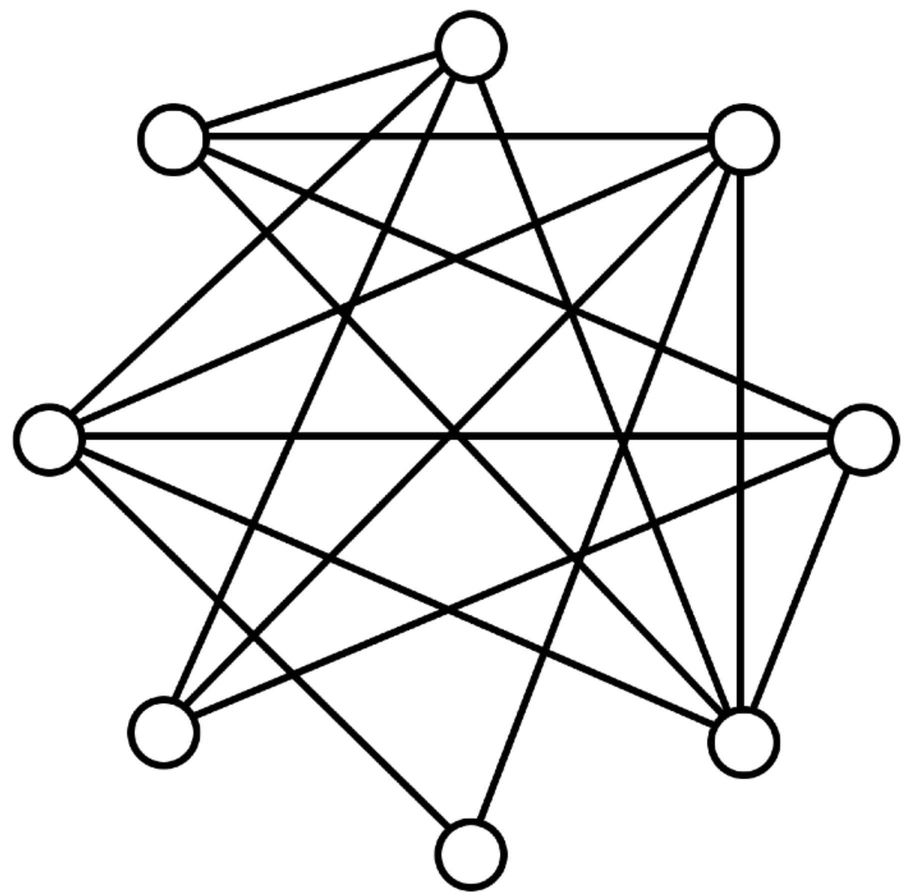
Container Lemmas and a Characterization of Testing Hereditary Graph Properties

Gaia Carenini (University of Cambridge), Cameron Seth, Yuichi Yoshida (National Institute of Informatics)

