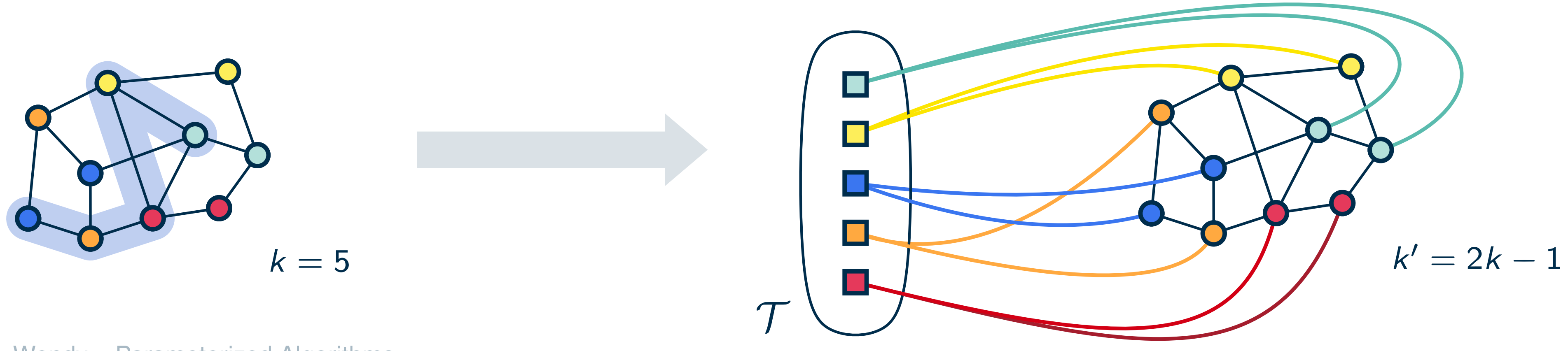


PPT: Colorful Graph Motif → Steiner Tree

instance (x, k) of problem P → **PPT** (runtime polynomial in $|x|$) → equivalent instance (x', k') of problem Q with $k' \in k^{\mathcal{O}(1)}$

Colorful Graph Motif:
Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

Steiner Tree:
Given a graph G with special vertices $\mathcal{T} \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?

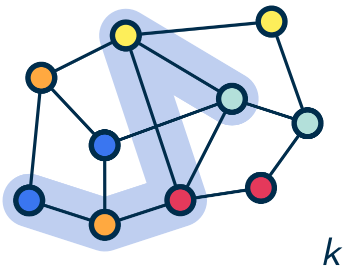


PPT: Colorful Graph Motif $\rightarrow$ Steiner Tree

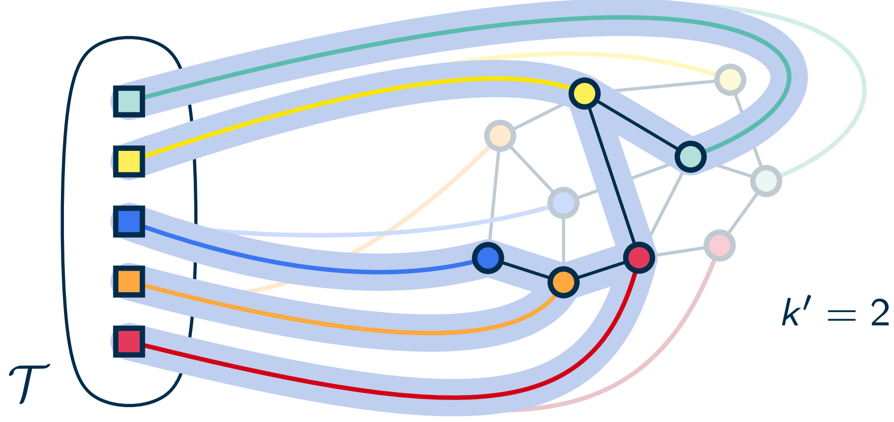
instance (x, k) of problem P → **PPT** (runtime polynomial in $|x|$) → equivalent instance (x', k') of problem Q with $k' \in k^{\mathcal{O}(1)}$

COLORFUL GRAPH MOTIF:
Given a graph G with a vertex coloring with k colors, does G have a colorful connected subgraph of size exactly k?

STEINER TREE:
Given a graph G with special vertices $\mathcal{T} \subseteq V(G)$ called the *terminals*, and an integer k, does G have a connected subgraph that contains all terminals and has at most k edges?



$k = 5$

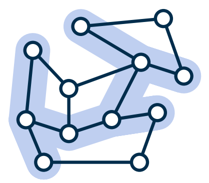


$k' = 2k - 1$

Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

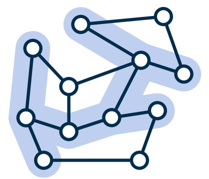
$\Sigma = \{a, b, c\}$

bcaccbbacbbab

Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

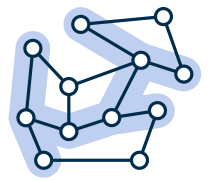
$\Sigma = \{a, b, c\}$

bcaccbbacbbab

Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

Max Leaf Subtree (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

$\Sigma = \{a, b, c\}$

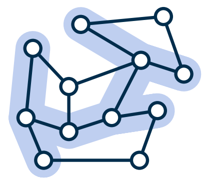
bcaccbbacbbab

Connected Vertex Cover (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

Max Leaf Subtree (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

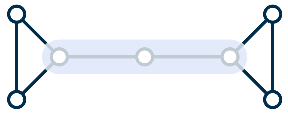
Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \dots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

$\Sigma = \{a, b, c\}$

bcaccbbacbbab

Connected Vertex Cover (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

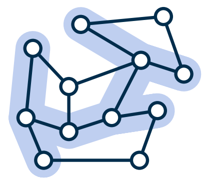
Connected Feedback Vertex Set (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

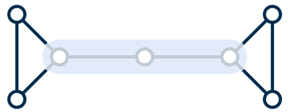
DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \dots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

$\Sigma = \{a, b, c\}$

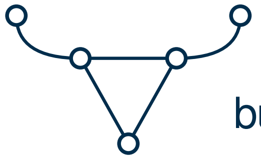
bcaccbbacbbab

CONNECTED VERTEX COVER (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



CONNECTED BULL HITTING (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?

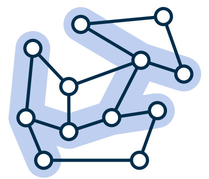


bull

Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

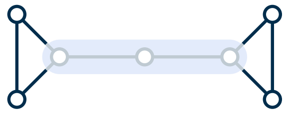
DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

$\Sigma = \{a, b, c\}$

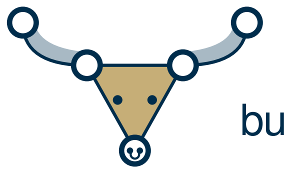
bcaccbbacbbab

CONNECTED VERTEX COVER (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



CONNECTED BULL HITTING (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?

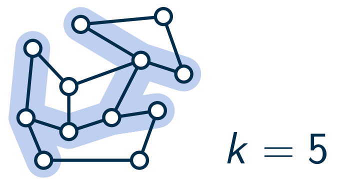


bull

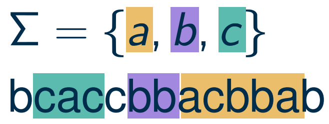
Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?

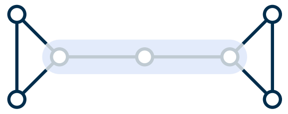


DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

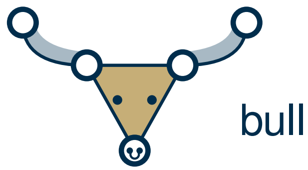


CONNECTED VERTEX COVER (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



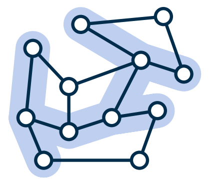
CONNECTED BULL HITTING (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

Max Leaf Subtree (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves? ⋆



$k = 5$

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \dots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ? ⋆⋆⋆

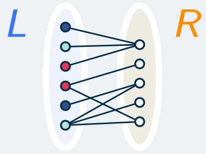
$\Sigma = \{a, b, c\}$

bcaccbbacbbab

Connected Vertex Cover (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G? ⋆⋆

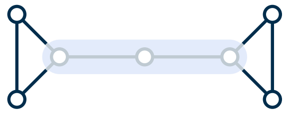
Colorful Left-Right DomSet

$k = |R| + \ell$

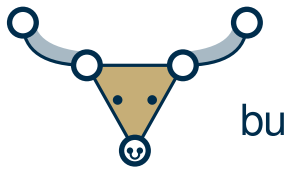


⋆⋆ ⋆⋆

Connected Feedback Vertex Set (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



Connected Bull Hitting (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



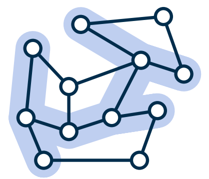
bull

Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

NP-hard

Max Leaf Subtree (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \dots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

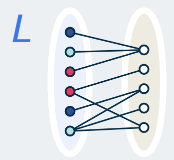
$\Sigma = \{a, b, c\}$

bcaccbbacbbab

no polynomial compression

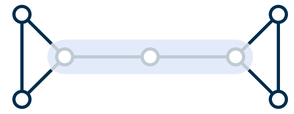
Colorful Left-Right DomSet

$k = |R| + \ell$

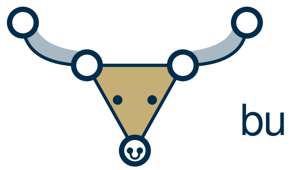


Connected Vertex Cover (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

Connected Feedback Vertex Set (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



Connected Bull Hitting (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



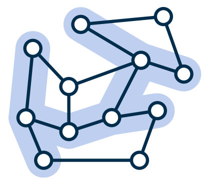
bull

Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

NP-hard

Max Leaf Subtree (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

introduce $\log t$ many new characters

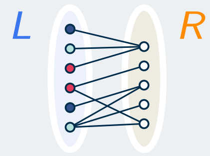
$\Sigma = \{a, b, c\}$

bcaccbbacbbab

no polynomial compression

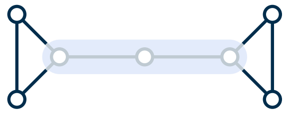
Colorful Left-Right DomSet

$k = |R| + \ell$

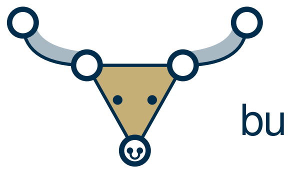


Connected Vertex Cover (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

Connected Feedback Vertex Set (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



Connected Bull Hitting (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



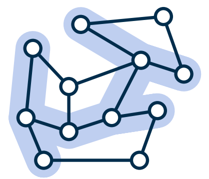
bull

Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

NP-hard

Max Leaf Subtree (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \dots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

introduce $\log t$ many new characters

for instance selector: think about binary search

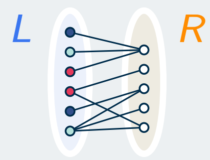
$\Sigma = \{a, b, c\}$

bcaccbbacbbab

no polynomial compression

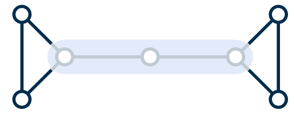
Colorful Left-Right DomSet

$k = |R| + \ell$

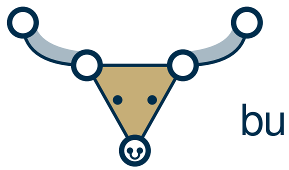


Connected Vertex Cover (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

Connected Feedback Vertex Set (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



Connected Bull Hitting (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



bull

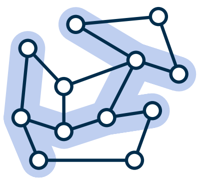
Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

NP-hard

MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?

⋆



$k = 5$

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \dots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

introduce $\log t$ many new characters

for instance selector: think about binary search

⋆⋆⋆

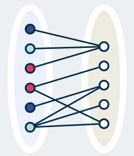
$\Sigma = \{a, b, c\}$

bcaccbbacbbab

no polynomial compression

COLORFUL LEFT-RIGHT DOMSET

$k = |R| + \ell$



⋆⋆

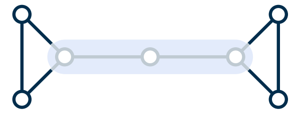
CONNECTED VERTEX COVER (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

start with same vertex set and add one for each color

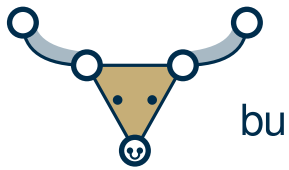
⋆⋆

CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?

⋆⋆



CONNECTED BULL HITTING (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



bull

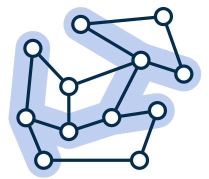
Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

NP-hard

Max Leaf Subtree (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?

★



$k = 5$

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

introduce $\log t$ many new characters

for instance selector: think about binary search

★★★

$\Sigma = \{a, b, c\}$

bcaccbbacbbab

no polynomial compression

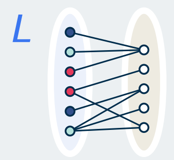
Connected Vertex Cover (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

start with same vertex set and add one for each color

★★

Colorful Left-Right DomSet

$k = |R| + \ell$

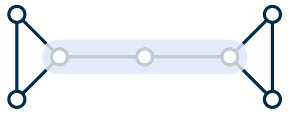


★★

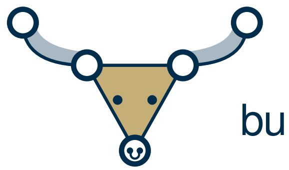
Connected Feedback Vertex Set (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?

how could edges be extended to cycles?

★★



Connected Bull Hitting (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



bull

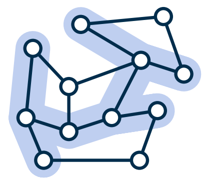
Cross Compositions and PPTs

Show that there are no polynomial kernels (unless...)

NP-hard

MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?

