Reductions in local certification

Louis Esperet, Sébastien Zeitoun

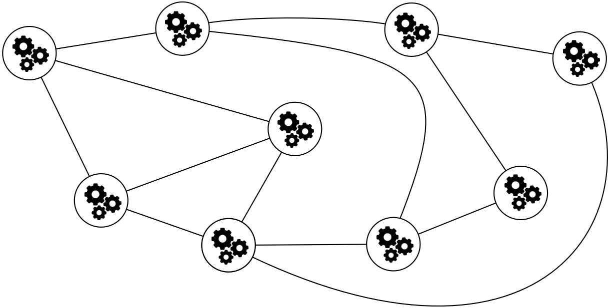
June 13, 2025

Local certification

Local certification

Context: distributed computing

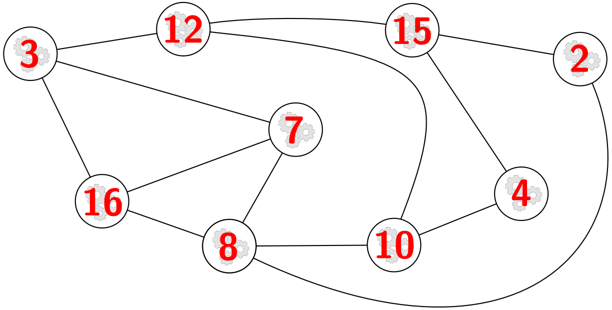
Model: graph, $\begin{cases} \text{vertices} = \text{computation units} \\ \text{edges} = \text{communication channels} \end{cases}$



Local certification

Context: distributed computing

Model: graph, $\left\{ \begin{array}{l} \text{vertices} = \text{computation units} \longrightarrow \text{have } \textcolor{red}{\text{unique identifiers}} \text{ in } \{1, \ldots, n^c\} \\ \text{edges} = \text{communication channels} \end{array} \right.$

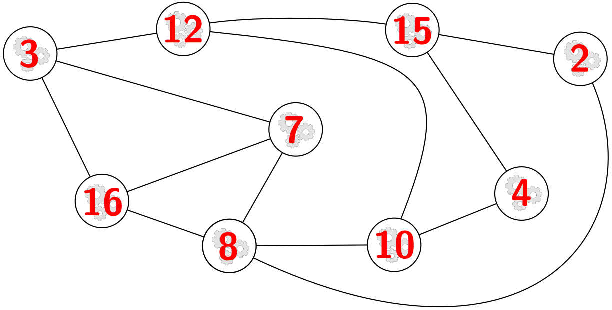


Local certification

Context: distributed computing

Model: graph, $\left\{ \begin{array}{l} \text{vertices} = \text{computation units} \longrightarrow \text{have unique identifiers in } \{1, \dots, n^c\} \\ \text{edges} = \text{communication channels} \end{array} \right.$

Goal: verify locally a graph property $\mathcal{P}$, thanks to certificates

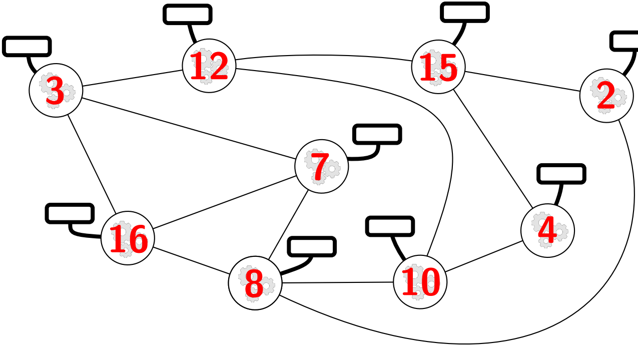


Local certification

Context: distributed computing

Model: graph, $\begin{cases} \text{vertices} = \text{computation units} \longrightarrow \text{have unique identifiers in } \{1, \ldots, n^c\} \\ \text{edges} = \text{communication channels} \end{cases}$

Goal: verify locally a graph property $\mathcal{P}$, thanks to certificates

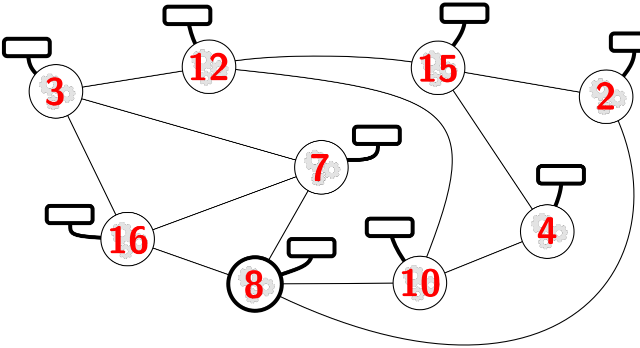


Local certification

Context: distributed computing

Model: graph, $\begin{cases} \text{vertices} = \text{computation units} \longrightarrow \text{have unique identifiers in } \{1, \ldots, n^c\} \\ \text{edges} = \text{communication channels} \end{cases}$

Goal: verify locally a graph property $\mathcal{P}$, thanks to certificates

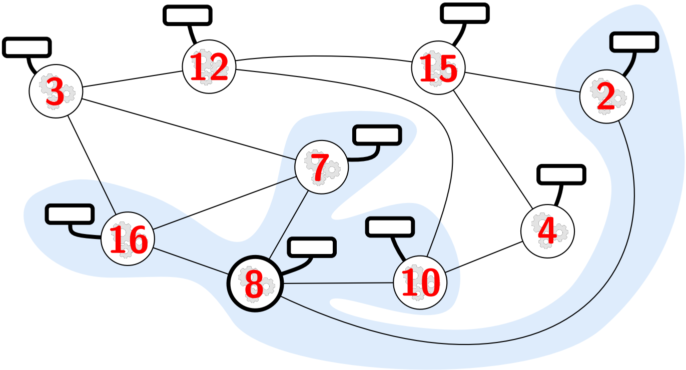


Local certification

Context: distributed computing

Model: graph, $\begin{cases} \text{vertices} = \text{computation units} & \longrightarrow \text{have unique identifiers in } \{1, \dots, n^c\} \\ \text{edges} = \text{communication channels} \end{cases}$

Goal: verify locally a graph property $\mathcal{P}$, thanks to certificates

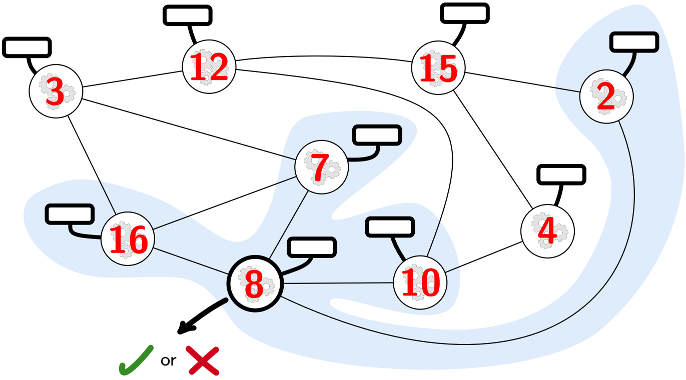


Local certification

Context: distributed computing

Model: graph, $\begin{cases} \text{vertices} = \text{computation units} \longrightarrow \text{have unique identifiers in } \{1, \ldots, n^c\} \\ \text{edges} = \text{communication channels} \end{cases}$