August 14 2026

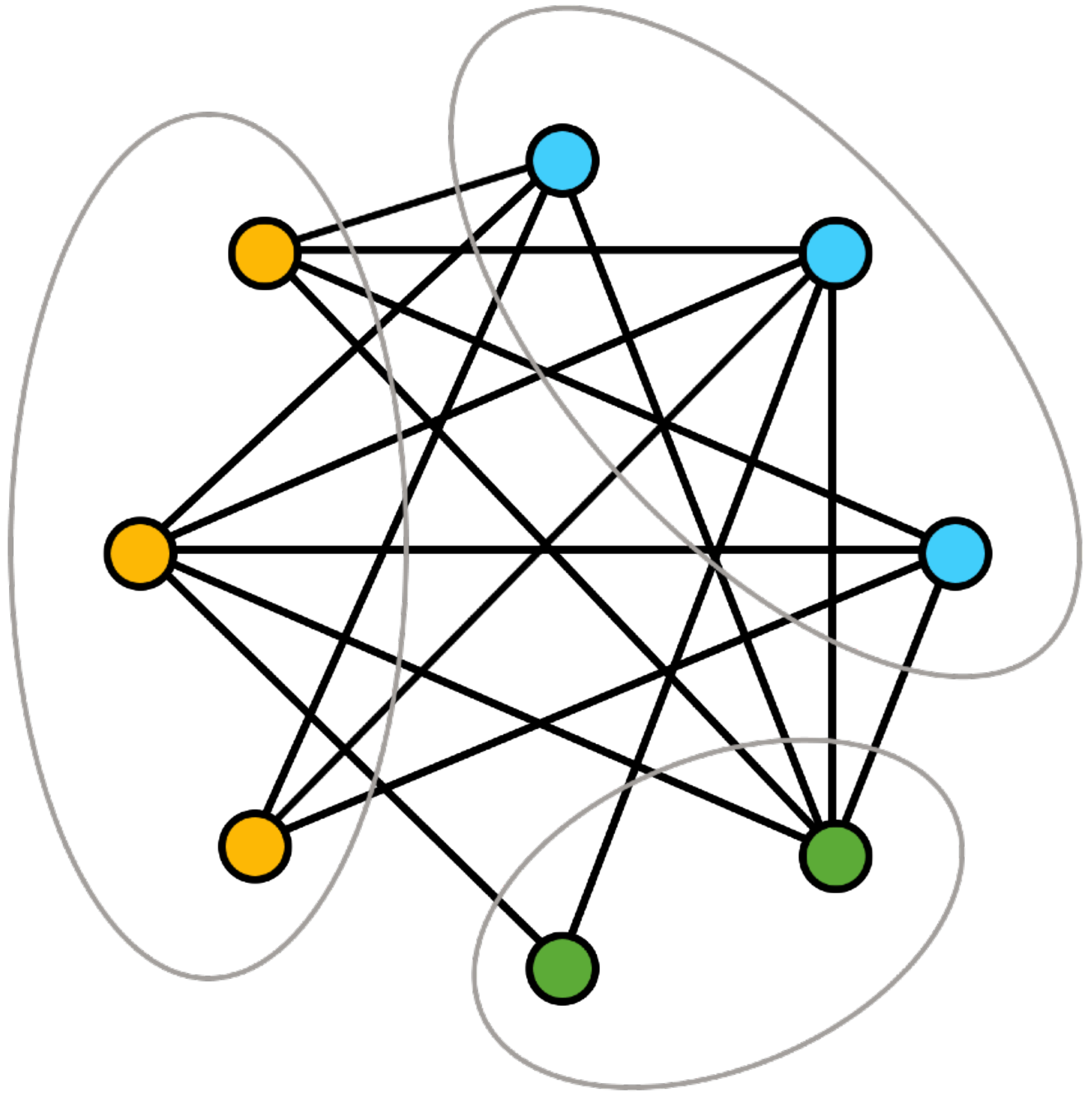
3-Colorability Problem

Given a graph G on n vertices, is it 3-colorable?



3-Colorability Problem

Given a graph G on n vertices, is it 3-colorable?



Testing 3-Colorability

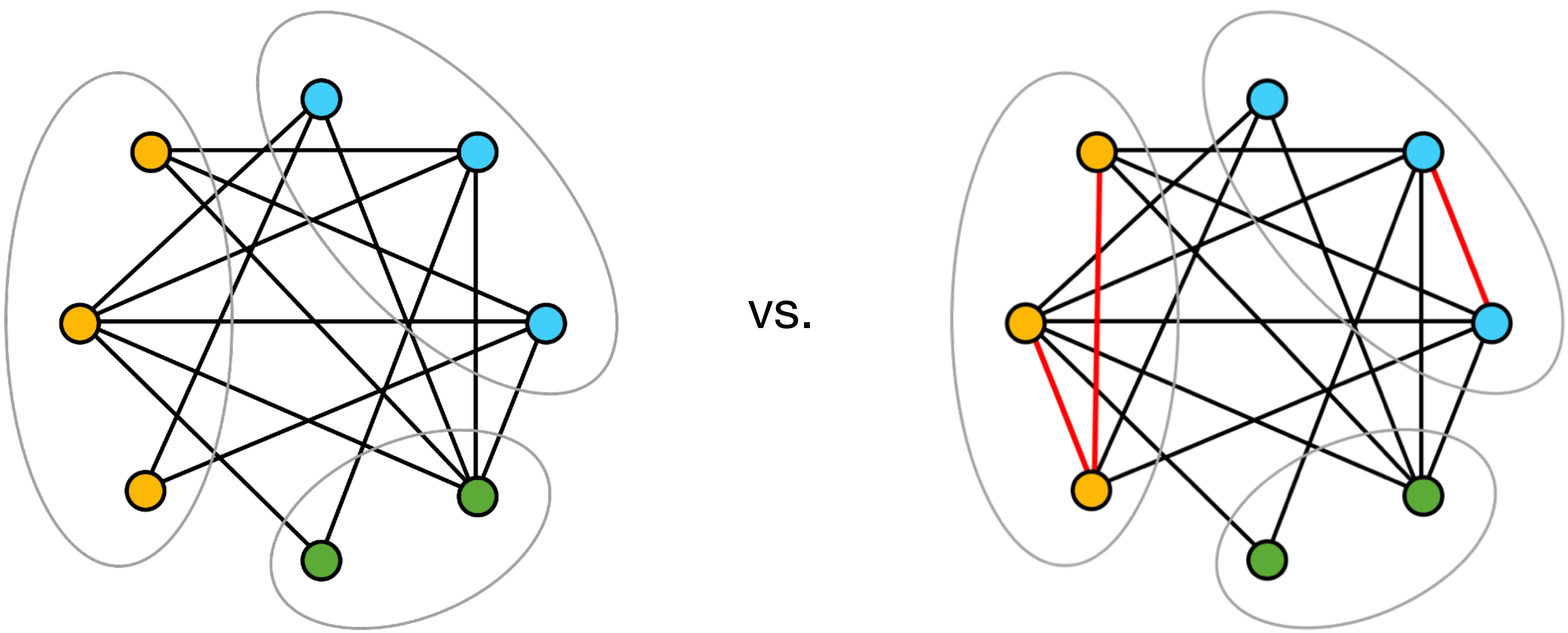
Given a graph G on n vertices, distinguish between the following cases:

1. G is 3-colorable
2. At least ϵn^2 edges of G must be modified to make G be 3-colorable

Testing 3-Colorability

Given a graph G on n vertices, distinguish between the following cases:

1. G is 3-colorable
2. At least ϵn^2 edges of G must be modified to make G be 3-colorable



Testing 3-Colorability

Given a graph G on n vertices, distinguish between the following cases:

1. G is 3-colorable
2. At least ϵn^2 edges of G must be modified to make G be 3-colorable

Theorem: Inspecting a random induced subgraph on $\mathrm{poly}(1/\epsilon)$ vertices is sufficient to distinguish between case 1 and 2 (whp).

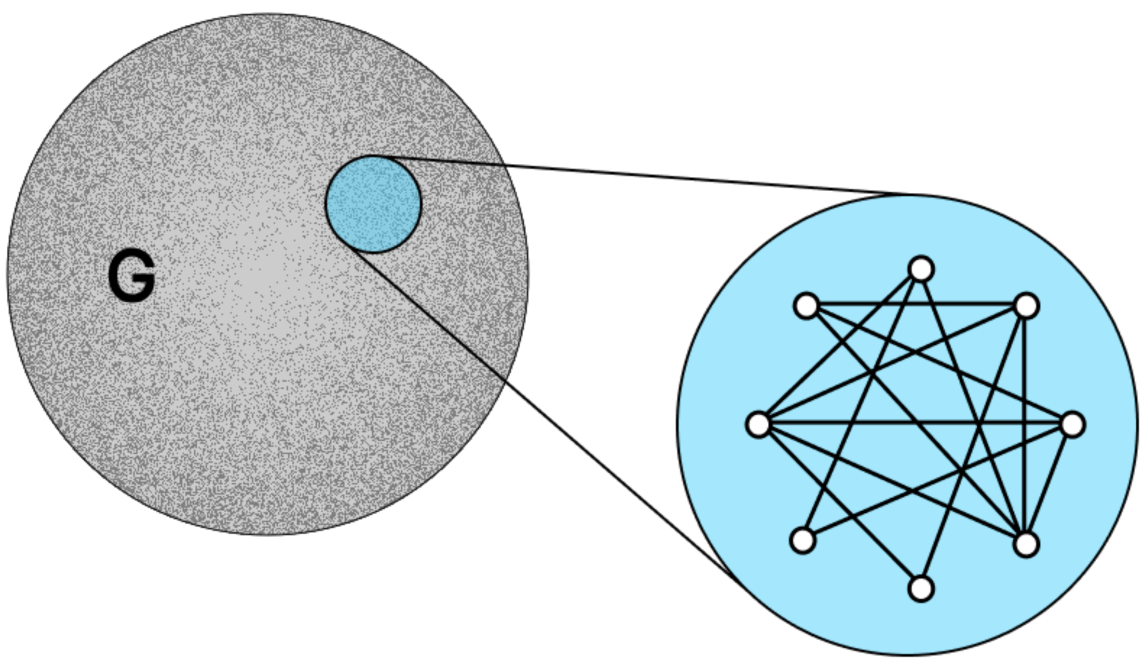
[Goldreich, Goldwasser, Ron '98]

Algorithm: On input graph $G = (V, E)$, take a random set $S \subseteq V$ of $\mathrm{poly}(1/\epsilon)$ vertices, accept iff $G[S]$ is 3-colorable.