⋆



$k = 5$

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

introduce $\log t$ many new characters

for instance selector: think about binary search

⋆⋆⋆

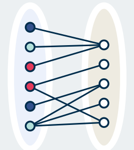
$\Sigma = \{a, b, c\}$

bcaccbbacbbab

no polynomial compression

COLORFUL LEFT-RIGHT DOMSET

$k = |R| + \ell$



⋆⋆

CONNECTED VERTEX COVER (k): Given a graph $G = (V, E)$ and an integer k, does G contain a vertex cover of size k that induces a connected subgraph of G?

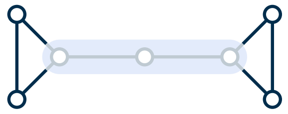
start with same vertex set and add one for each color

⋆⋆

CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?

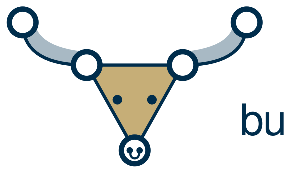
how could edges be extended to cycles?

⋆⋆



CONNECTED BULL HITTING (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?

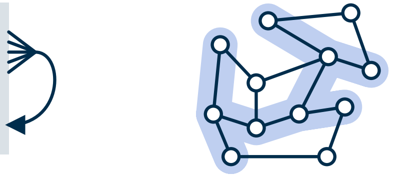
... to bulls?



bull

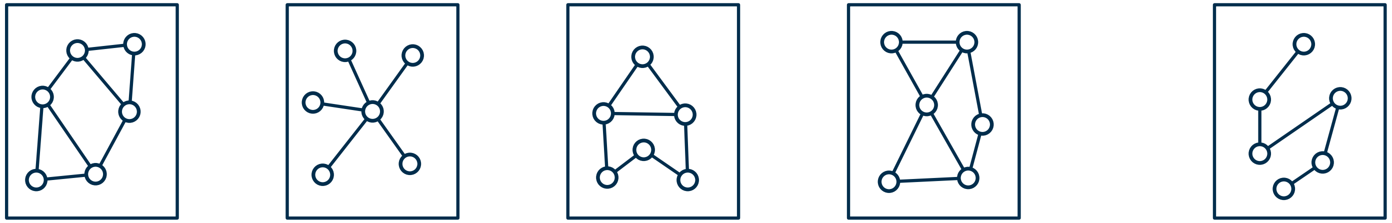
Max Leaf Subtree – Cross Composition

MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

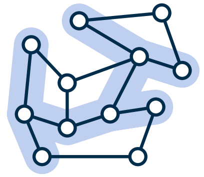
G_1 G_2 G_3 G_4 G_5



$k = 5$

Max Leaf Subtree – Cross Composition

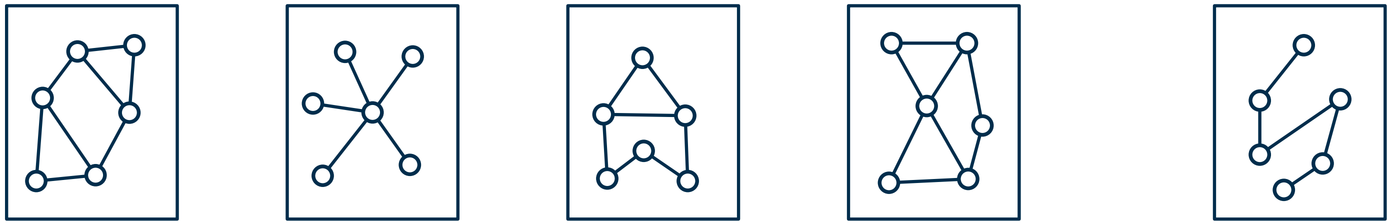
MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

"similar"?

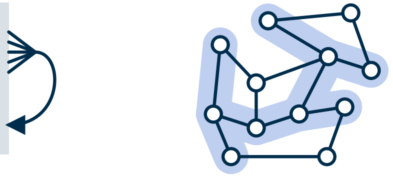
G_1 G_2 G_3 G_4 G_5



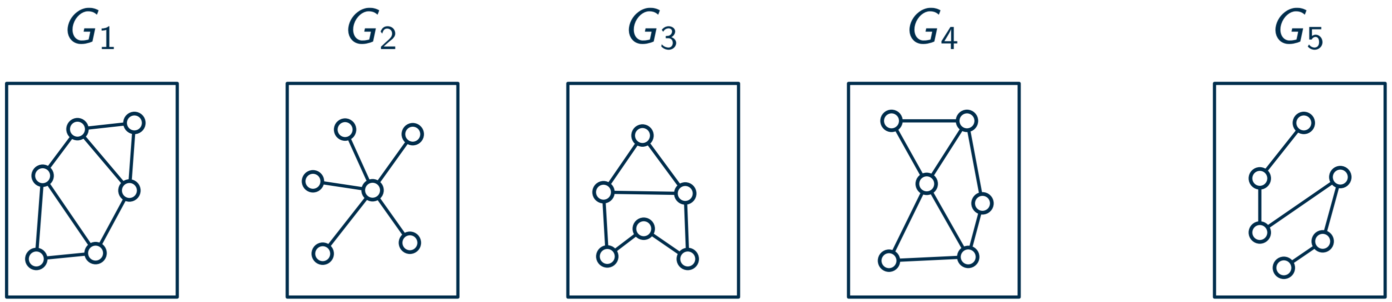
$k = 5$

Max Leaf Subtree – Cross Composition

Max Leaf Subtree (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$



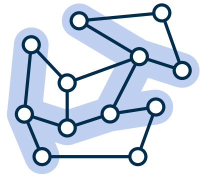
“similar”?

e.g. same $|V|$

$k = 5$

Max Leaf Subtree – Cross Composition

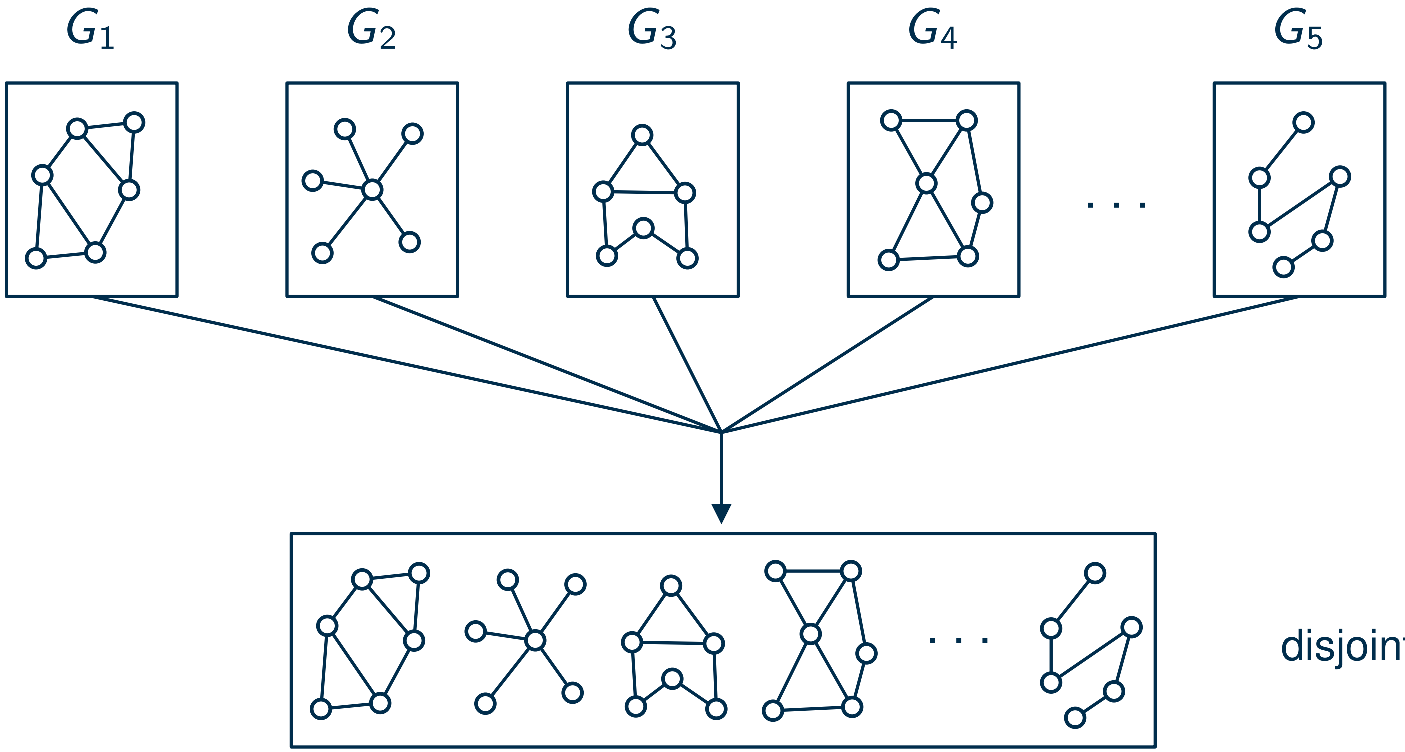
MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

"similar"?

e.g. same $|V|$

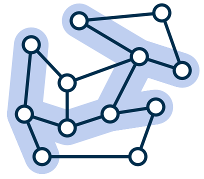


$k = 5$

disjoint union

Max Leaf Subtree – Cross Composition

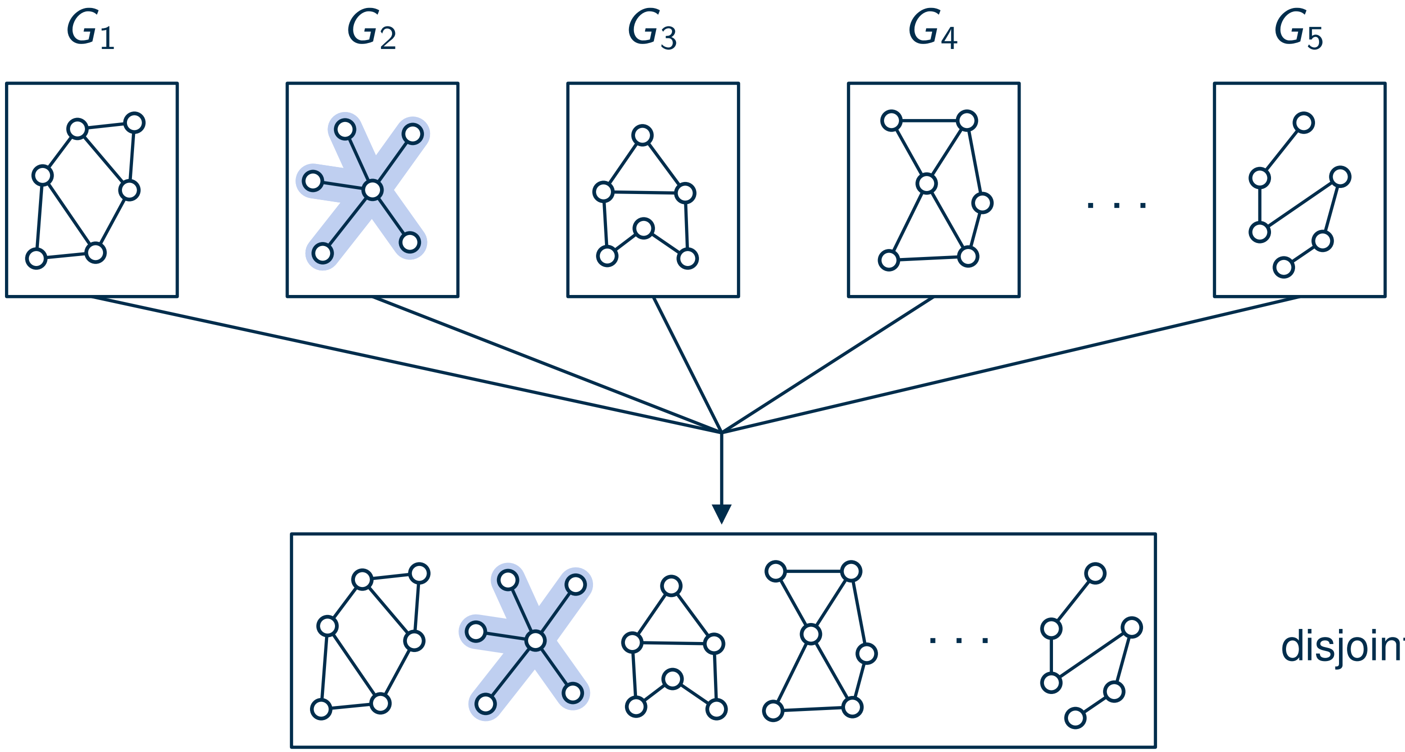
MAX LEAF SUBTREE (k): Given a graph $G = (V, E)$ and an integer k, does G contain a subtree that has at least k leaves?



$k = 5$

“similar”?

e.g. same $|V|$



$k = 5$

disjoint union

Disjoint Factors – Cross Composition

DISJOINT FACTORS (|Σ|): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

$\Sigma = \{a, b, c\}$

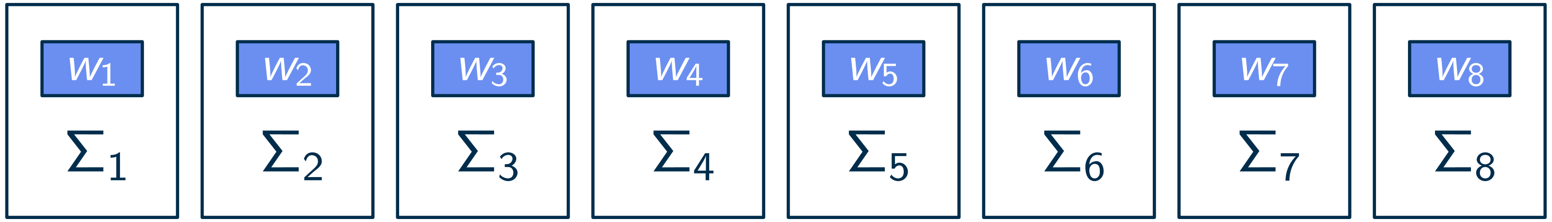
bcacabbacbbab

Disjoint Factors – Cross Composition

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

$\Sigma = \{a, b, c\}$

bcacabbacbbab



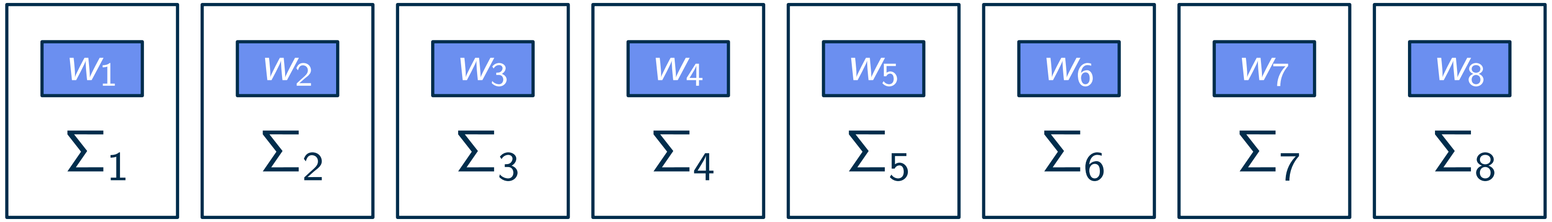
Disjoint Factors – Cross Composition

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?



Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

$\Sigma = \{a, b, c\}$

bcacabbacbbab

“similar”?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6	Σ_7	Σ_8

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

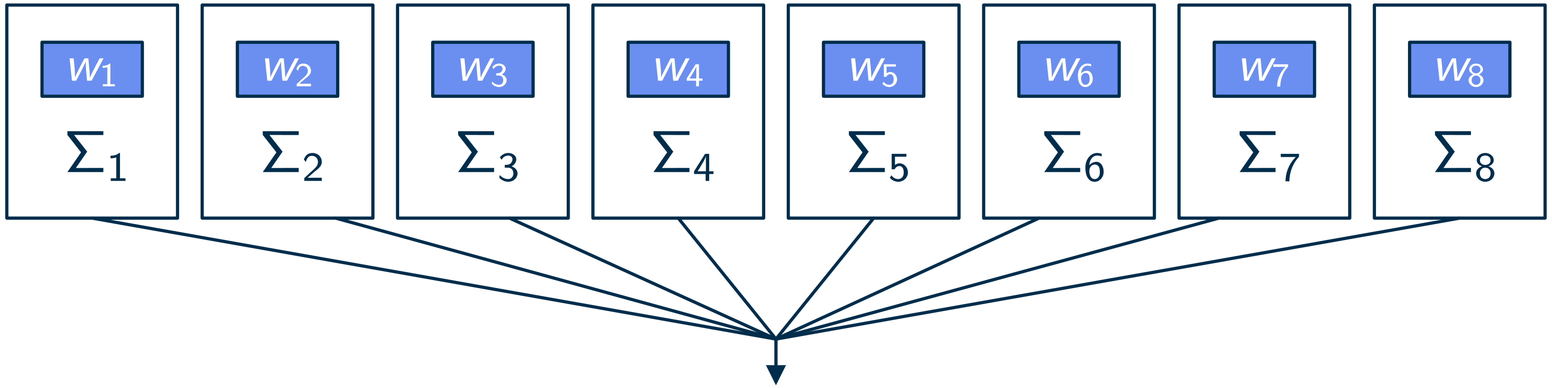
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \ \forall i, j$

wlog $\Sigma_i = \Sigma_1 \ \forall i$



Disjoint Factors – Cross Composition

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

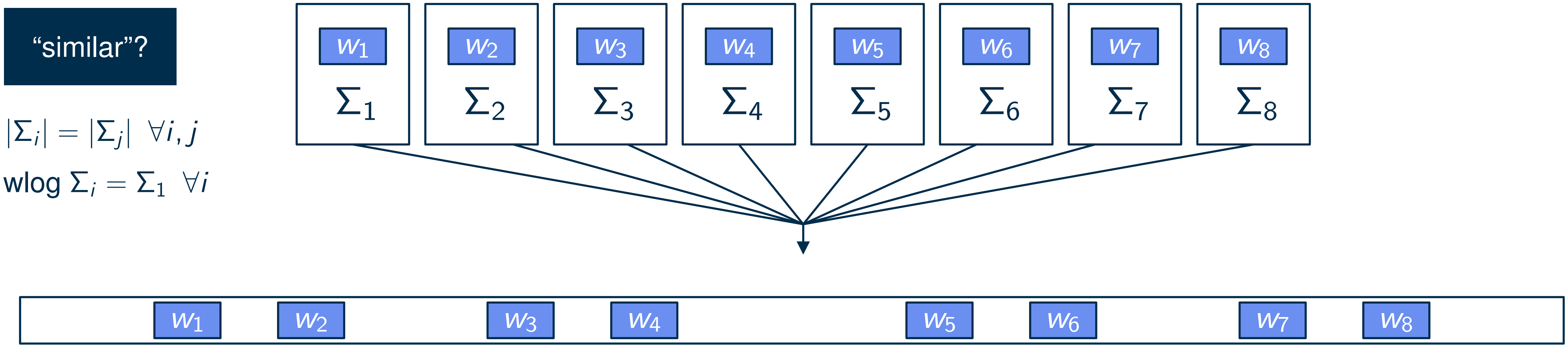
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

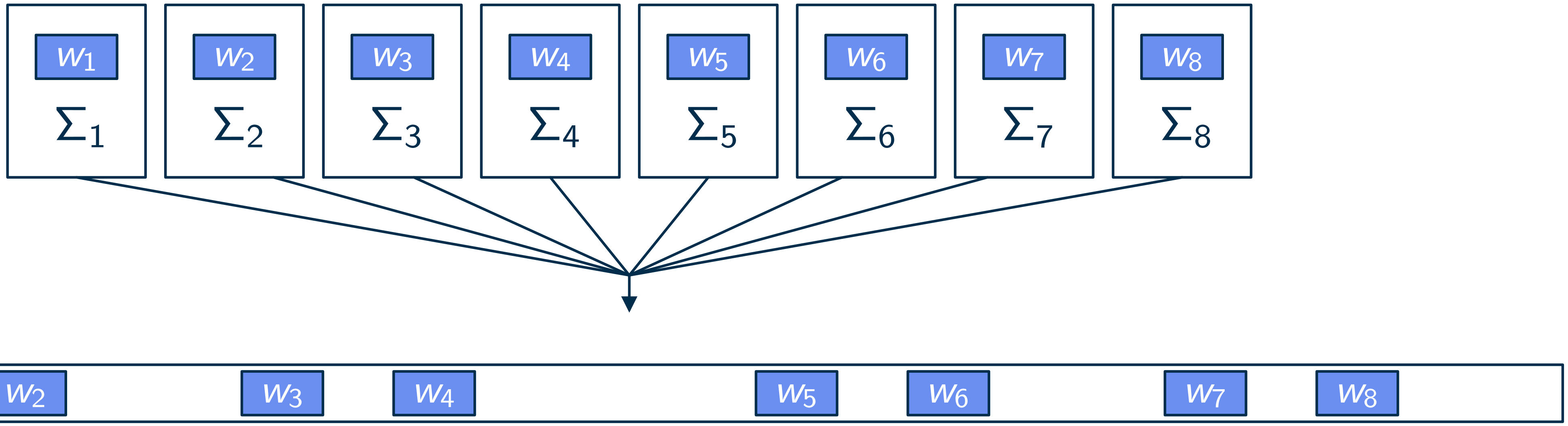
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



$$\Sigma' = \Sigma_1 \cup \underbrace{\{0, 1, 2\}}_{\log t \text{ new letters that don't appear in } \Sigma_1}$$

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

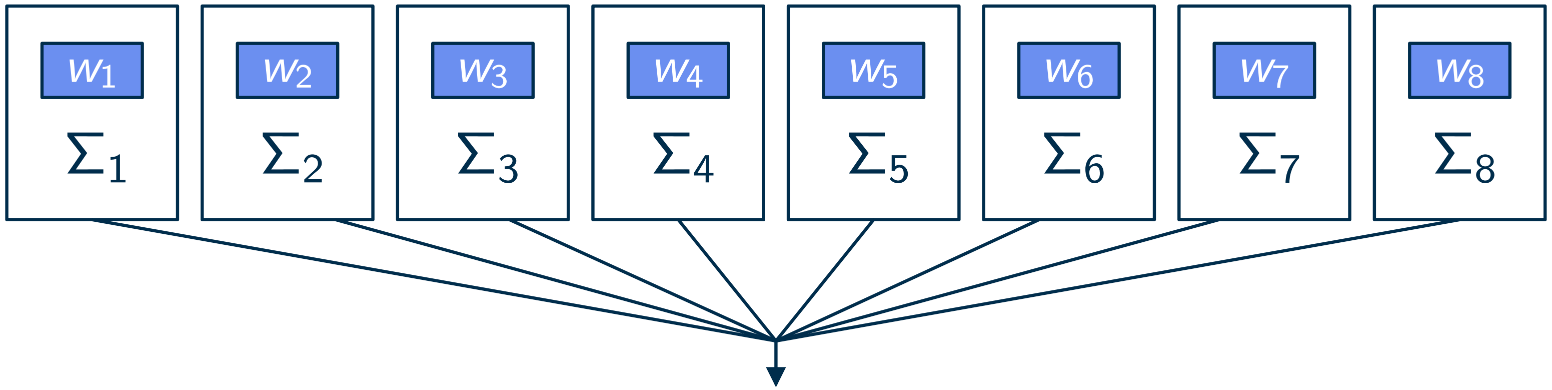
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



$$\Sigma' = \Sigma_1 \cup \underbrace{\{0, 1, 2\}}_{}$$

log t new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

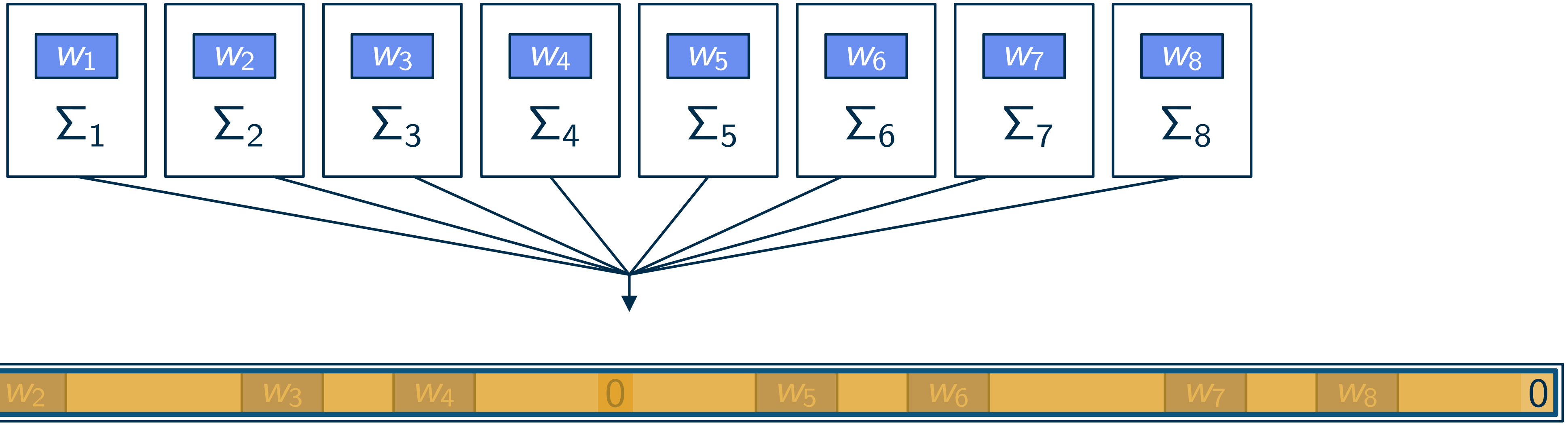
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

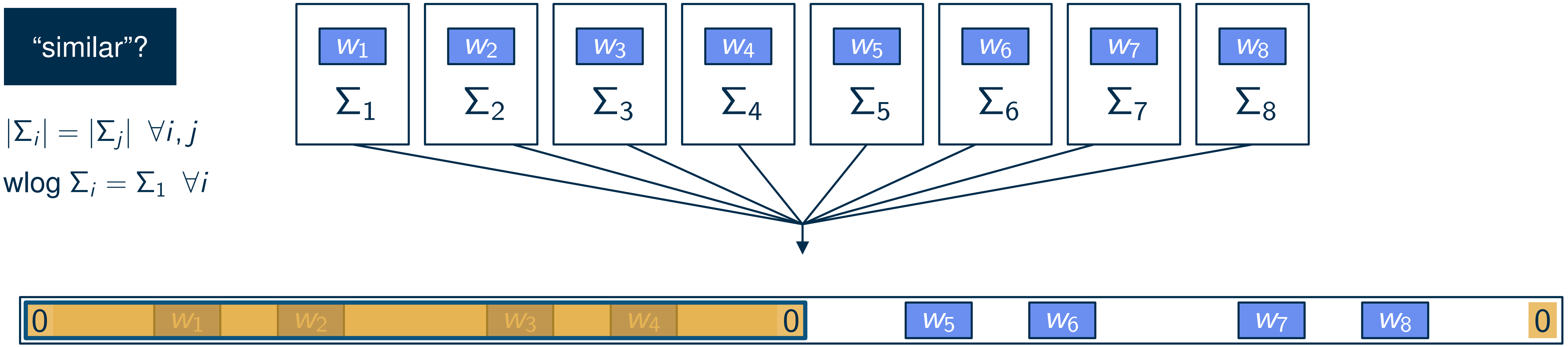
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \ \ \forall i, j$

wlog $\Sigma_i = \Sigma_1 \ \ \forall i$



$$\Sigma' = \Sigma_1 \cup \underbrace{\{0, 1, 2\}}$$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \dots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