Goal: verify locally a graph property $\mathcal{P}$, thanks to certificates

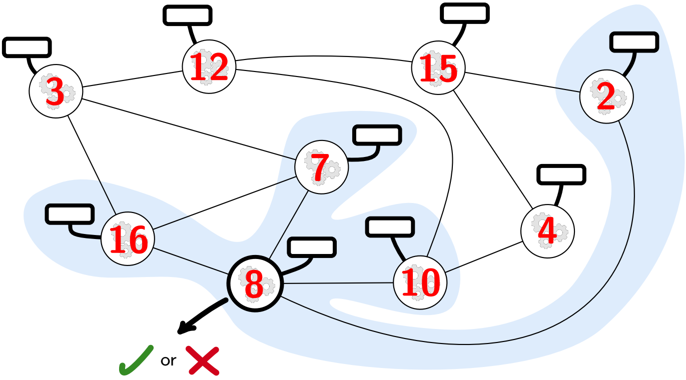


Local certification

Context: distributed computing

Model: graph, $\begin{cases} \text{vertices} = \text{computation units} \longrightarrow \text{have } \textcolor{red}{\text{unique identifiers}} \text{ in } \{1, \ldots, n^c\} \\ \text{edges} = \text{communication channels} \end{cases}$

Goal: verify locally a graph property $\mathcal{P}$, thanks to certificates



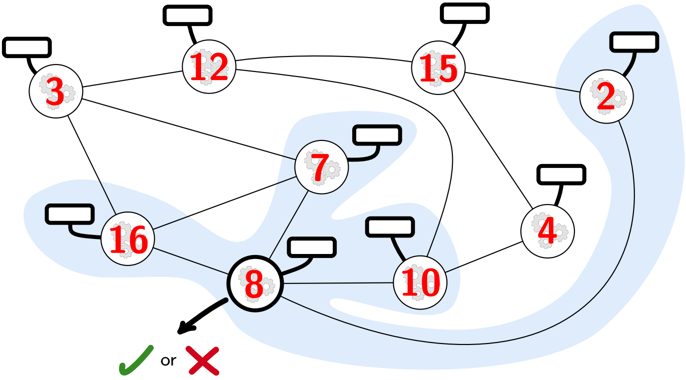
Graph (globally) accepted $\iff$ all the vertices accept (consensus)

Local certification

Context: distributed computing

Model: graph, $\begin{cases} \text{vertices} = \text{computation units} \longrightarrow \text{have unique identifiers in } \{1, \ldots, n^c\} \\ \text{edges} = \text{communication channels} \end{cases}$

Goal: verify locally a graph property $\mathcal{P}$, thanks to certificates

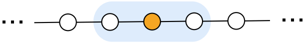


Graph (globally) accepted $\iff$ all the vertices accept (consensus)

G satisfies $\mathcal{P} \iff$ there exists an assignment of the certificates such that G is accepted

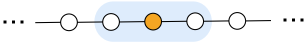
Example: how to certify that a graph is a path ?

Example: how to certify that a graph is a path ?



$\longrightarrow$ Path ? Cycle ?

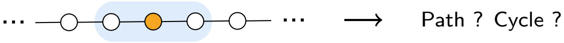
Example: how to certify that a graph is a path ?



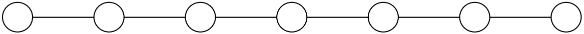
$\longrightarrow$ Path ? Cycle ?

Certificate = distance to a fixed endpoint.

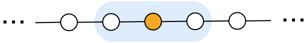
Example: how to certify that a graph is a path ?



Certificate = distance to a fixed endpoint.

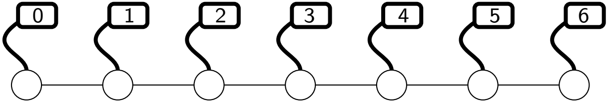


Example: how to certify that a graph is a path ?

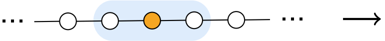


$\longrightarrow$ Path ? Cycle ?

Certificate = distance to a fixed endpoint.

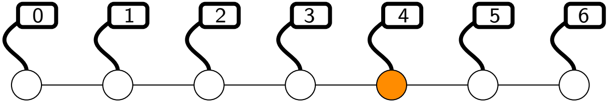


Example: how to certify that a graph is a path ?

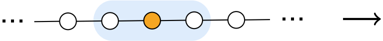


Path ? Cycle ?

Certificate = distance to a fixed endpoint.

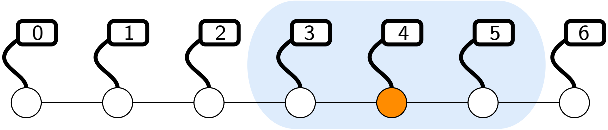


Example: how to certify that a graph is a path ?

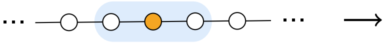


Path ? Cycle ?

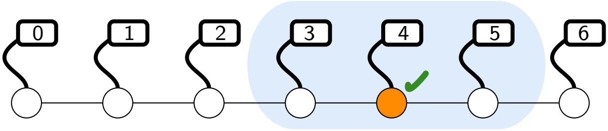
Certificate = distance to a fixed endpoint.



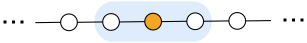
Example: how to certify that a graph is a path ?



Certificate = distance to a fixed endpoint.

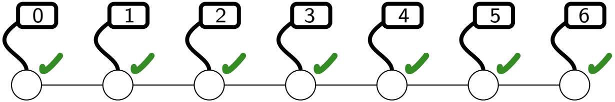


Example: how to certify that a graph is a path ?

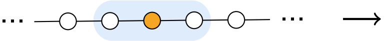


$\longrightarrow$ Path ? Cycle ?

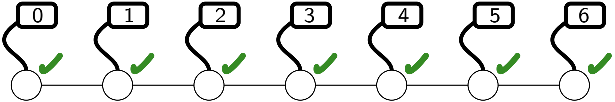
Certificate = distance to a fixed endpoint.

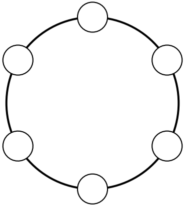


Example: how to certify that a graph is a path ?

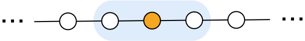


Certificate = distance to a fixed endpoint.



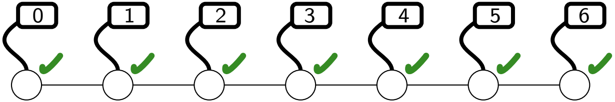


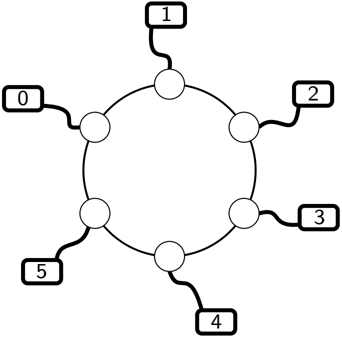
Example: how to certify that a graph is a path ?