Definitions

An ϵ-tester for Π is an algorithm that samples a set S of s random vertices, examines the induced subgraph $G[S]$, and distinguishes between the cases (with probability at least $2/3$):

1. G has the property Π, and
2. At least ϵn^2 edges of G must be modified to make G have Π (ϵ-far)

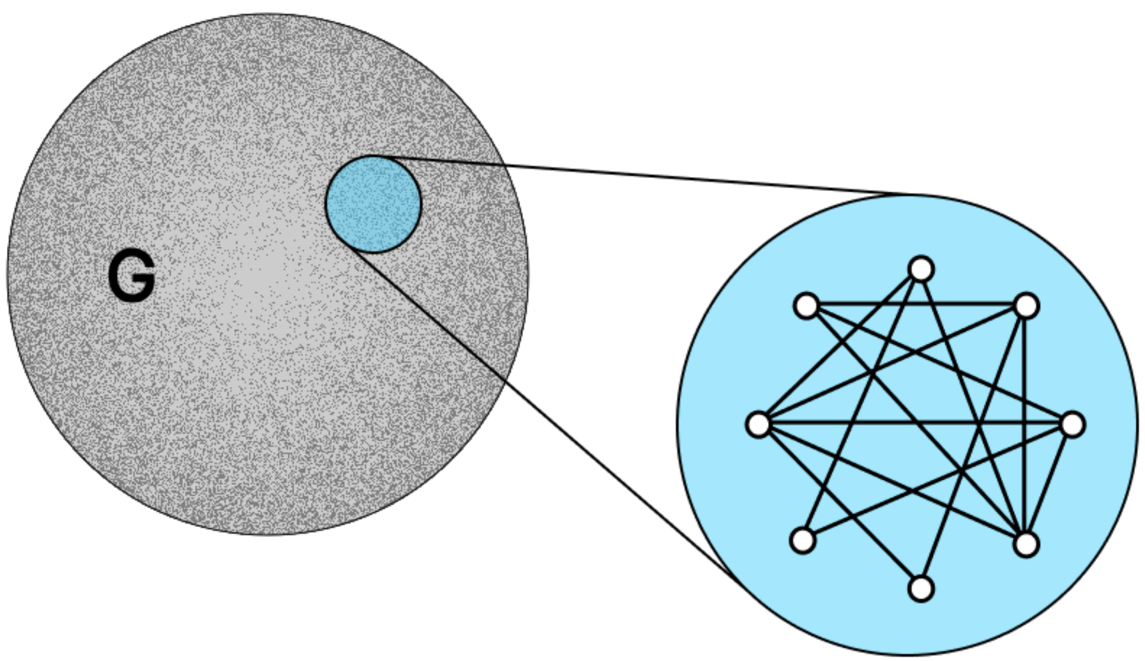


s is the sample complexity of the tester.

Definitions

An ϵ-tester for Π is an algorithm that samples a set S of s random vertices, examines the induced subgraph $G[S]$, and distinguishes between the cases (with probability at least $2/3$):

1. G has the property Π, and
2. At least ϵn^2 edges of G must be modified to make G have Π (ϵ-far)



s is the sample complexity of the tester.

Remark: An alternate measure is the edge query complexity, but the two measures are related by at most a quadratic factor.

[Goldreich, Trevisan '03]

First Results in Graph Property Testing

Theorem: There are ϵ-testers for k-colorability, ρ-clique, ρ-cut, and many other partition properties with sample complexity $\mathrm{poly}(1/\epsilon)$.

[Goldreich, Goldwasser, Ron '98]

First Results in Graph Property Testing

Theorem: There are ϵ-testers for k-colorability, ρ-clique, ρ-cut, and many other partition properties with sample complexity $\mathrm{poly}(1/\epsilon)$.

[Goldreich, Goldwasser, Ron '98]

The Characterization Question: What graph properties can be ϵ-tested with a sample complexity that only depends on ϵ (and not n)?