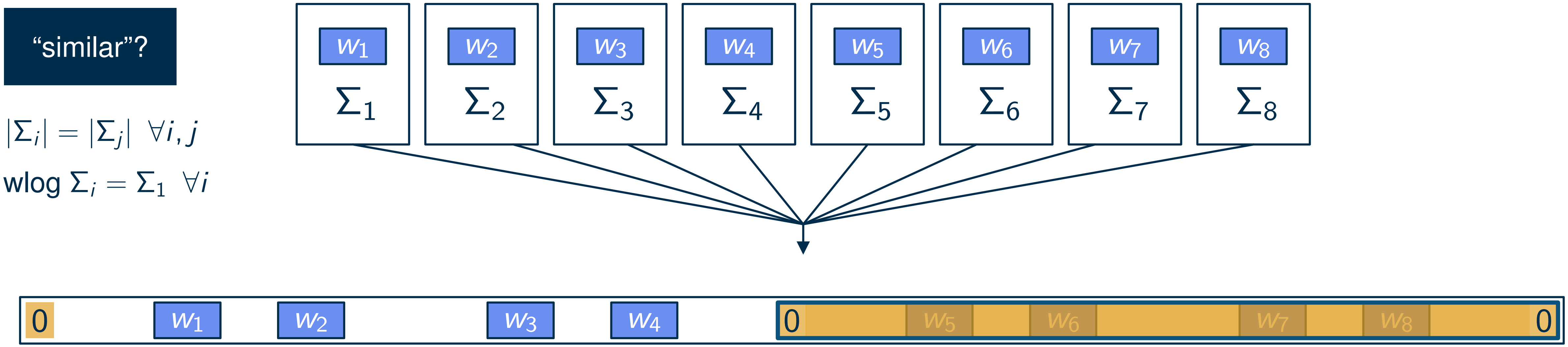
$\Sigma = \{a, b, c\}$

bcacabbacbbab

“similar”?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

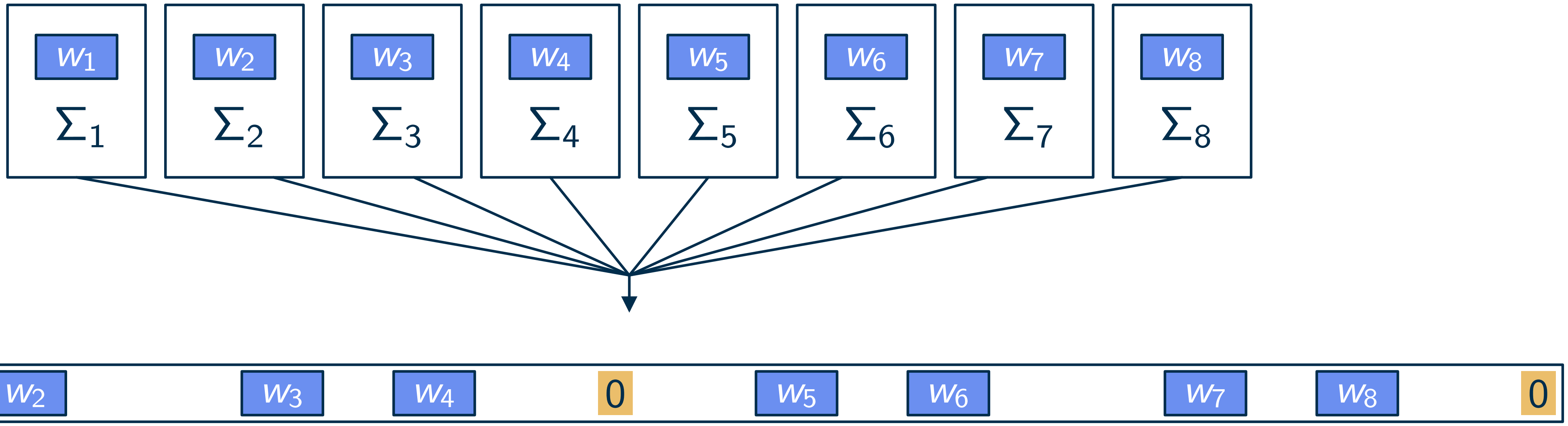
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

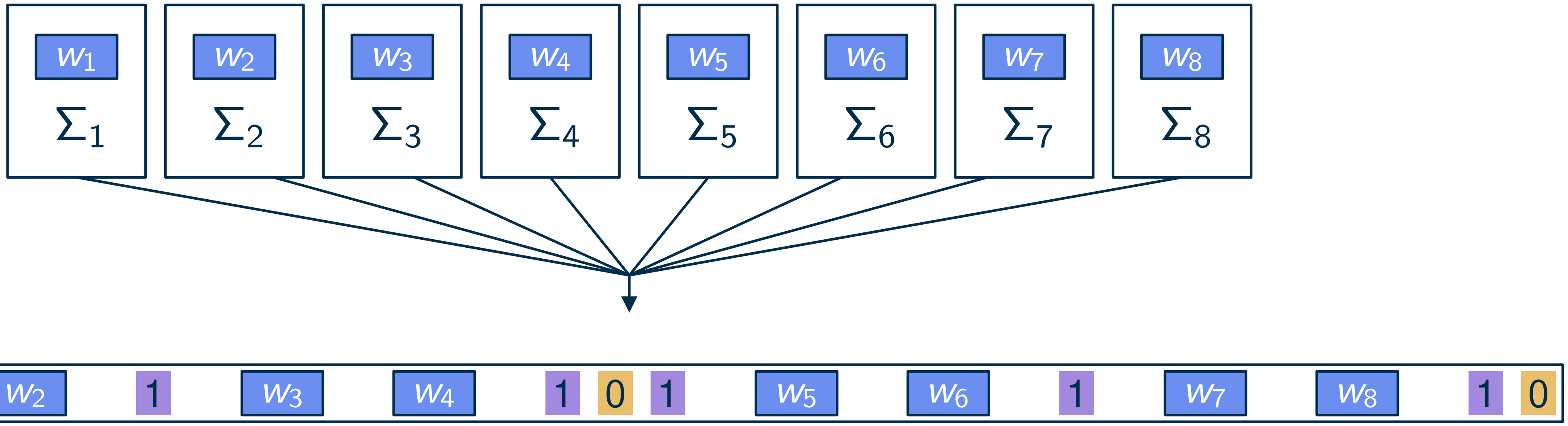
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



$$\Sigma' = \Sigma_1 \cup \underbrace{\{0, 1, 2\}}$$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS (|Σ|): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \dots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

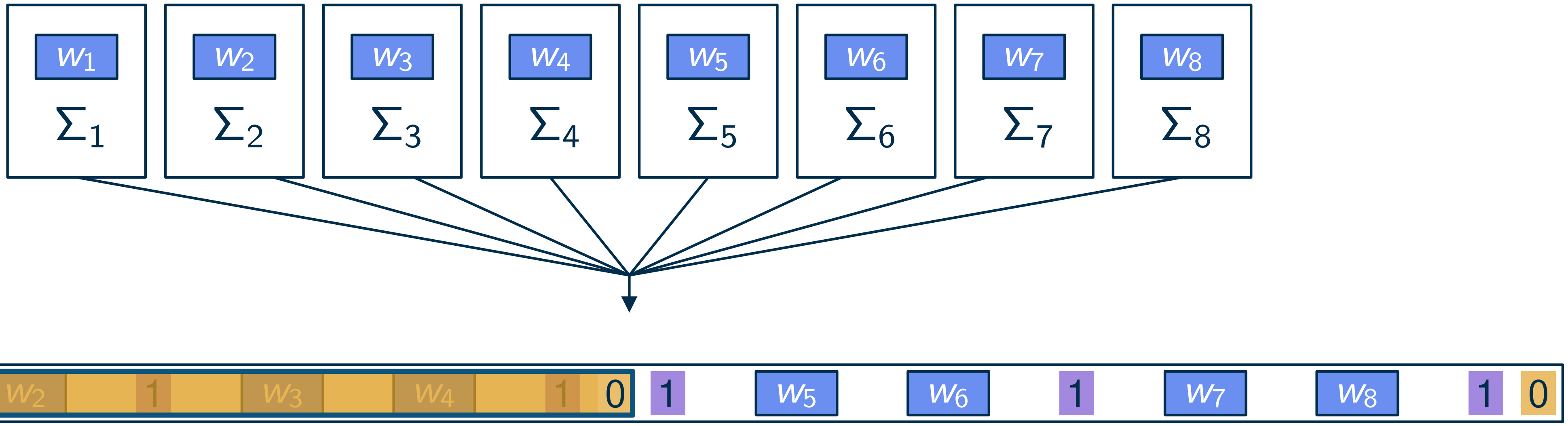
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

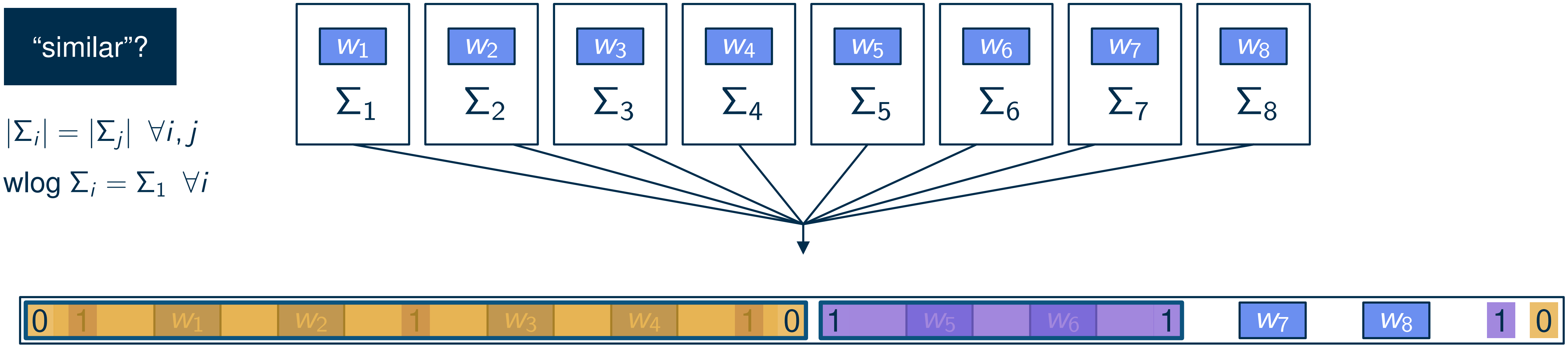
$\Sigma = \{a, b, c\}$

bcacabbacbbab

“similar”?

$|\Sigma_i| = |\Sigma_j| \ \ \forall i, j$

wlog $\Sigma_i = \Sigma_1 \ \ \forall i$



$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

log t new letters that don’t appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

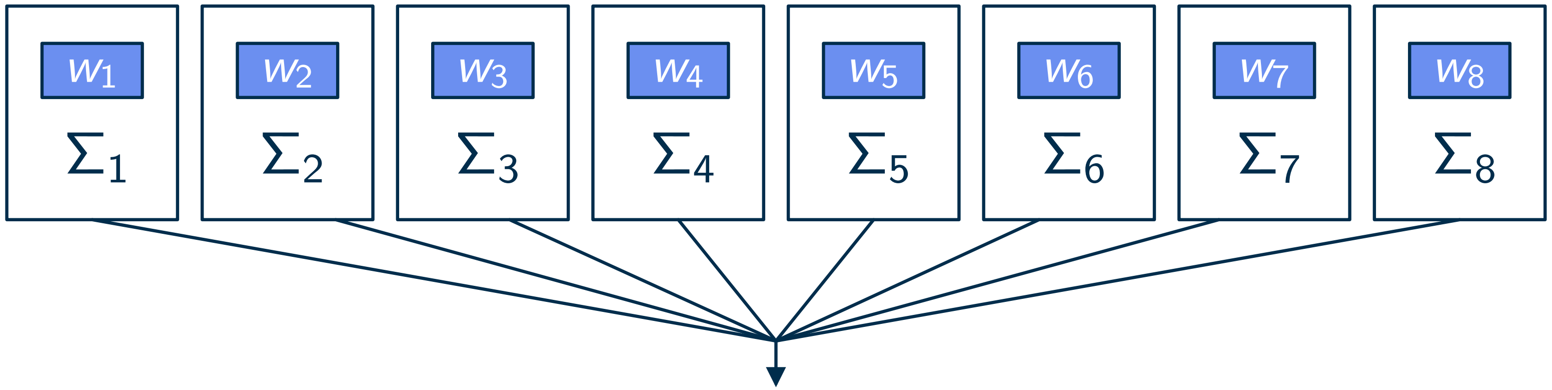
$\Sigma = \{a, b, c\}$

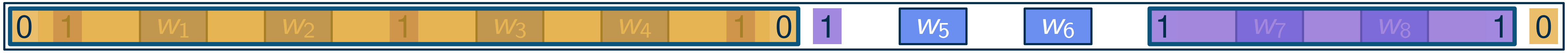
bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \ \ \forall i, j$

wlog $\Sigma_i = \Sigma_1 \ \ \forall i$





$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

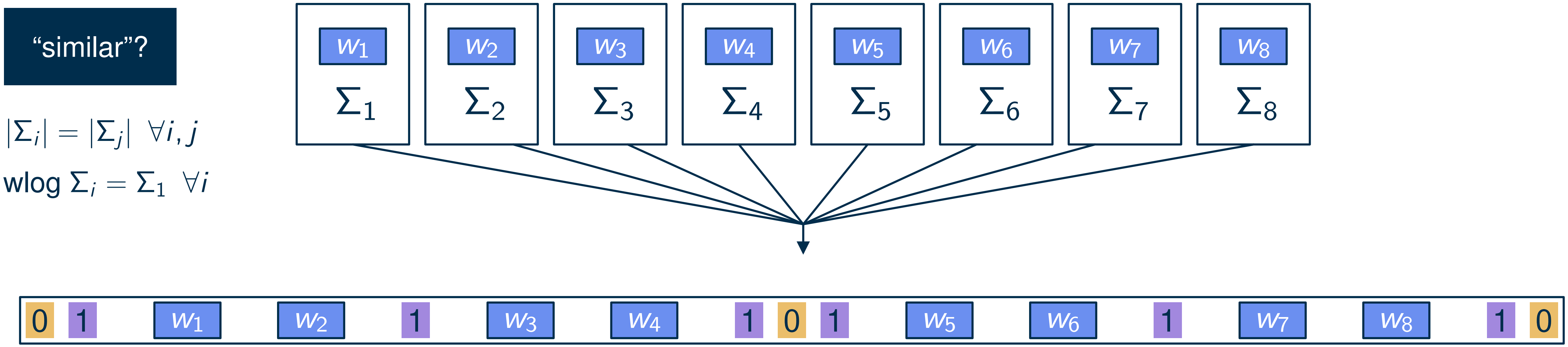
$\Sigma = \{a, b, c\}$

bcacabbacbbab

“similar”?