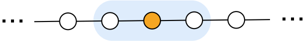
$\longrightarrow$ Path ? Cycle ?

Certificate = distance to a fixed endpoint.



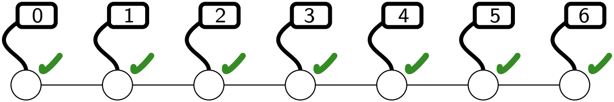


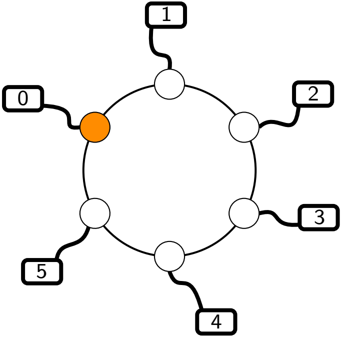
Example: how to certify that a graph is a path ?



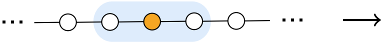
$\longrightarrow$ Path ? Cycle ?

Certificate = distance to a fixed endpoint.

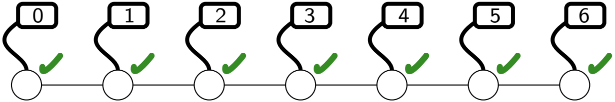


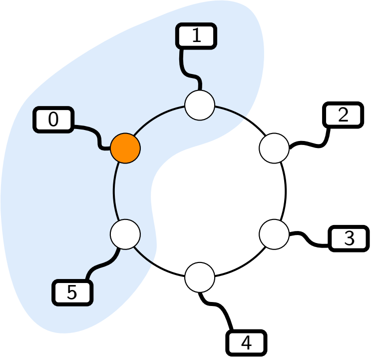


Example: how to certify that a graph is a path ?

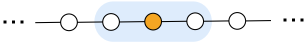


Certificate = distance to a fixed endpoint.



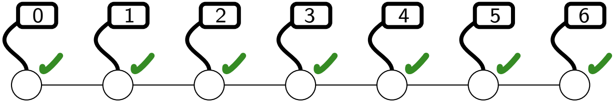


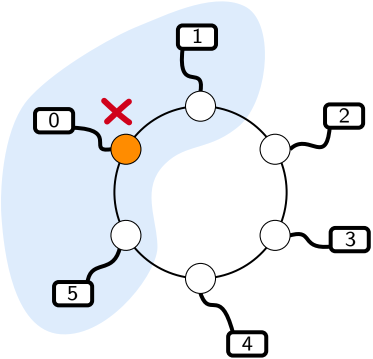
Example: how to certify that a graph is a path ?



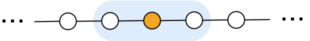
$\longrightarrow$ Path ? Cycle ?

Certificate = distance to a fixed endpoint.

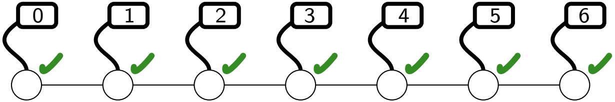


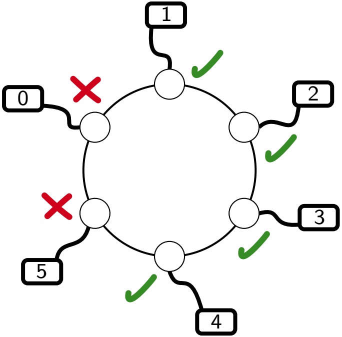


Example: how to certify that a graph is a path ?

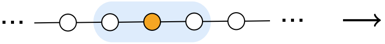


Certificate = distance to a fixed endpoint.



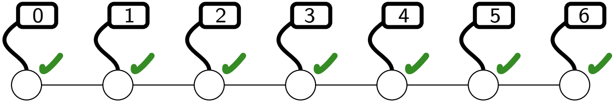


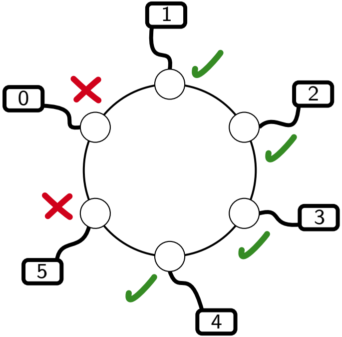
Example: how to certify that a graph is a path ?



Path ? Cycle ?

Certificate = distance to a fixed endpoint.





Size of the certificates: $O(\log n)$

What should be the minimum size of the certificates ?

What should be the minimum size of the certificates ?

Usual parameter: n (number of vertices in the graph)

What should be the minimum size of the certificates ?

Usual parameter: n (number of vertices in the graph)

Theorem

Any property can be certified with certificates of size $O(n^2)$.

idea: write the full graph in the certificate of each vertex

What should be the minimum size of the certificates ?

Usual parameter: n (number of vertices in the graph)

Theorem

Any property can be certified with certificates of size $O(n^2)$.

idea: write the full graph in the certificate of each vertex

Typical size of certificates :

$\Theta(poly(n))$	$\Theta(\log n)$	$O(1)$

What should be the minimum size of the certificates ?

Usual parameter: n (number of vertices in the graph)

Theorem

Any property can be certified with certificates of size $O(n^2)$.

idea: write the full graph in the certificate of each vertex

Typical size of certificates :

$\Theta(poly(n))$		$\Theta(\log n)$	$O(1)$
$\tilde{\Theta}(n^2)$	$\tilde{\Theta}(n)$		
▪ Non-3-colorability ▪ Non-trivial automorphism	▪ Unit-disk graphs ▪ H-freeness (for some graphs H)		

What should be the minimum size of the certificates ?

Usual parameter: n (number of vertices in the graph)

Theorem

Any property can be certified with certificates of size $O(n^2)$.

idea: write the full graph in the certificate of each vertex

Typical size of certificates :

$\Theta(poly(n))$		$\Theta(\log n)$	$O(1)$
$\tilde{\Theta}(n^2)$	$\tilde{\Theta}(n)$	▪ Paths ▪ Trees ▪ Odd number of vertices ▪ Planar graphs	
▪ Non-3-colorability ▪ Non-trivial automorphism	▪ Unit-disk graphs ▪ H-freeness (for some graphs H)		

What should be the minimum size of the certificates ?

Usual parameter: n (number of vertices in the graph)

Theorem

Any property can be certified with certificates of size $O(n^2)$.

↪ idea: write the full graph in the certificate of each vertex

Typical size of certificates :

$\Theta(poly(n))$		$\Theta(\log n)$	$O(1)$
$\tilde{\Theta}(n^2)$	$\tilde{\Theta}(n)$	▪ Paths ▪ Trees ▪ Odd number of vertices ▪ Planar graphs	▪ k-colorability ▪ Pointed vertices are dominant at distance t
▪ Non-3-colorability ▪ Non-trivial automorphism	▪ Unit-disk graphs ▪ H-freeness (for some graphs H)		

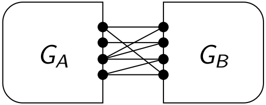
Local reductions: motivations

Local reductions: motivations

- Lower bounds: usually, harder to prove than upper bounds.