Answer to the Characterization Question

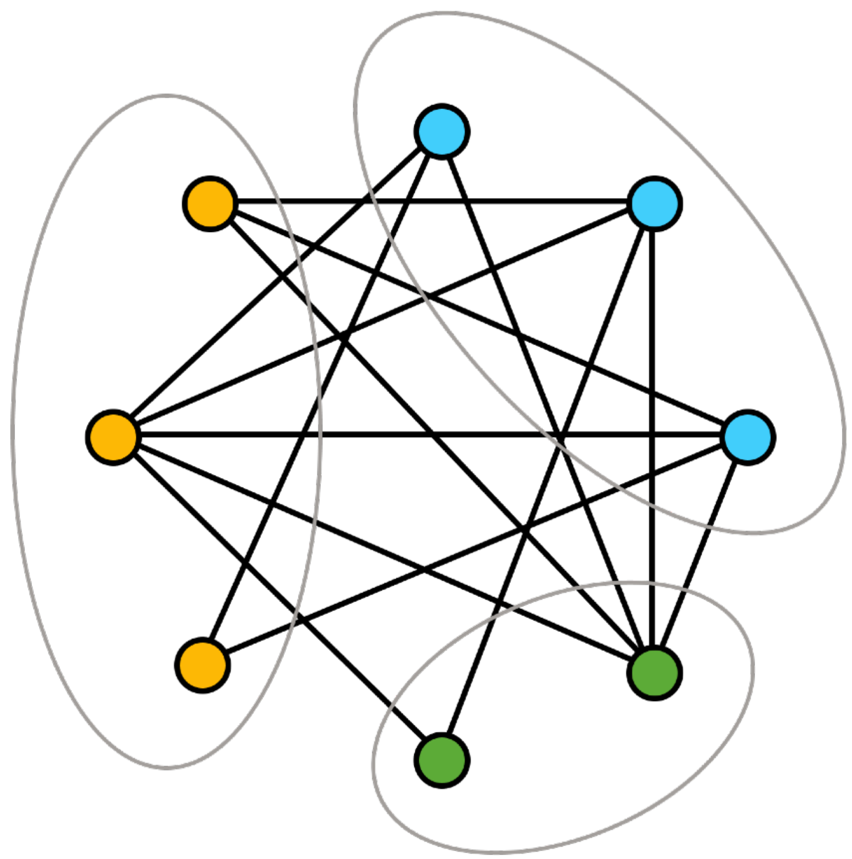
Answer to the Characterization Question

Definition: A graph property Π is hereditary if it is closed under taking induced subgraphs

Answer to the Characterization Question

Definition: A graph property Π is hereditary if it is closed under taking induced subgraphs

Examples: k-colorability (every induced subgraph of a k-colorable graph is also k-colorable)



Answer to the Characterization Question

Definition: A graph property Π is hereditary if it is closed under taking induced subgraphs

Examples: k-colorability, triangle-freeness, perfect graphs, chordal graphs

Answer to the Characterization Question

Definition: A graph property Π is hereditary if it is closed under taking induced subgraphs

Examples: k-colorability, triangle-freeness, perfect graphs, chordal graphs

one-sided error testers accept yes instances with probability 1

Theorem: Every hereditary property is ϵ-testable with one-sided error and sample complexity only depending on ϵ.

[Alon, Shapira '08]

Answer to the Characterization Question

Definition: A graph property Π is hereditary if it is closed under taking induced subgraphs

Examples: k-colorability, triangle-freeness, perfect graphs, chordal graphs

one-sided error testers accept yes instances with probability 1

Theorem: Every hereditary property is ϵ-testable with one-sided error and sample complexity only depending on ϵ.

[Alon, Shapira '08]

Question: Is there a more quantitative answer? For example, what is the class of graph properties that can be ϵ-tested with one sided error and sample complexity that is $\mathrm{poly}(1/\epsilon)$?

Recent Container Method based Results

Recently there have been a number of results using the graph container method to get new testing bounds:

Recent Container Method based Results

Recently there have been a number of results using the graph container method to get new testing bounds:

- Tight bounds on testing the ρ-independent set property

  [Blais, **S** '23]

- Improved bounds on testing k-colorability and satisfiability

  [Blais, **S** '24]

- Tight bounds on tolerant testing the ρ-independent set property

  [**S** '25]

- Improved bounds on testing the ρ-independent set property in hypergraphs

  [Grigorescu, Nasa, **S** '26]

Main Result (Very Informal)

Theorem: Let Π be a hereditary graph property. Then

Π is ϵ-testable with one sided error and sample complexity s

$$\Updownarrow$$

Π has an "ϵ-container lemma" parameterized by $\mathrm{poly}(s)$

Main Result (Very Informal)

Theorem: Let Π be a hereditary graph property. Then

Π is ϵ-testable with one sided error and sample complexity s

$$\Updownarrow$$

Π has an "ϵ-container lemma" parameterized by $\mathrm{poly}(s)$

Remarks:

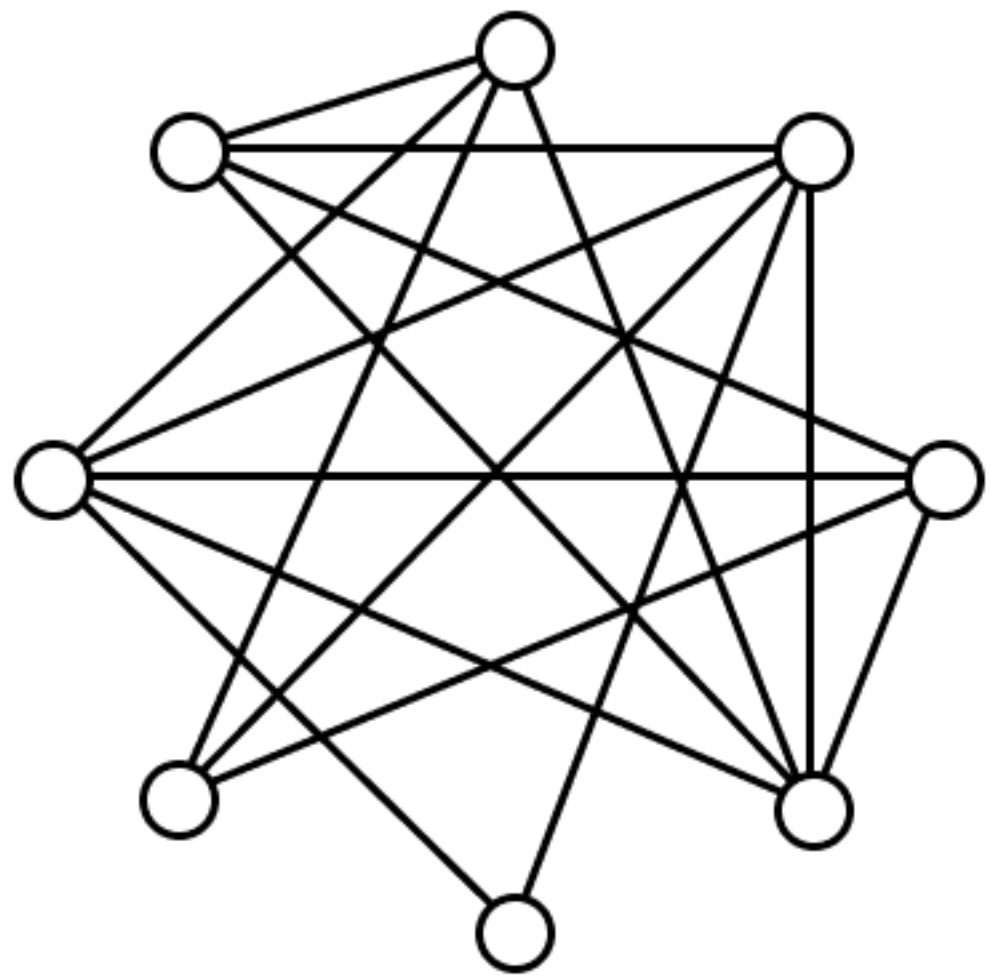
- A quantitative characterization of testability of hereditary properties
- Demonstrates that the container method captures testability of hereditary properties

Outline of Talk

- What is the container method and how can it be used for testing?
- Formally stating our theorem
- Proof sketch of our theorem
- An application to testing partitions of hereditary properties

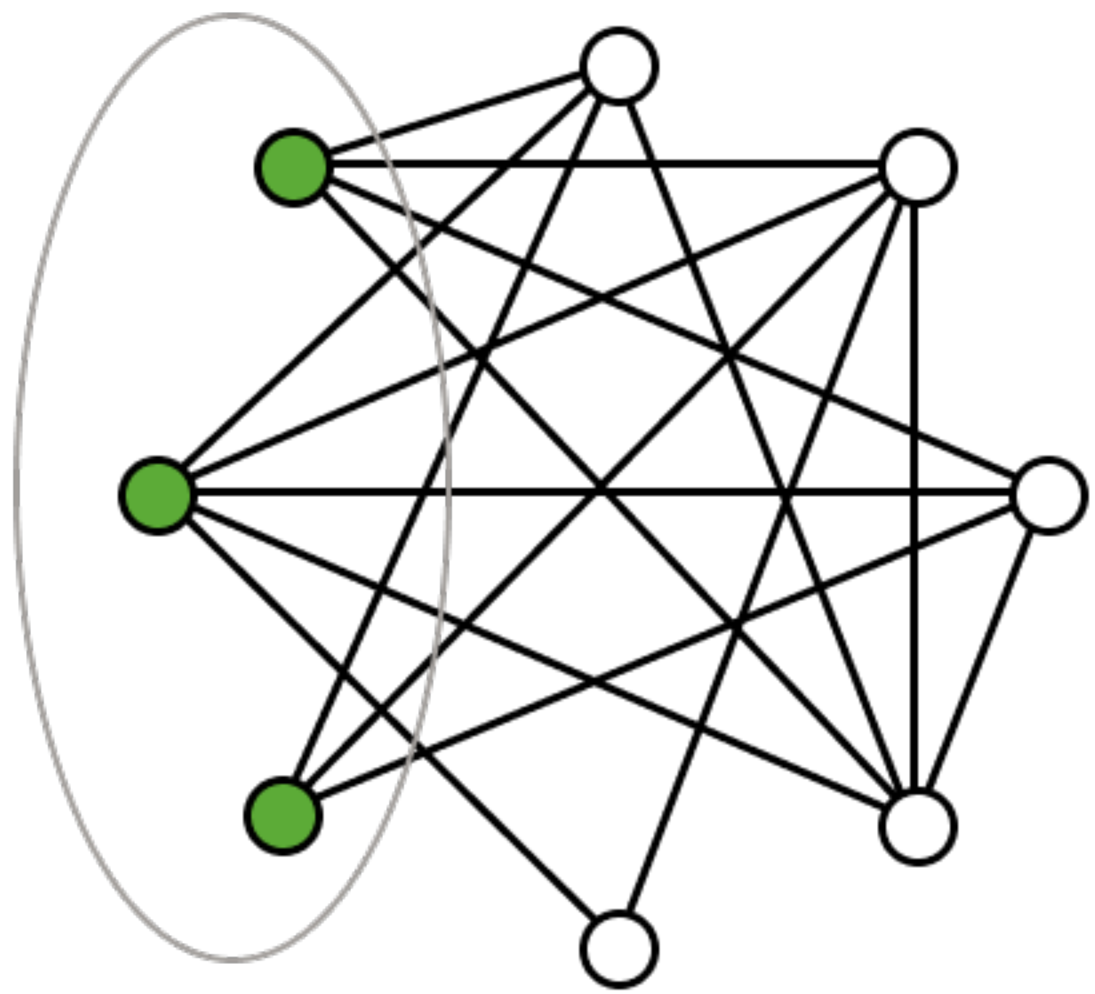
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