$|\Sigma_i| = |\Sigma_j| \ \forall i, j$

wlog $\Sigma_i = \Sigma_1 \ \forall i$



$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

$\log t$ new letters that don’t appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

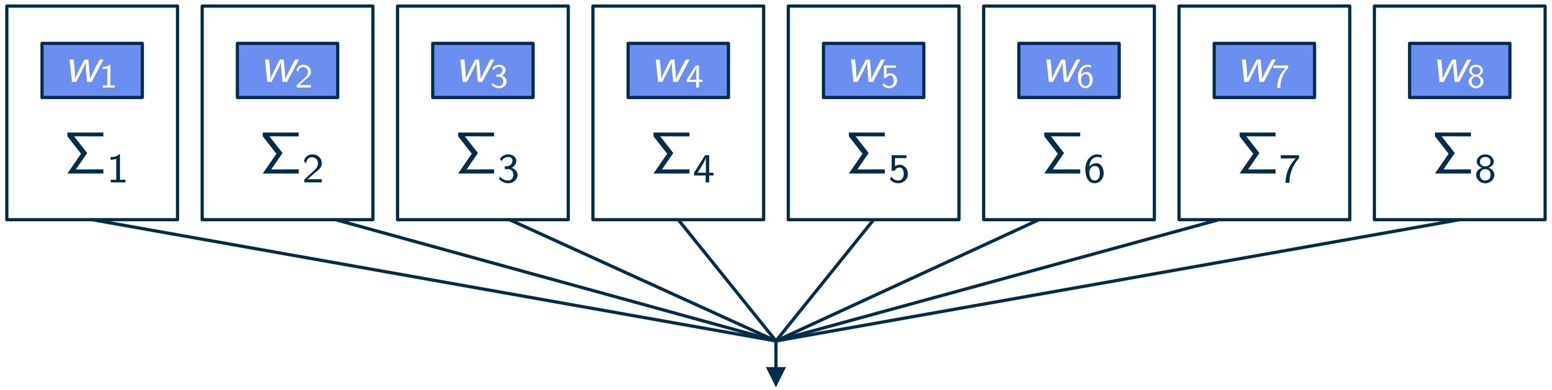
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



0 1 2 w_1 2 w_2 2 1 2 w_3 2 w_4 2 1 0 1 2 w_5 2 w_6 2 1 2 w_7 2 w_8 2 1 0

$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

DISJOINT FACTORS ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

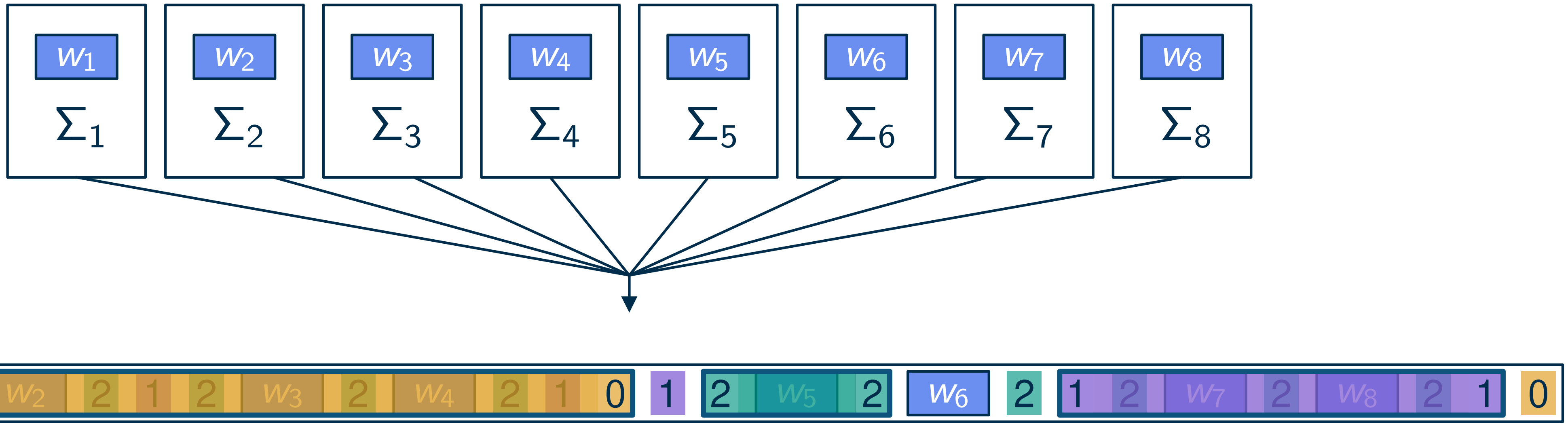
$\Sigma = \{a, b, c\}$

bcacabbacbbab

“similar”?

$|\Sigma_i| = |\Sigma_j| \ \forall i, j$

wlog $\Sigma_i = \Sigma_1 \ \forall i$



$\Sigma' = \Sigma_1 \cup \{0, 1, 2\}$

$\log t$ new letters that don't appear in Σ_1

Disjoint Factors – Cross Composition

Disjoint Factors ($|\Sigma|$): Given a word w over an alphabet $\Sigma = \{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, does it contain pairwise disjoint subwords $u_1, u_2, \ldots, u_s$ such that each u_i has length ≥ 2 and begins and ends with γ_i ?

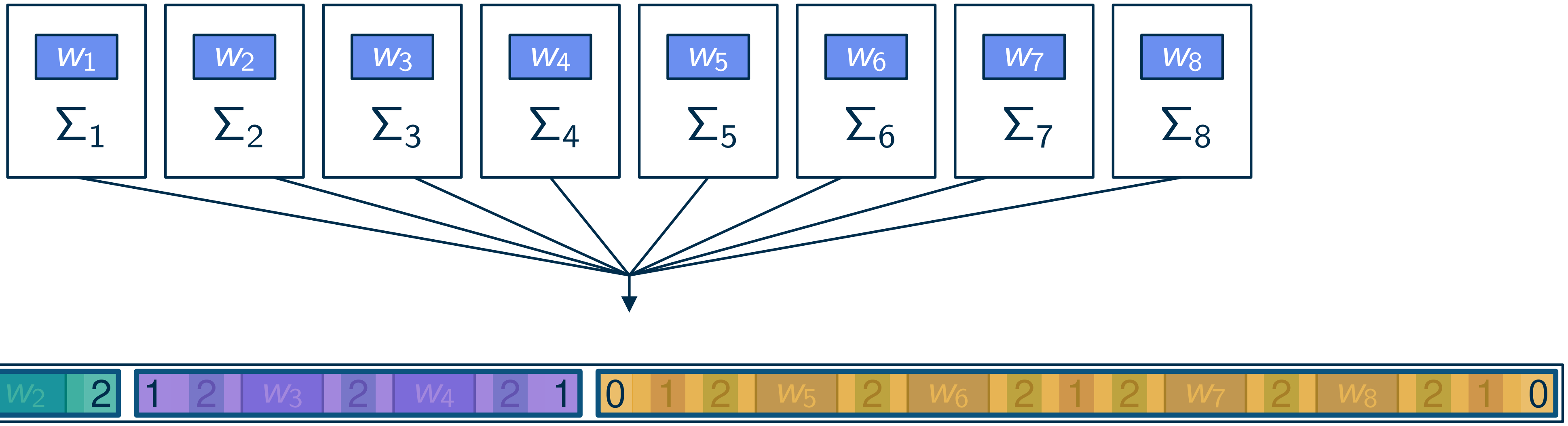
$\Sigma = \{a, b, c\}$

bcacabbacbbab

"similar"?

$|\Sigma_i| = |\Sigma_j| \;\; \forall i, j$

wlog $\Sigma_i = \Sigma_1 \;\; \forall i$



- any choice of subwords u_0, u_1, u_2 leaves exactly one word w_i uncovered

$\Longrightarrow$ solution iff one input instance has solution

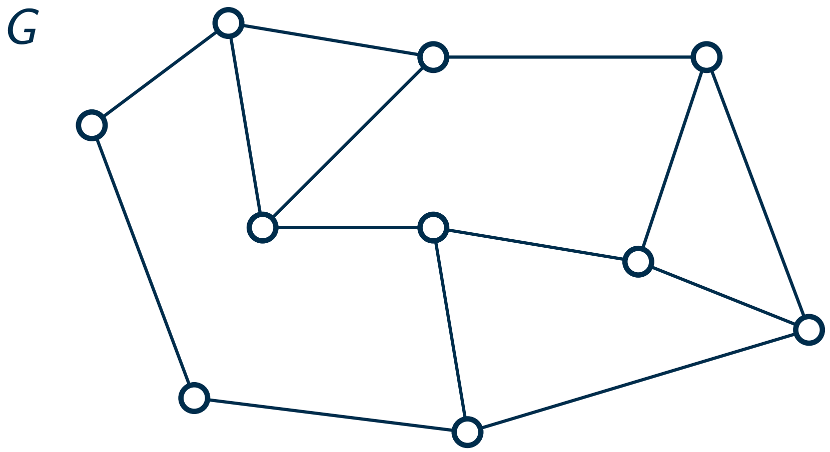
$\Sigma' = \Sigma_1 \cup \underbrace{\{0, 1, 2\}}$

log t new letters that don't appear in Σ_1

Polynomial Parameter Transformation

CONNECTED VERTEX COVER (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

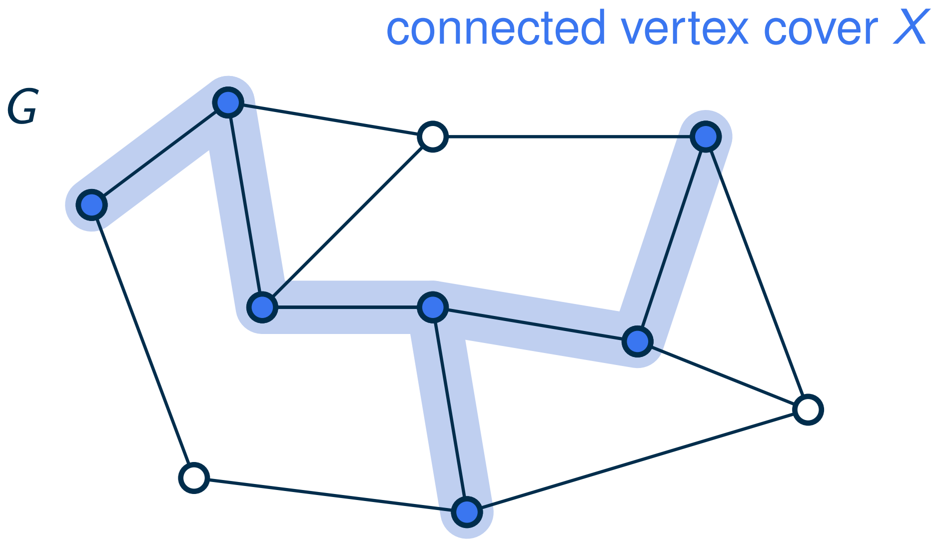
CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



Polynomial Parameter Transformation

CONNECTED VERTEX COVER (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?

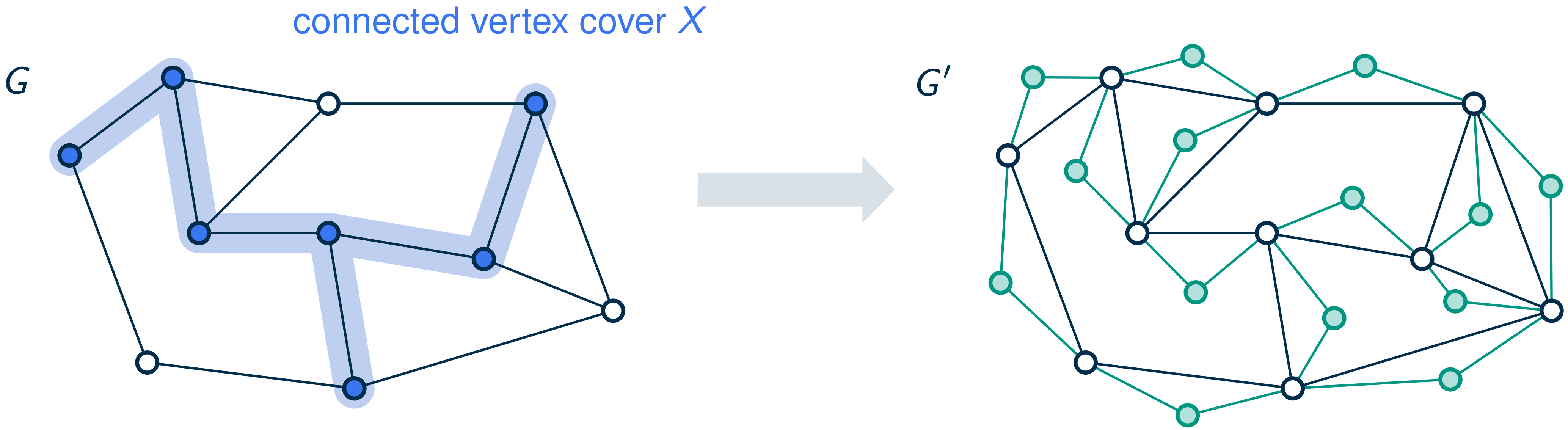


$V \setminus X$ is independent $\implies$ no cycles in $G[V \setminus X]$

Polynomial Parameter Transformation

CONNECTED VERTEX COVER (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?