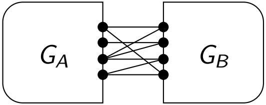
Local reductions: motivations

- Lower bounds: usually, harder to prove than upper bounds.
- Classical tool to prove lower bounds: communication complexity.



Local reductions: motivations

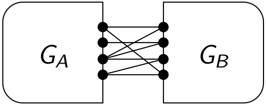
- Lower bounds: usually, harder to prove than upper bounds.
- Classical tool to prove lower bounds: communication complexity.



- A property difficult to certify: $\tilde{\Omega}(n^2)$ for non-3-colorability [Göös, Suomela]

Local reductions: motivations

- Lower bounds: usually, harder to prove than upper bounds.
- Classical tool to prove lower bounds: communication complexity.

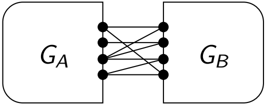


- A property difficult to certify: $\tilde{\Omega}(n^2)$ for non-3-colorability [Göös, Suomela]

  complicated proof based on communication complexity, using many gadgets

Local reductions: motivations

- Lower bounds: usually, harder to prove than upper bounds.
- Classical tool to prove lower bounds: communication complexity.



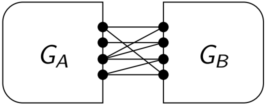
- A property difficult to certify: $\tilde{\Omega}(n^2)$ for non-3-colorability [Göös, Suomela]

  complicated proof based on communication complexity, using many gadgets

- **Our contribution:** use of **hardness reductions** in local certification.

Local reductions: motivations

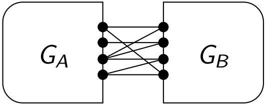
- Lower bounds: usually, harder to prove than upper bounds.
- Classical tool to prove lower bounds: communication complexity.



- A property difficult to certify: $\tilde{\Omega}(n^2)$ for non-3-colorability [Göös, Suomela]
  complicated proof based on communication complexity, using many gadgets
- **Our contribution:** use of **hardness reductions** in local certification.
- To prove a lower bound, instead of using communication complexity, we can just **transfer this lower bound using a reduction**.

Local reductions: motivations

- Lower bounds: usually, harder to prove than upper bounds.
- Classical tool to prove lower bounds: communication complexity.



- A property difficult to certify: $\tilde{\Omega}(n^2)$ for non-3-colorability [Göös, Suomela]

  complicated proof based on communication complexity, using many gadgets

- **Our contribution:** use of **hardness reductions** in local certification.
- To prove a lower bound, instead of using communication complexity, we can just **transfer this lower bound using a reduction**.
- Such a reduction has to be **local**: we identified the requirements that it should satisfy.

How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

Main idea: use a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ to show that:

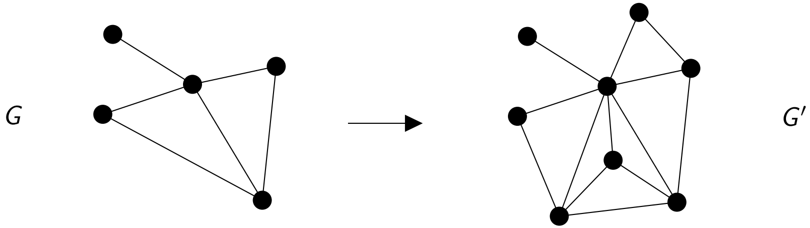
An efficient certification for $\mathcal{P}'$ can be transformed into an efficient certification for $\mathcal{P}$.

How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

Main idea: use a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ to show that:

An efficient certification for $\mathcal{P}'$ can be transformed into an efficient certification for $\mathcal{P}$.

G satisfies $\mathcal{P} \iff G'$ satisfies $\mathcal{P}'$



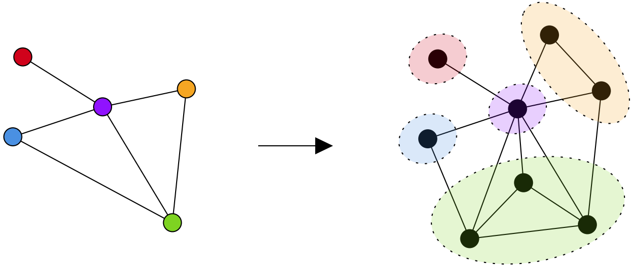
How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

Main idea: use a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ to show that:

An efficient certification for $\mathcal{P}'$ can be transformed into an efficient certification for $\mathcal{P}$.

$$G \text{ satisfies } \mathcal{P} \iff G' \text{ satisfies } \mathcal{P}'$$

G



G'

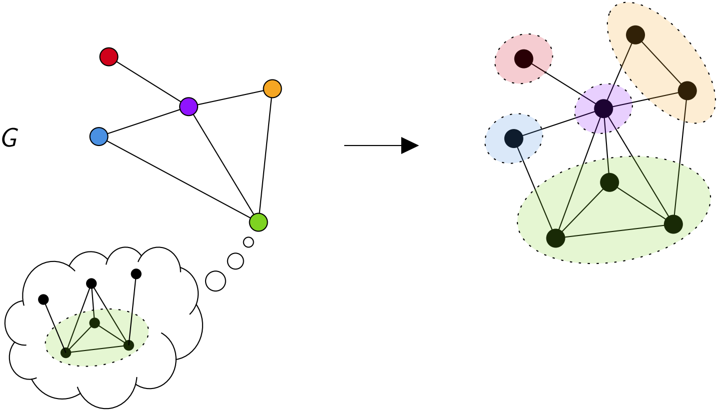
How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

Main idea: use a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ to show that:

An efficient certification for $\mathcal{P}'$ can be transformed into an efficient certification for $\mathcal{P}$.

G satisfies $\mathcal{P} \iff G'$ satisfies $\mathcal{P}'$

G



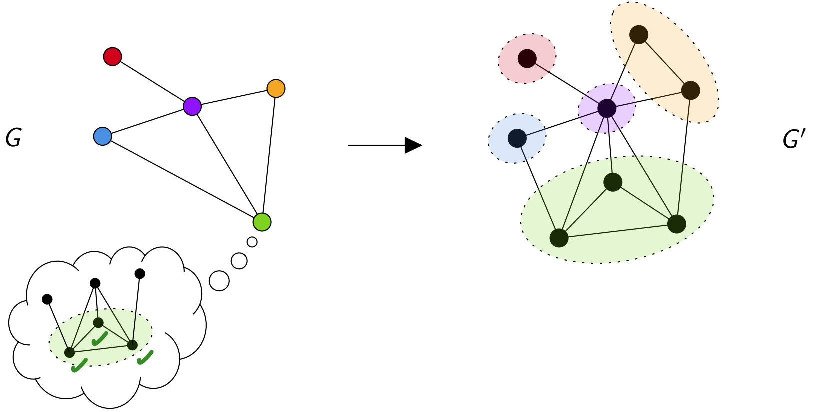
G'