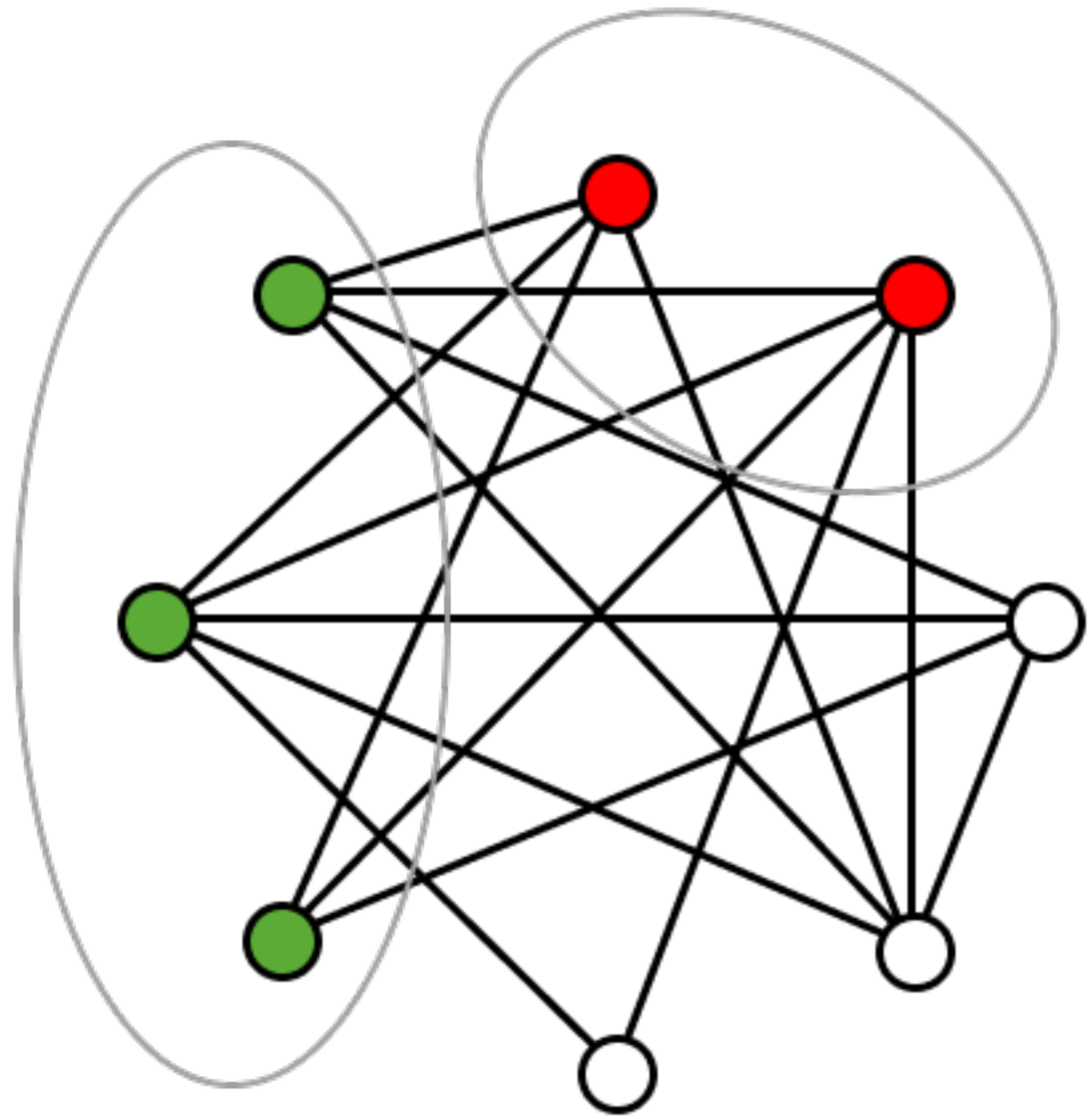
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