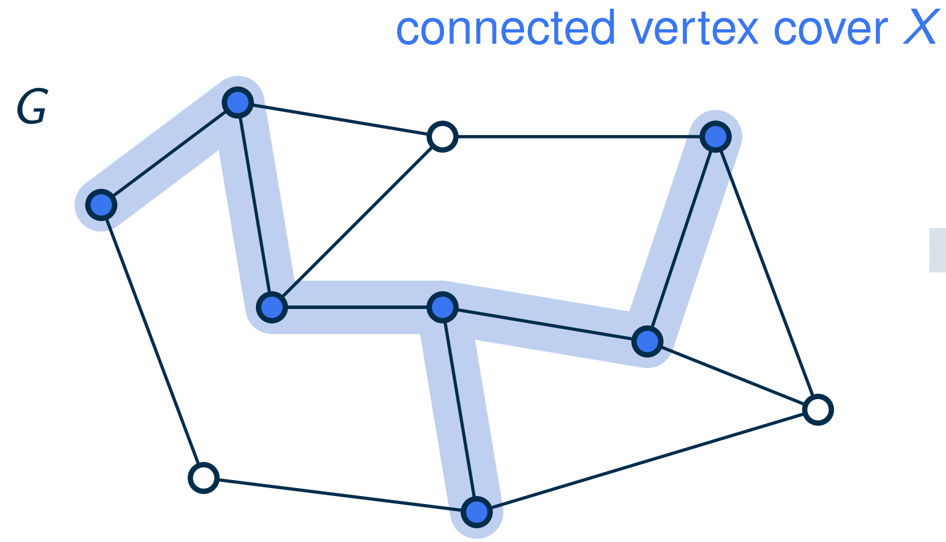
$V \setminus X$ is independent $\implies$ no cycles in $G[V \setminus X]$

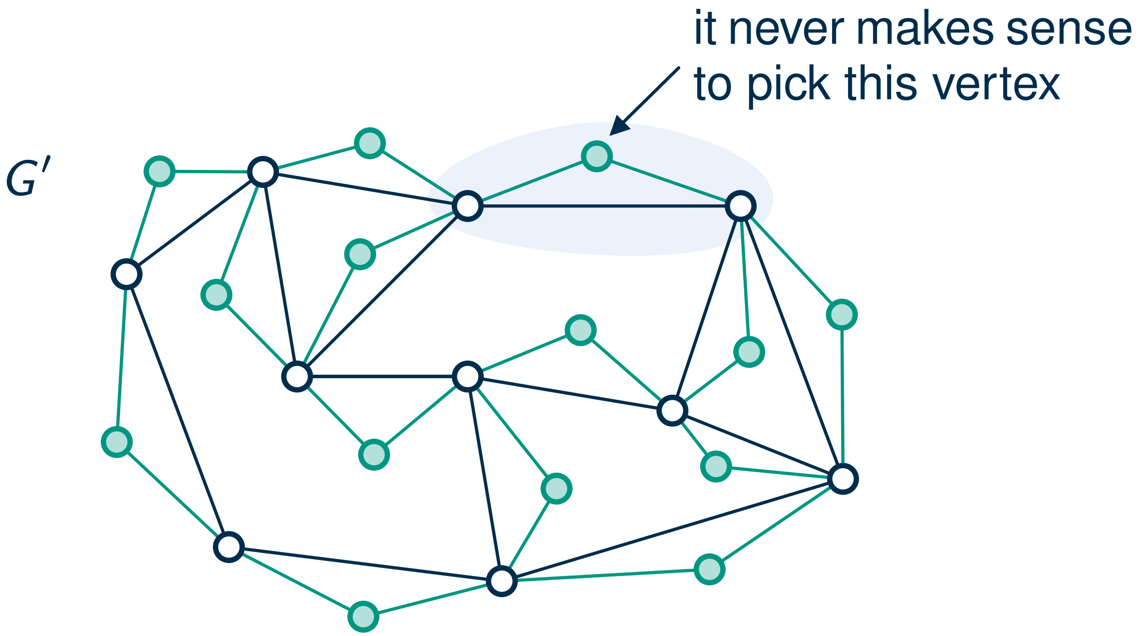
extend every edge to a triangle

Polynomial Parameter Transformation

CONNECTED VERTEX COVER (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

CONNECTED FEEDBACK VERTEX SET (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?





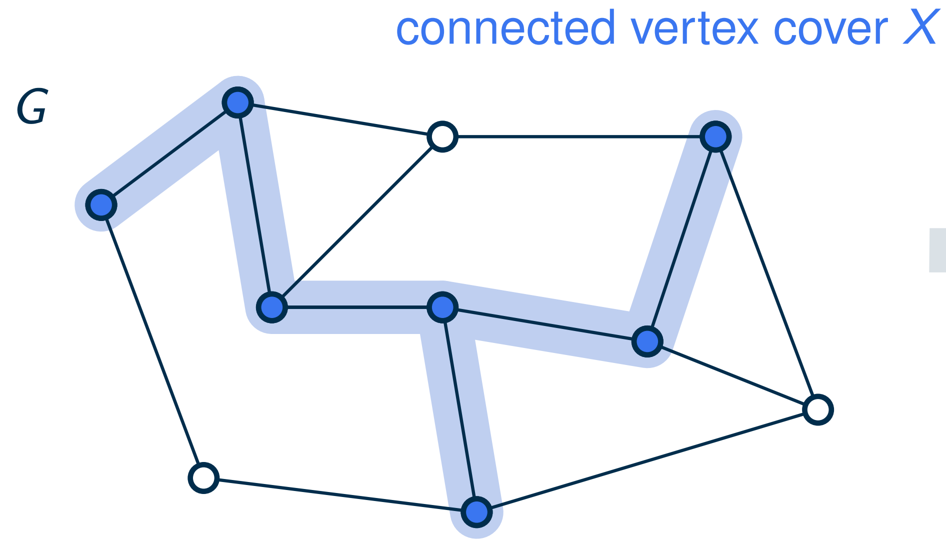
$V \setminus X$ is independent $\implies$ no cycles in $G[V \setminus X]$

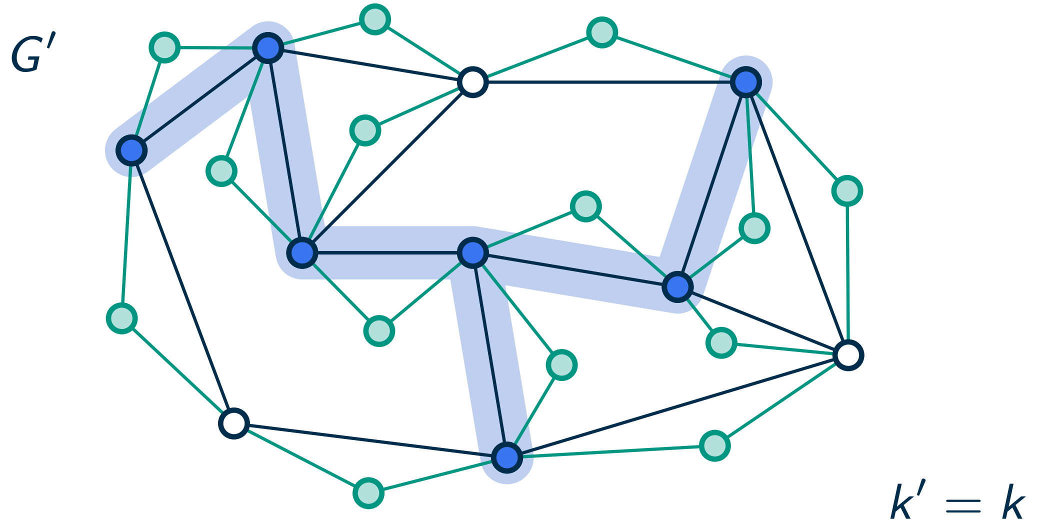
extend every edge to a triangle

Polynomial Parameter Transformation

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

Connected Feedback Vertex Set (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices such that G becomes acyclic?





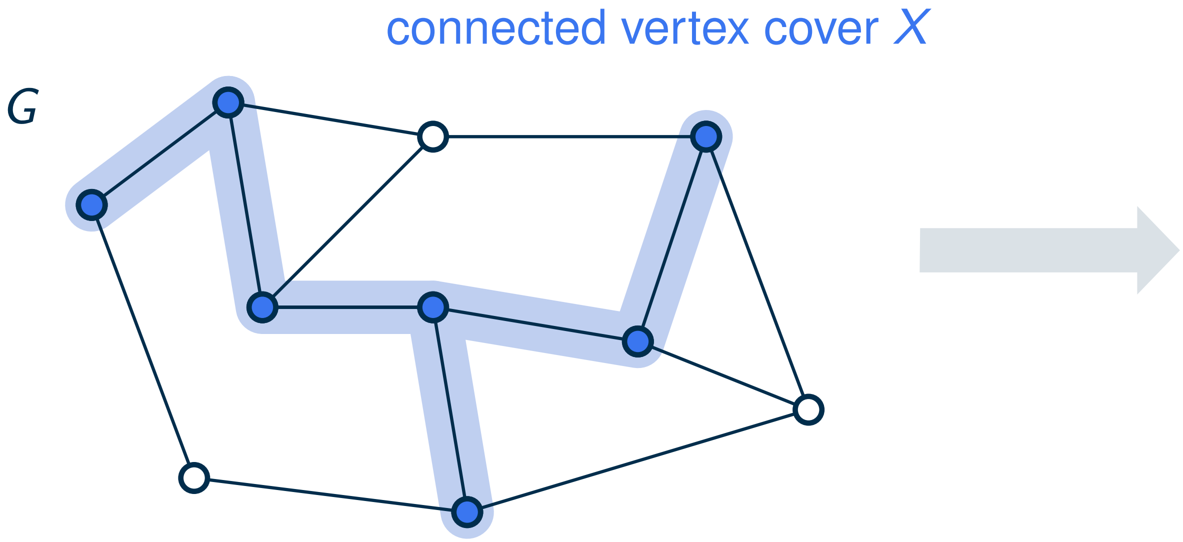
$V \setminus X$ is independent $\implies$ no cycles in $G[V \setminus X]$

extend every edge to a triangle

Polynomial Parameter Transformation

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

Connected Bull Hitting (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



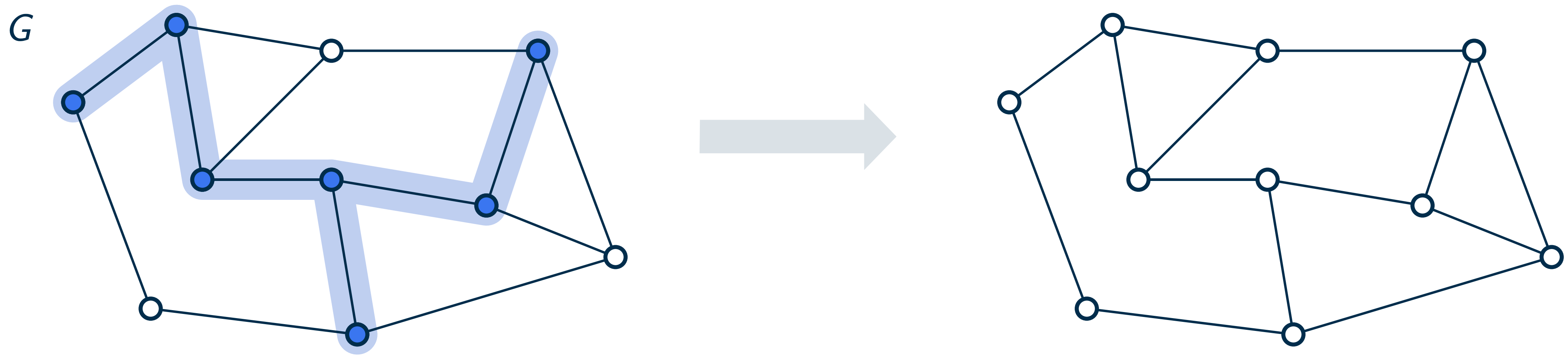
$V \setminus X$ is independent $\implies$ no cycles in $G[V \setminus X]$

Polynomial Parameter Transformation

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

Connected Bull Hitting (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?

connected vertex cover X



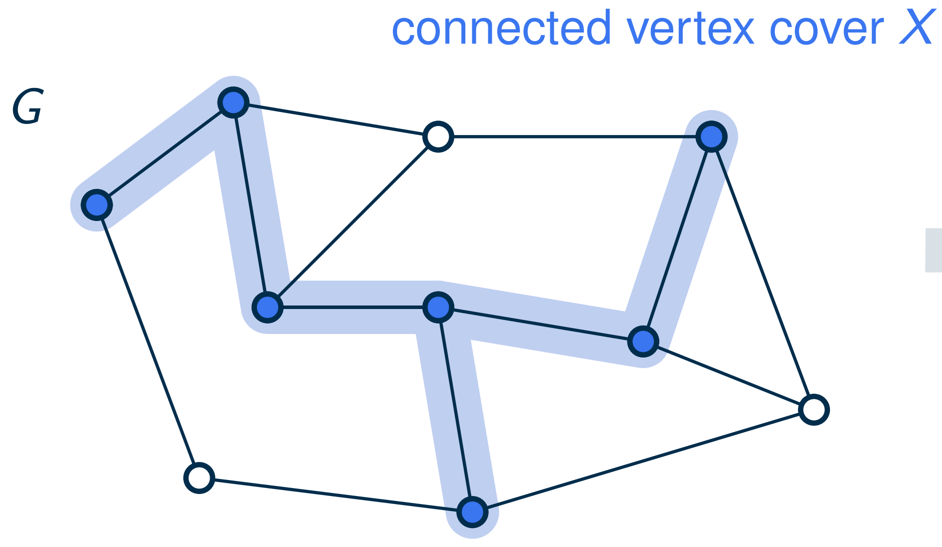
$V \setminus X$ is independent $\implies$ no cycles in $G[V \setminus X]$

extend every edge to a bull

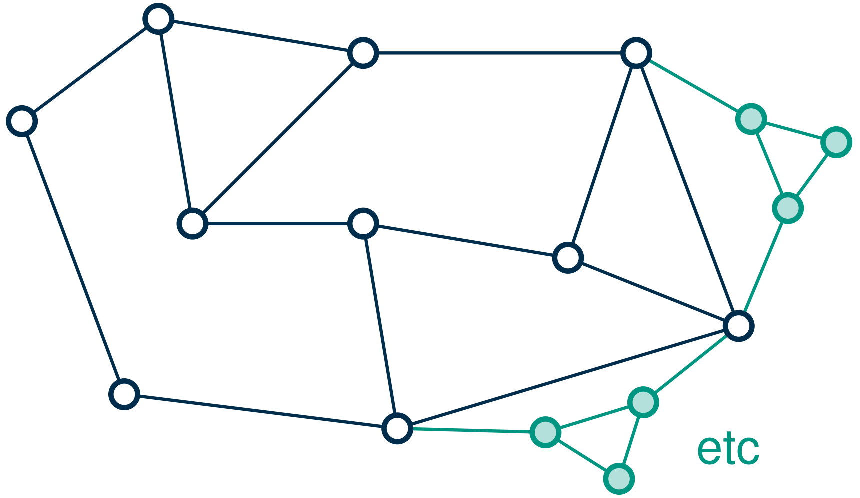
Polynomial Parameter Transformation

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

Connected Bull Hitting (k): Given a graph G and an integer k, can we delete a connected subgraph induced by k vertices s.t. G does not contain any bull after the deletion?