How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

Main idea: use a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ to show that:

An efficient certification for $\mathcal{P}'$ can be transformed into an efficient certification for $\mathcal{P}$.

$$G \text{ satisfies } \mathcal{P} \iff G' \text{ satisfies } \mathcal{P}'$$

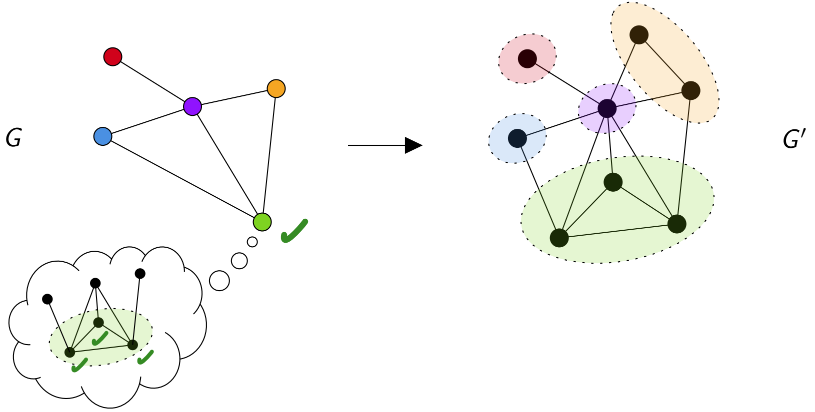


How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

Main idea: use a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ to show that:

An efficient certification for $\mathcal{P}'$ can be transformed into an efficient certification for $\mathcal{P}$.

$$G \text{ satisfies } \mathcal{P} \iff G' \text{ satisfies } \mathcal{P}'$$

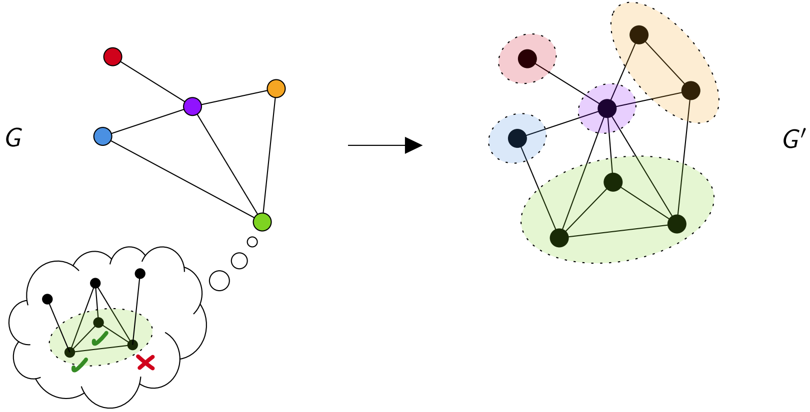


How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

Main idea: use a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ to show that:

An efficient certification for $\mathcal{P}'$ can be transformed into an efficient certification for $\mathcal{P}$.

G satisfies $\mathcal{P} \iff G'$ satisfies $\mathcal{P}'$

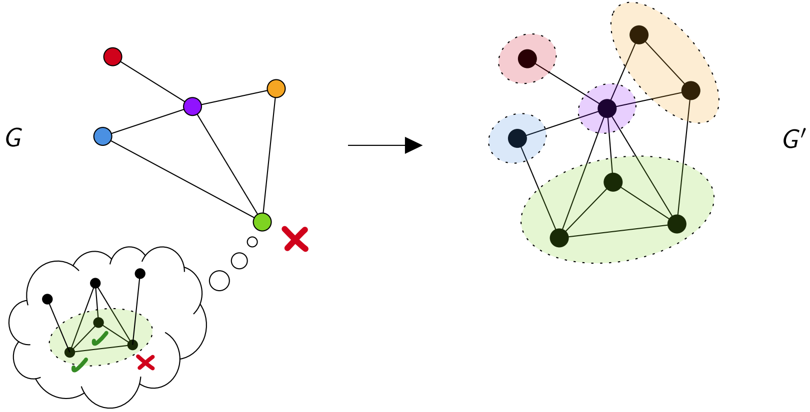


How to transfer a lower bound from $\mathcal{P}$ to $\mathcal{P}'$?

Main idea: use a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ to show that:

An efficient certification for $\mathcal{P}'$ can be transformed into an efficient certification for $\mathcal{P}$.

G satisfies $\mathcal{P} \iff G'$ satisfies $\mathcal{P}'$



Example: reduction from non-3-colorability to non-4-colorability

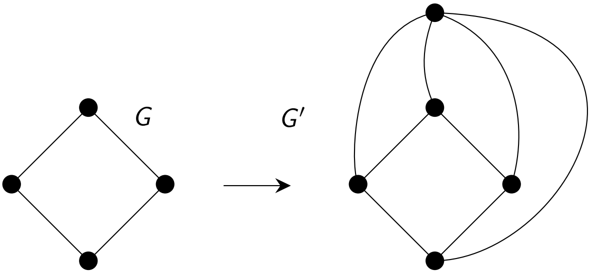
Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\iff G'$ 4-colorable

Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\iff$ G' 4-colorable	

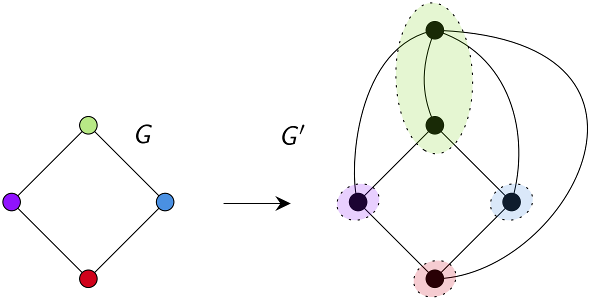
First attempt: add a universal vertex.



Example: reduction from non-3-colorability to non-4-colorability

We want:	$G \longrightarrow G'$	and	G 3-colorable $\iff$ G' 4-colorable

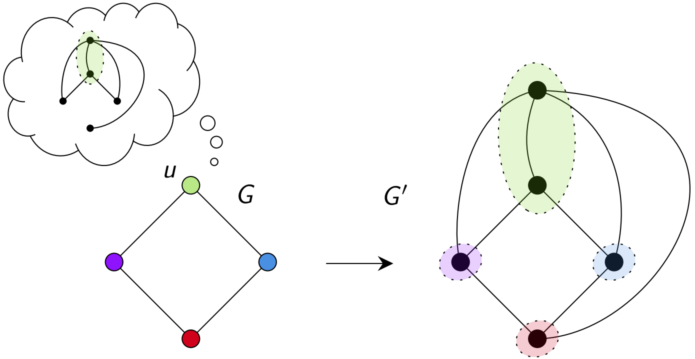
First attempt: add a universal vertex.



Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\Longleftrightarrow$ G' 4-colorable

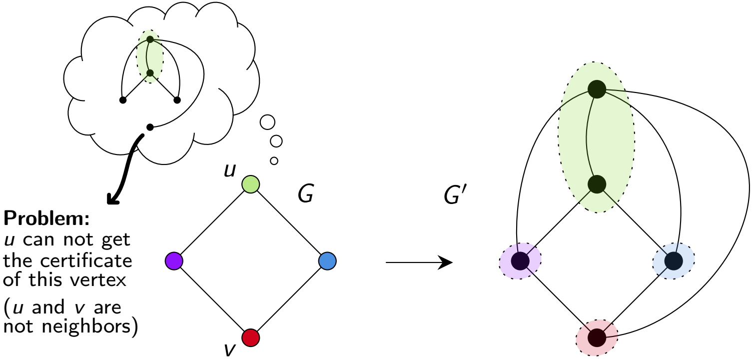
First attempt: add a universal vertex.



Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\Longleftrightarrow$ G' 4-colorable

First attempt: add a universal vertex.

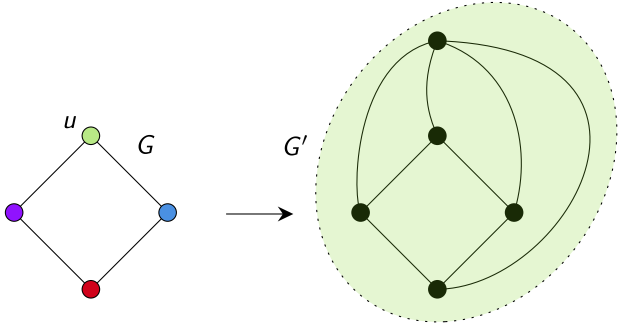


Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\Longleftrightarrow$ G' 4-colorable

First attempt: add a universal vertex.

Solution:



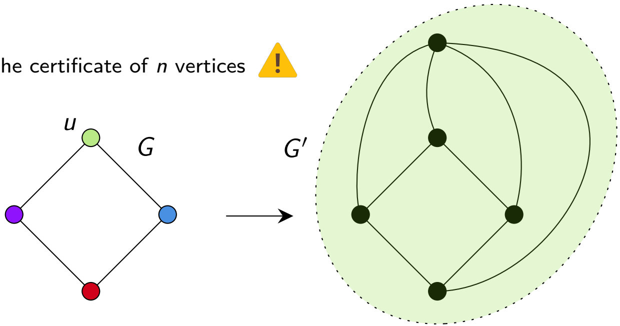
Example: reduction from non-3-colorability to non-4-colorability

We want:	$G \longrightarrow G'$	and	G 3-colorable $\iff G'$ 4-colorable

First attempt: add a universal vertex.

Solution:

⚠ u receives the certificate of n vertices ⚠



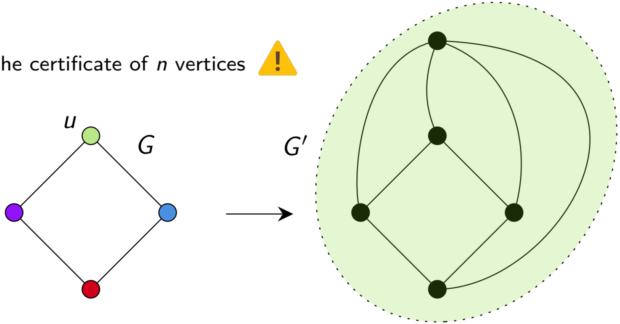
Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\iff$ G' 4-colorable

First attempt: add a universal vertex.

Solution:

u receives the certificate of n vertices



Certification of size $\Omega\left(\frac{n^2}{\log n}\right)$ for non-3-colorability $\Longrightarrow$ $\boxed{\Omega\left(\frac{n}{\log n}\right) \textbf{ for non-4-colorability}}$

Example: reduction from non-3-colorability to non-4-colorability

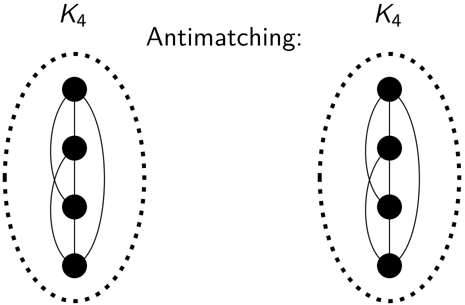
We want: $G \longrightarrow G'$ and G 3-colorable $\iff$ G' 4-colorable

Second attempt:

Example: reduction from non-3-colorability to non-4-colorability

We want:	$G \longrightarrow G'$	and	G 3-colorable $\Longleftrightarrow$ G' 4-colorable

Second attempt:

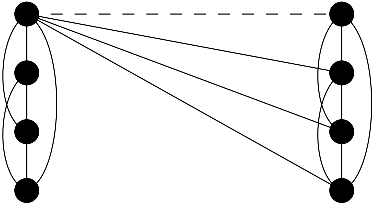


Example: reduction from non-3-colorability to non-4-colorability

We want:	$G \longrightarrow G'$	and	G 3-colorable $\iff$ G' 4-colorable

Second attempt:

Antimatching:

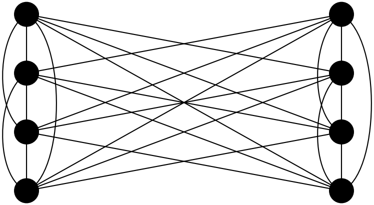


Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\Longleftrightarrow$ G' 4-colorable

Second attempt:

Antimatching:



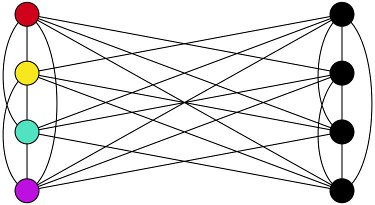
Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\iff$ G' 4-colorable

Second attempt:

Antimatching:

Coloring:



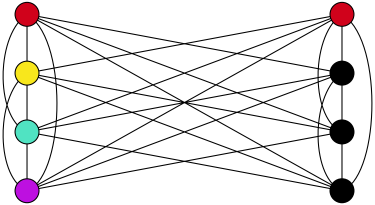
Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\iff$ G' 4-colorable

Second attempt:

Antimatching:

Coloring:



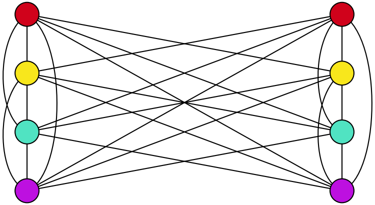
Example: reduction from non-3-colorability to non-4-colorability

We want:	$G \longrightarrow G'$	and	G 3-colorable $\iff G'$ 4-colorable

Second attempt:

Antimatching:

Coloring:



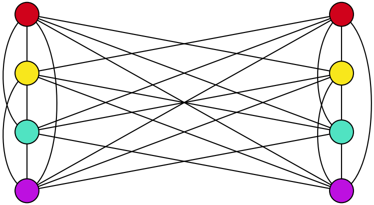
Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\Longleftrightarrow$ G' 4-colorable

Second attempt:

Antimatching:

Coloring:



Representation:

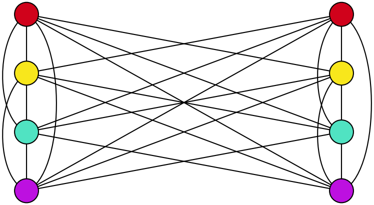
Example: reduction from non-3-colorability to non-4-colorability

We want:	$G \longrightarrow G'$	and	G 3-colorable $\Longleftrightarrow$ G' 4-colorable

<u>Second attempt</u>:

Antimatching:

Coloring:

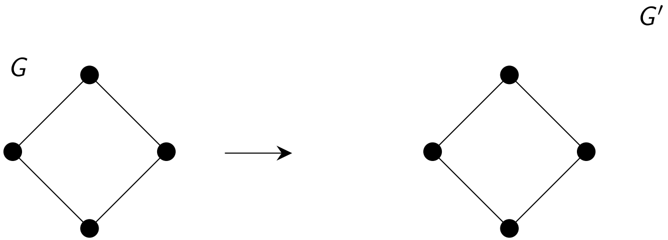


Representation:

Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\iff$ G' 4-colorable

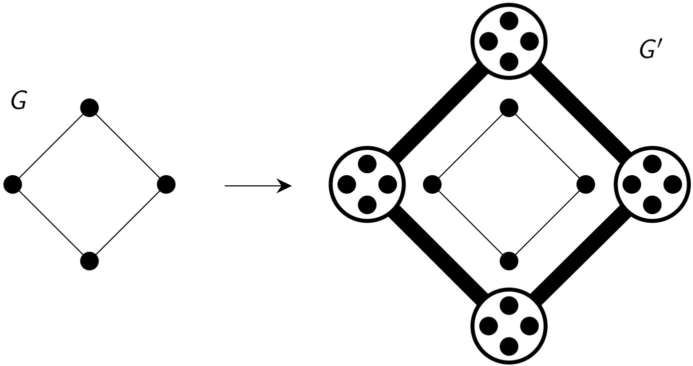
<u>Second attempt</u>:



Example: reduction from non-3-colorability to non-4-colorability

We want:	$G \longrightarrow G'$	and	G 3-colorable $\iff$ G' 4-colorable

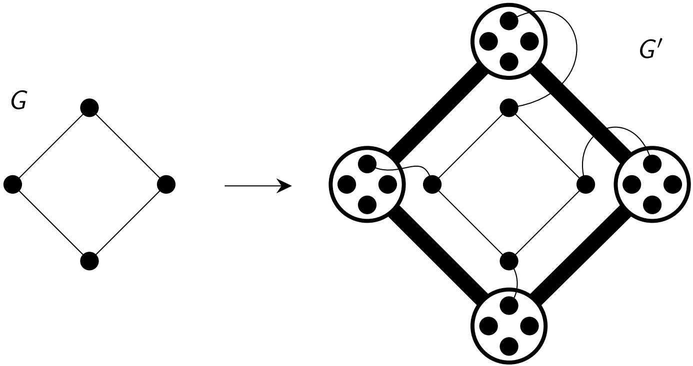
Second attempt:



Example: reduction from non-3-colorability to non-4-colorability

We want:	$G \longrightarrow G'$	and	G 3-colorable $\Longleftrightarrow$ G' 4-colorable

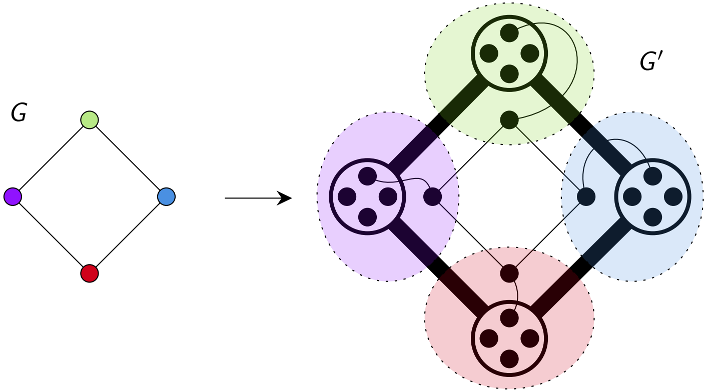
Second attempt:



Example: reduction from non-3-colorability to non-4-colorability

$$\boxed{\text{We want:} \qquad G \longrightarrow G' \qquad \text{and} \qquad G \text{ 3-colorable} \iff G' \text{ 4-colorable}}$$

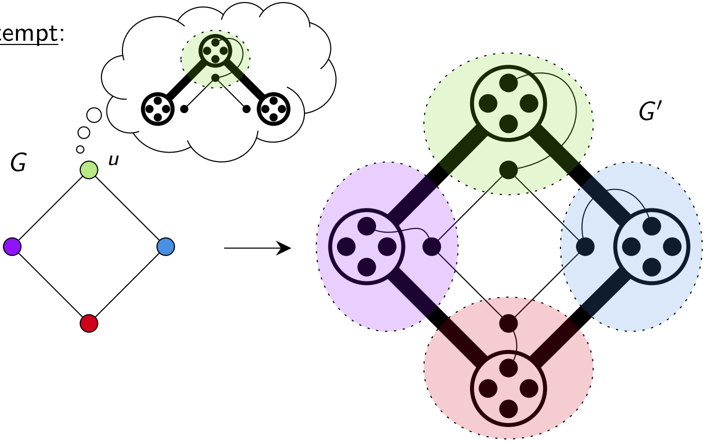
Second attempt:



Example: reduction from non-3-colorability to non-4-colorability

We want: $G \longrightarrow G'$ and G 3-colorable $\iff$ G' 4-colorable

Second attempt:

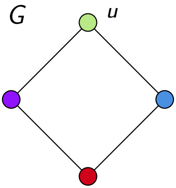


Example: reduction from non-3-colorability to non-4-colorability

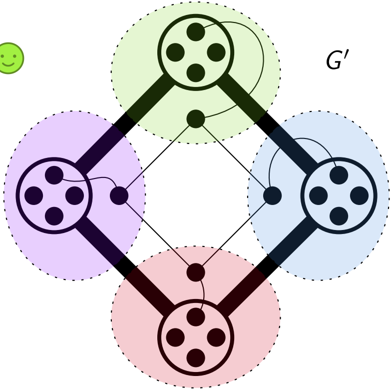
We want:	$G \longrightarrow G'$	and	G 3-colorable $\Longleftrightarrow$ G' 4-colorable

<u>Second attempt</u>:

u gets the certificate of $O(1)$ vertices



$\longrightarrow$

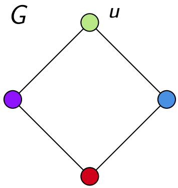


Example: reduction from non-3-colorability to non-4-colorability

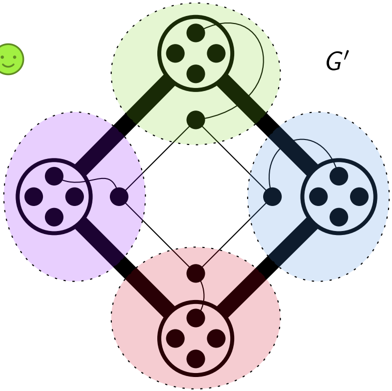
We want: $G \longrightarrow G'$ and G 3-colorable $\iff$ G' 4-colorable

Second attempt:

u gets the certificate of $O(1)$ vertices



$\longrightarrow$



Certification of size $\Omega\left(\frac{n^2}{\log n}\right)$ for non-3-colorability $\Longrightarrow$ $\boxed{\Omega\left(\frac{n^2}{\log n}\right) \textbf{ for non-4-colorability}}$

Applications of our reduction framework

Applications of our reduction framework

Theorem (Esperet, Z.)