$V \setminus X$ is independent $\implies$ no cycles in $G[V \setminus X]$



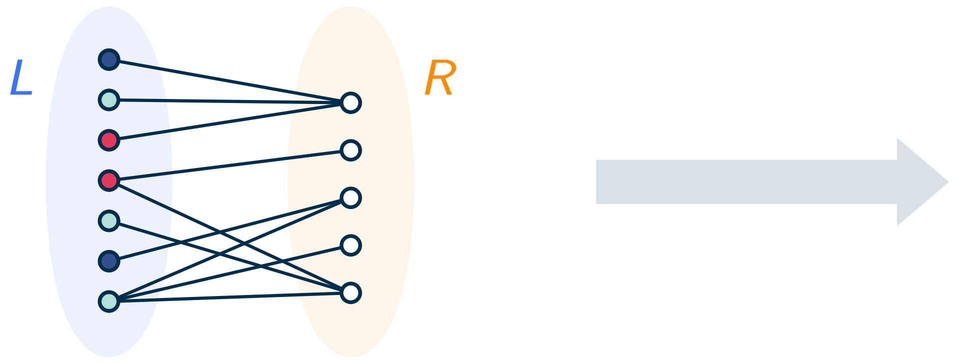
extend every edge to a bull

Polynomial Parameter Transformation

Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



$\ell = 3$

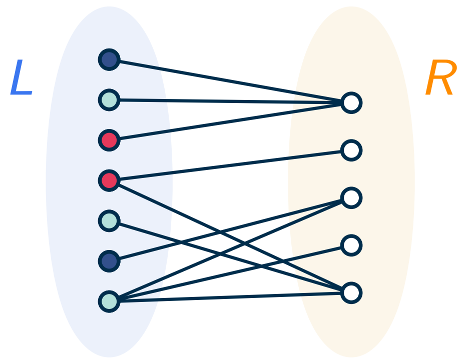
w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

Polynomial Parameter Transformation

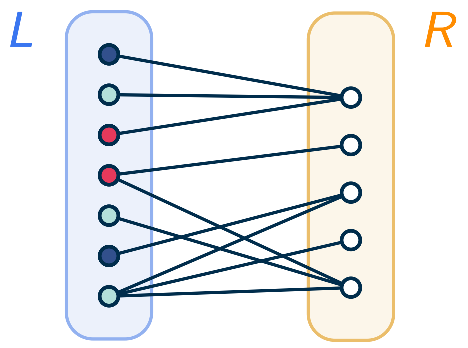
COLORFUL LR DOMSET ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

CONNECTED VERTEX COVER (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



$\ell = 3$



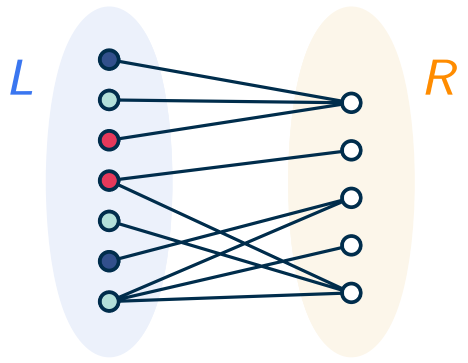
w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

Polynomial Parameter Transformation

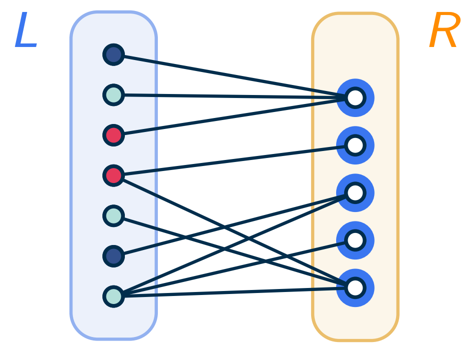
Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



$\ell = 3$



w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

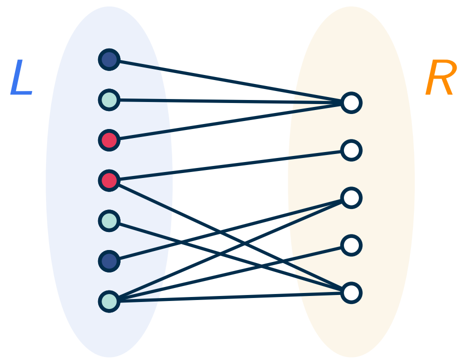
always has vertex cover of size $|R|$
... but not necessarily connected
... and instance not equivalent

Polynomial Parameter Transformation

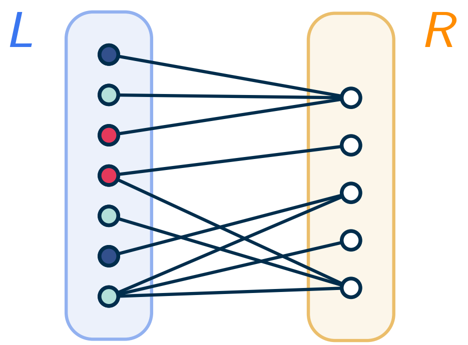
COLORFUL LR DOMSET ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

CONNECTED VERTEX COVER (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



$\ell = 3$



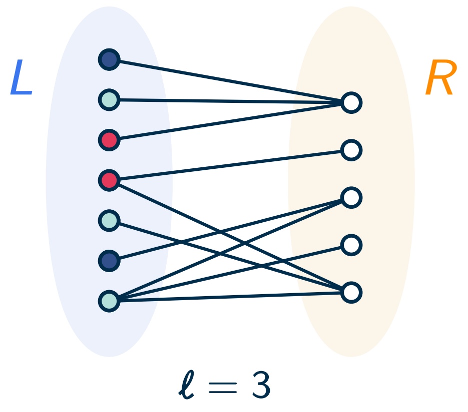
w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

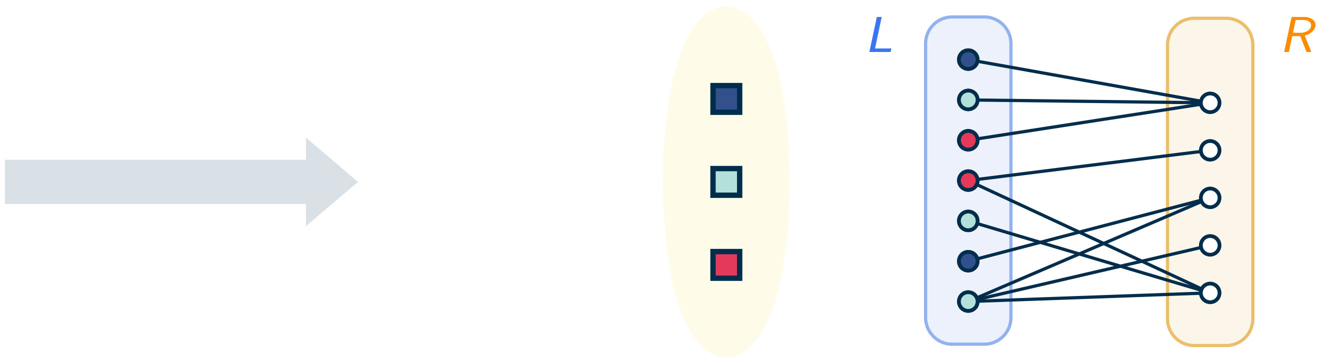
Polynomial Parameter Transformation

Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$





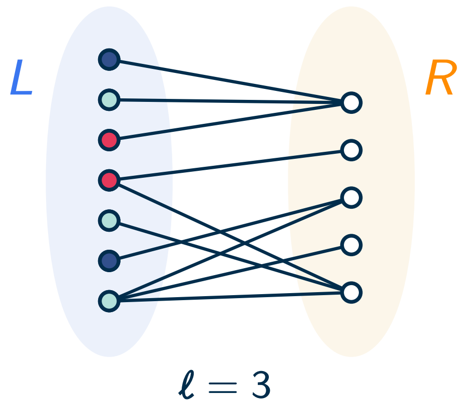
w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

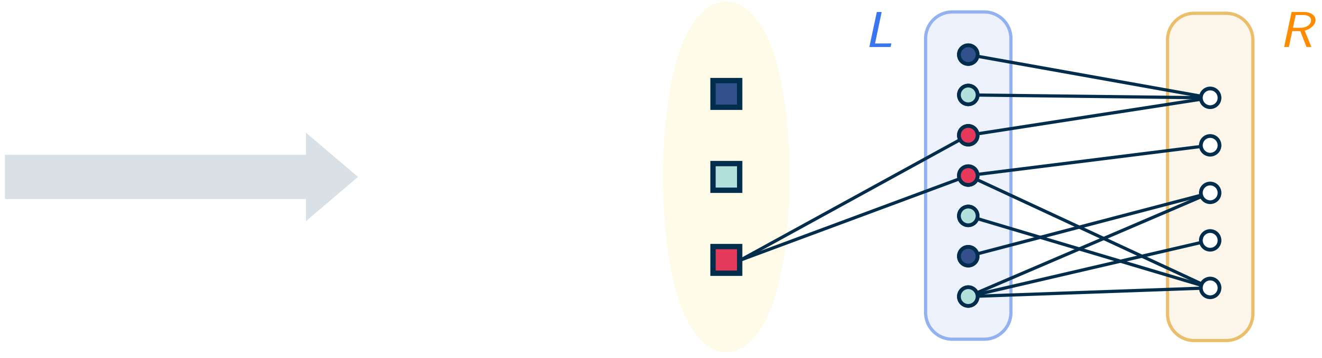
Polynomial Parameter Transformation

Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$





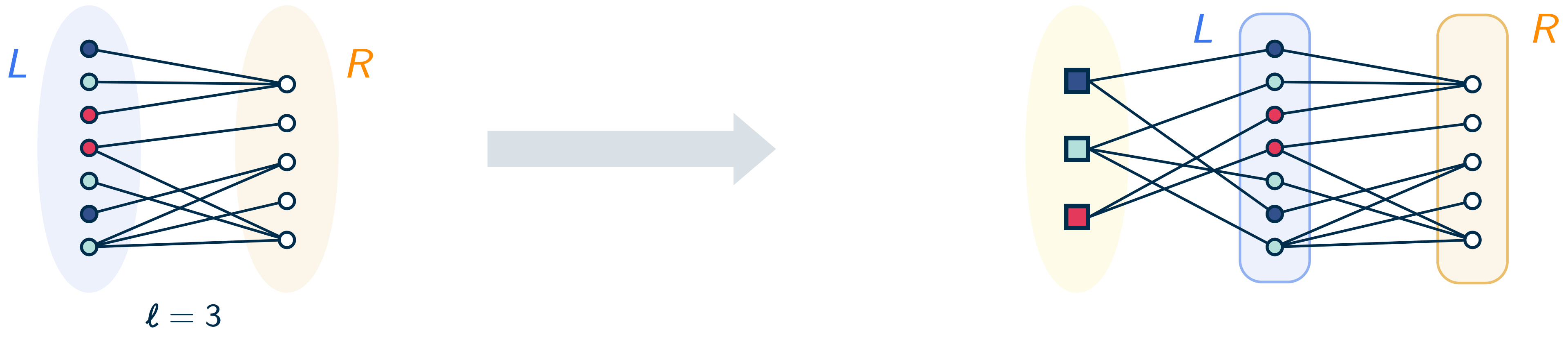
w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