The following properties require certificates of polynomial size:

- Non-k-colorability ($k \geqslant 4$)
- Domatic number at most k ($k \geqslant 2$)
- No cubic subgraph
- No partition into k acyclic subgraphs ($k \geqslant 3$)
- Non-existence of an edge-coloring without monochromatic triangle
- Non-hamiltonicity
- Chromatic index equal to $\Delta + 1$ (Δ = max. degree)

Applications of our reduction framework

Theorem (Esperet, Z.)

The following properties require certificates of polynomial size:

- Non-k-colorability ($k \geqslant 4$) $\tilde{\Omega}(n^2)$
- Domatic number at most k ($k \geqslant 2$) $\tilde{\Omega}(n)$
- No cubic subgraph $\tilde{\Omega}(n)$
- No partition into k acyclic subgraphs ($k \geqslant 3$) $\tilde{\Omega}(n)$
- Non-existence of an edge-coloring without monochromatic triangle $\tilde{\Omega}(n)$
- Non-hamiltonicity $\tilde{\Omega}(\sqrt{n})$
- Chromatic index equal to $\Delta + 1$ ($\Delta =$ max. degree) $\tilde{\Omega}(n)$

Final remarks:

- Most of these lower bounds are $\tilde{\Omega}(n)$

Applications of our reduction framework

Theorem (Esperet, Z.)

The following properties require certificates of polynomial size:

- Non-k-colorability ($k \geqslant 4$) $\tilde{\Omega}(n^2)$
- Domatic number at most k ($k \geqslant 2$) $\tilde{\Omega}(n)$
- No cubic subgraph $\tilde{\Omega}(n)$
- No partition into k acyclic subgraphs ($k \geqslant 3$) $\tilde{\Omega}(n)$
- Non-existence of an edge-coloring without monochromatic triangle $\tilde{\Omega}(n)$
- Non-hamiltonicity $\tilde{\Omega}(\sqrt{n})$
- Chromatic index equal to $\Delta + 1$ (Δ = max. degree) $\tilde{\Omega}(n)$

Final remarks:

- Most of these lower bounds are $\tilde{\Omega}(n)$ $\longrightarrow$ **Question:** improvement to $\tilde{\Omega}(n^2)$?

Applications of our reduction framework

Theorem (Esperet, Z.)

The following properties require certificates of polynomial size:

- Non-k-colorability ($k \geqslant 4$) $\tilde{\Omega}(n^2)$
- Domatic number at most k ($k \geqslant 2$) $\tilde{\Omega}(n)$
- No cubic subgraph $\tilde{\Omega}(n)$
- No partition into k acyclic subgraphs ($k \geqslant 3$) $\tilde{\Omega}(n)$
- Non-existence of an edge-coloring without monochromatic triangle $\tilde{\Omega}(n)$
- Non-hamiltonicity $\tilde{\Omega}(\sqrt{n})$
- Chromatic index equal to $\Delta + 1$ ($\Delta =$ max. degree) $\tilde{\Omega}(n)$

Final remarks:

- Most of these lower bounds are $\tilde{\Omega}(n)$ $\longrightarrow$ **Question:** improvement to $\tilde{\Omega}\left(n^2\right)$?
- Most of these lower bounds hold for bounded-degree graphs

Applications of our reduction framework

Theorem (Esperet, Z.)

The following properties require certificates of polynomial size:

- Non-k-colorability ($k \geqslant 4$) $\tilde{\Omega}(n^2)$
- Domatic number at most k ($k \geqslant 2$) $\tilde{\Omega}(n)$
- No cubic subgraph $\tilde{\Omega}(n)$
- No partition into k acyclic subgraphs ($k \geqslant 3$) $\tilde{\Omega}(n)$
- Non-existence of an edge-coloring without monochromatic triangle $\tilde{\Omega}(n)$
- Non-hamiltonicity $\tilde{\Omega}(\sqrt{n})$
- Chromatic index equal to $\Delta + 1$ (Δ = max. degree) $\tilde{\Omega}(n)$

Final remarks:

- Most of these lower bounds are $\tilde{\Omega}(n)$ $\longrightarrow$ **Question:** improvement to $\tilde{\Omega}(n^2)$?
- Most of these lower bounds hold for bounded-degree graphs
- All these properties are coNP-hard.

Applications of our reduction framework

Theorem (Esperet, Z.)

The following properties require certificates of polynomial size:

- Non-k-colorability ($k \geqslant 4$) $\tilde{\Omega}(n^2)$
- Domatic number at most k ($k \geqslant 2$) $\tilde{\Omega}(n)$
- No cubic subgraph $\tilde{\Omega}(n)$
- No partition into k acyclic subgraphs ($k \geqslant 3$) $\tilde{\Omega}(n)$
- Non-existence of an edge-coloring without monochromatic triangle $\tilde{\Omega}(n)$
- Non-hamiltonicity $\tilde{\Omega}(\sqrt{n})$
- Chromatic index equal to $\Delta + 1$ ($\Delta =$ max. degree) $\tilde{\Omega}(n)$

Final remarks:

- Most of these lower bounds are $\tilde{\Omega}(n)$ $\longrightarrow$ **Question:** improvement to $\tilde{\Omega}\left(n^2\right)$?
- Most of these lower bounds hold for bounded-degree graphs
- All these properties are coNP-hard. But:
  - there are coNP-hard problems which can be certified with $O(\log n)$ bits

Applications of our reduction framework

Theorem (Esperet, Z.)

The following properties require certificates of polynomial size:

- Non-k-colorability ($k \geqslant 4$) $\tilde{\Omega}(n^2)$
- Domatic number at most k ($k \geqslant 2$) $\tilde{\Omega}(n)$
- No cubic subgraph $\tilde{\Omega}(n)$
- No partition into k acyclic subgraphs ($k \geqslant 3$) $\tilde{\Omega}(n)$
- Non-existence of an edge-coloring without monochromatic triangle $\tilde{\Omega}(n)$
- Non-hamiltonicity $\tilde{\Omega}(\sqrt{n})$
- Chromatic index equal to $\Delta + 1$ ($\Delta =$ max. degree) $\tilde{\Omega}(n)$

Final remarks:

- Most of these lower bounds are $\tilde{\Omega}(n)$ $\longrightarrow$ **Question:** improvement to $\tilde{\Omega}(n^2)$?
- Most of these lower bounds hold for bounded-degree graphs
- All these properties are coNP-hard. But:
  - there are coNP-hard problems which can be certified with $O(\log n)$ bits
  - our reduction framework also applies to properties that are not coNP-hard (e.g. H-freeness for a fixed graph H)

Thanks for your attention !