Polynomial Parameter Transformation

Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



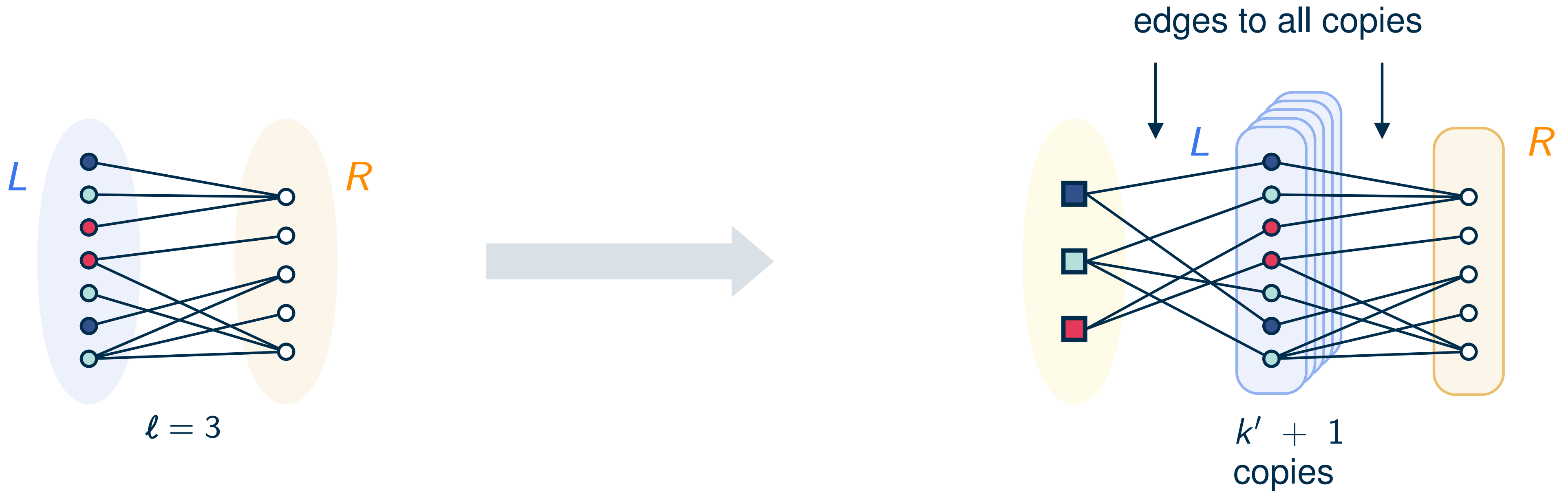
w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

Polynomial Parameter Transformation

Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



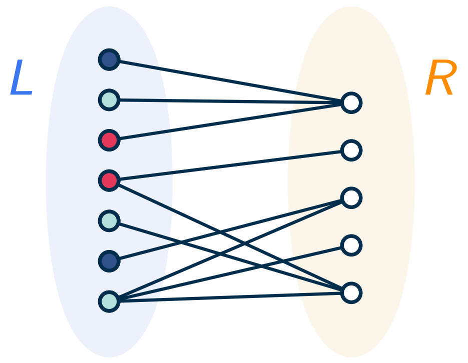
w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

Polynomial Parameter Transformation

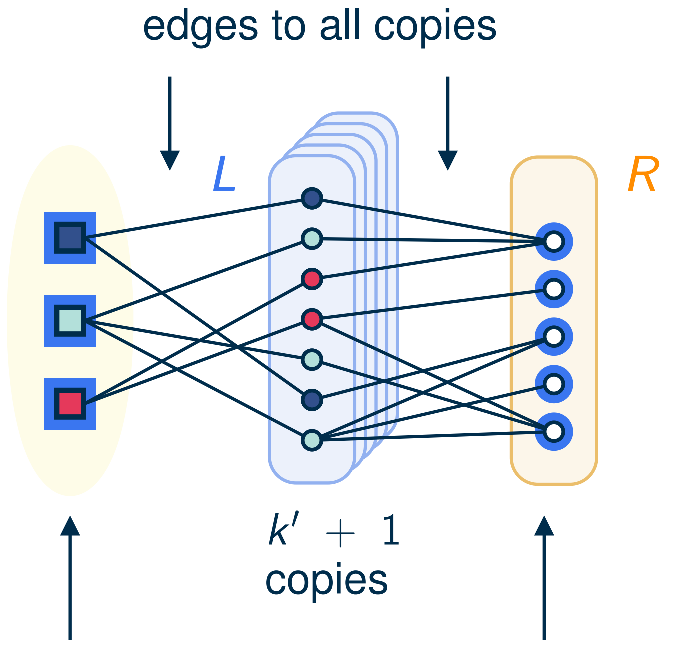
COLORFUL LR DOMSET ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

CONNECTED VERTEX COVER (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



$\ell = 3$



w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

all these vertices have degree $\geq k' + 1$

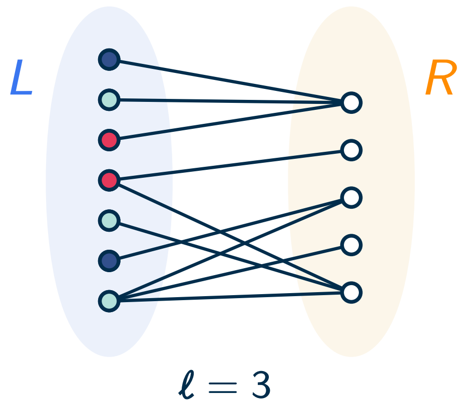
$\Longrightarrow$ must be selected, yields vertex cover

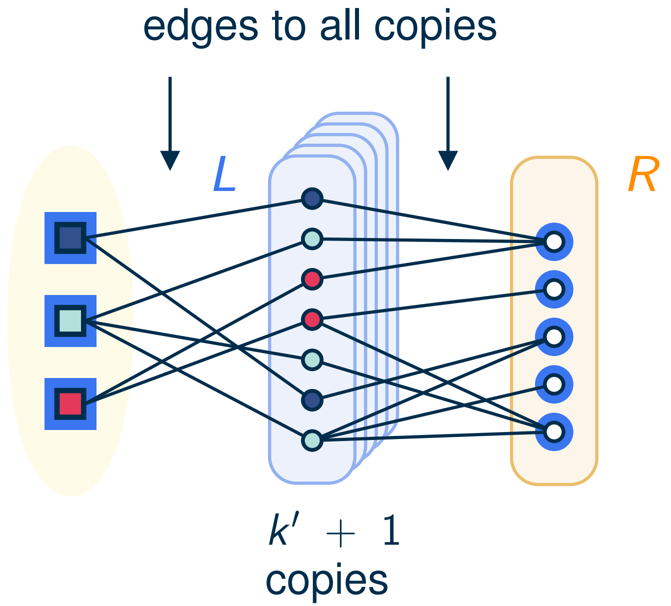
Polynomial Parameter Transformation

COLORFUL LR DOMSET ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

CONNECTED VERTEX COVER (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$





w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

to achieve connectedness:

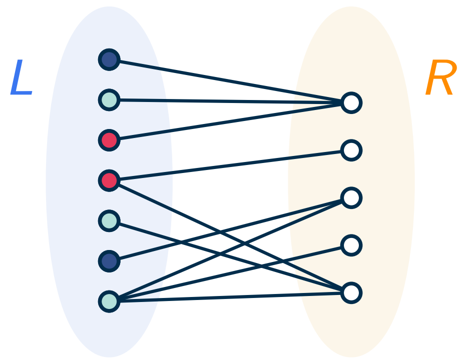
- in L: choose one of each color
- in L: choose set that dominates R

Polynomial Parameter Transformation

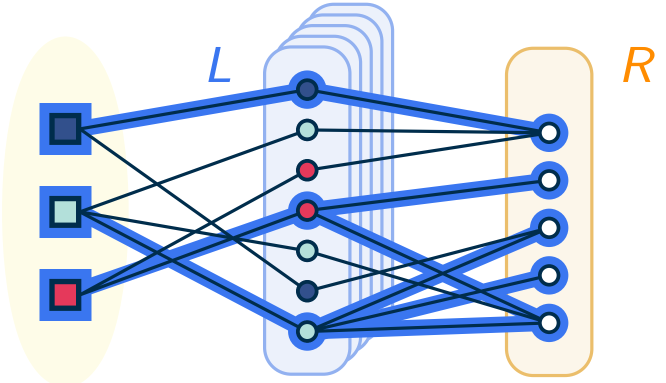
Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



$\ell = 3$



w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

to achieve connectedness:

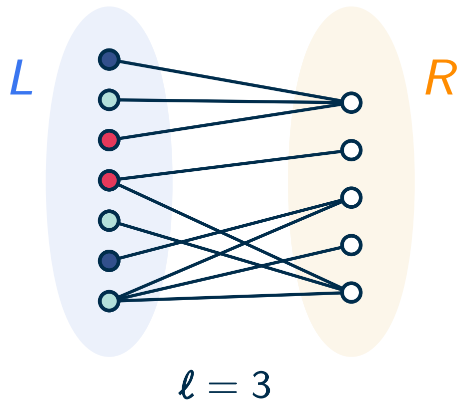
- in L: choose one of each color
- in L: choose set that dominates R

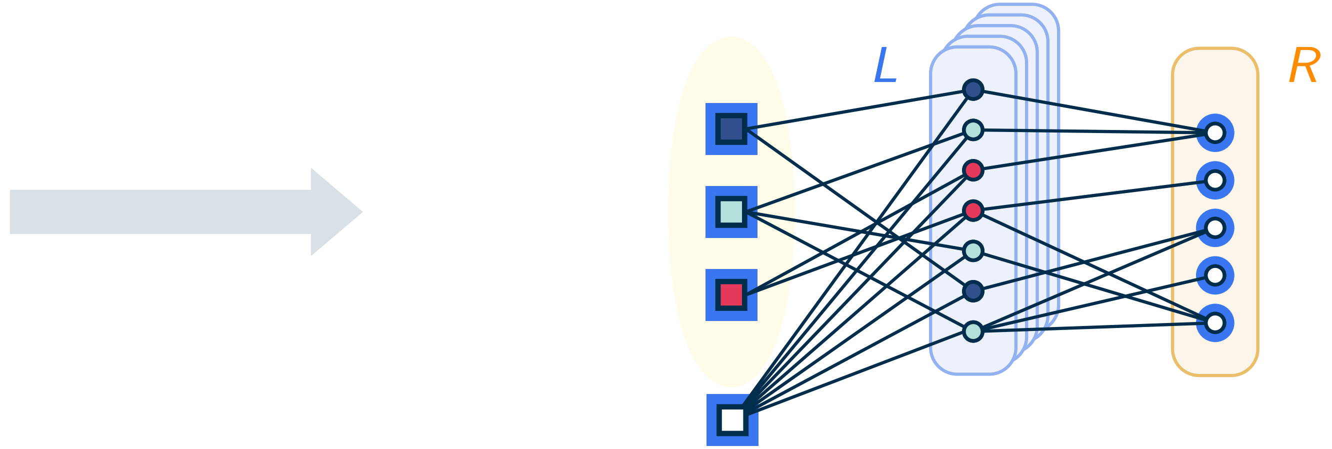
Polynomial Parameter Transformation

Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$





w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

to achieve connectedness:

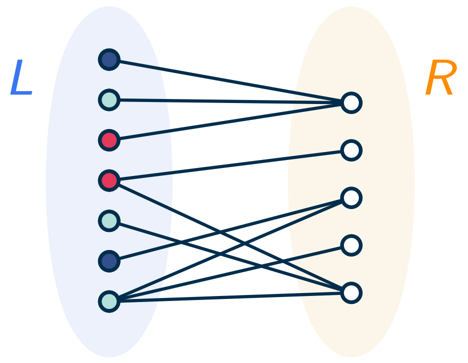
- in L: choose one of each color
- in L: choose set that dominates R
- add universal vertex for L

Polynomial Parameter Transformation

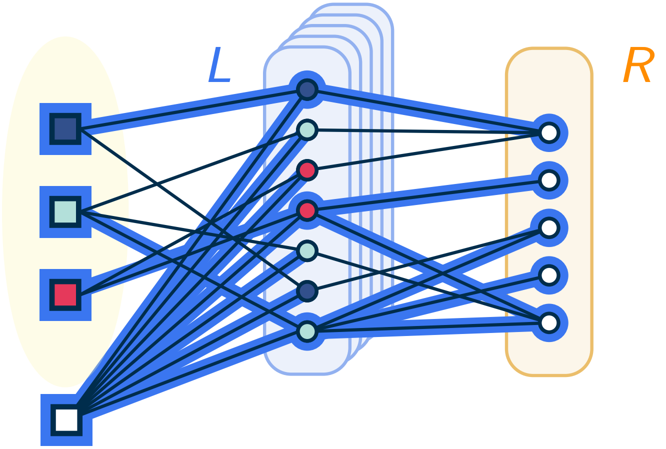
Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k = |R| + \ell$



$\ell = 3$



w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

to achieve connectedness:

- in L: choose one of each color
- in L: choose set that dominates R
- add universal vertex for L

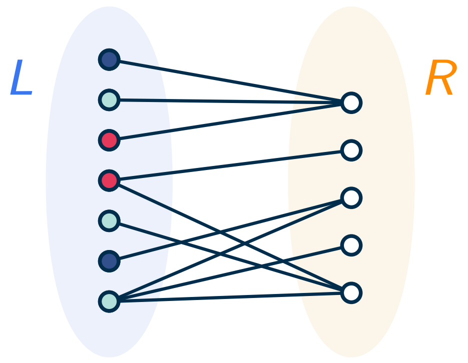
Polynomial Parameter Transformation

Colorful LR DomSet ($k = |R| + \ell$): Given a bipartite graph G with sides L and R and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of size exactly ℓ that dominates all vertices in R?

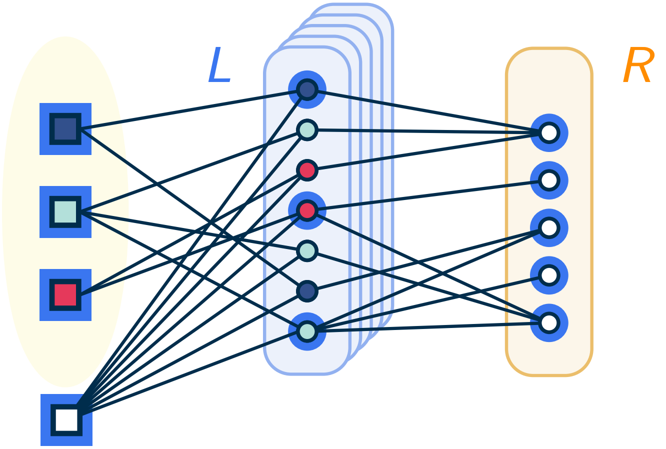
Connected Vertex Cover (k): Given a graph G and an integer k, does G contain a vertex cover that induces a connected subgraph of G?

$k' = |R| + 2\ell + 1 \leq 2k + 1$

$k = |R| + \ell$



$\ell = 3$



w.l.o.g. there are no isolated vertices in L (otherwise reduce instance)

to achieve connectedness:

- in L: choose one of each color
- in L: choose set that dominates R
- add universal vertex for L