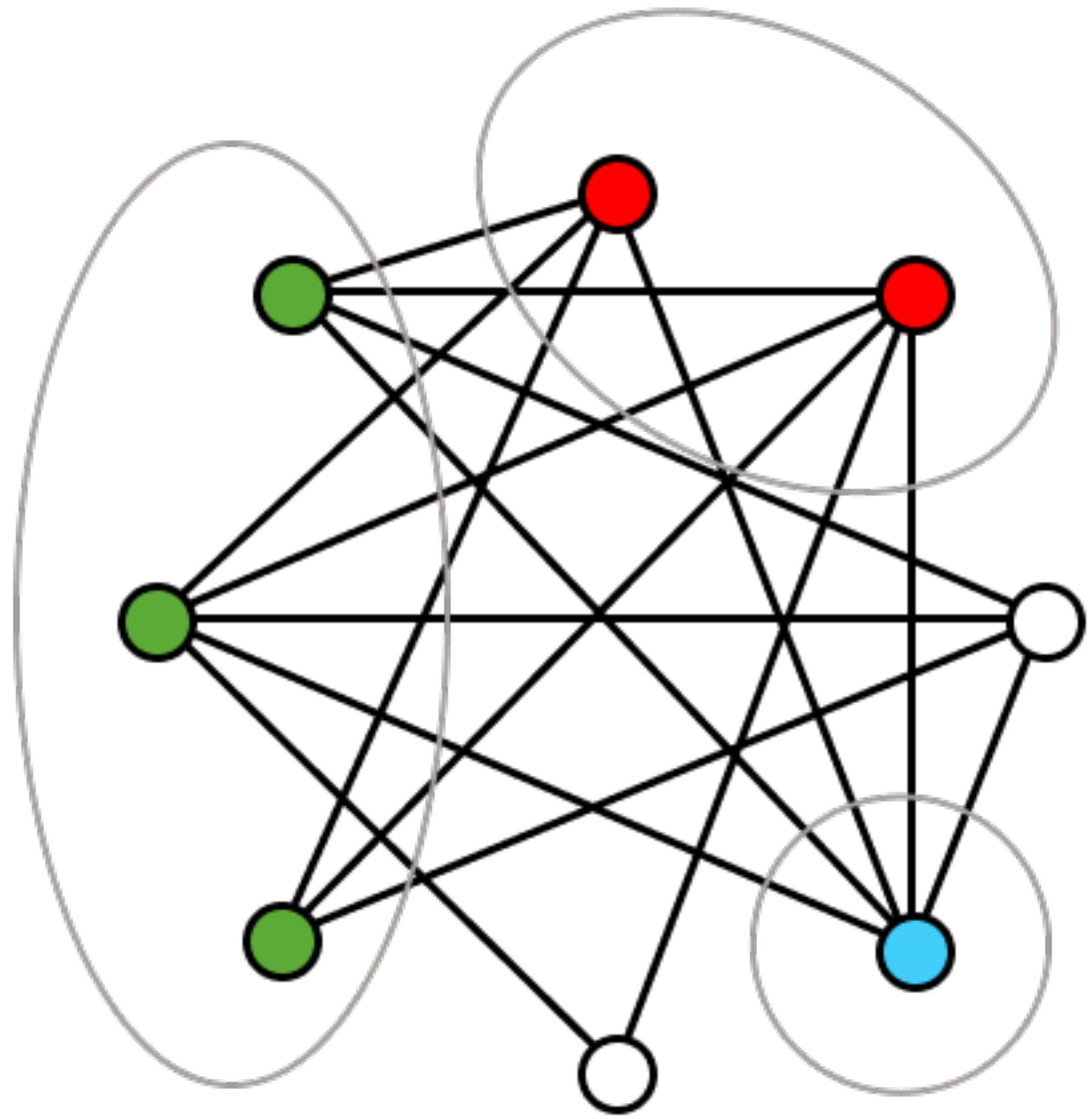
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



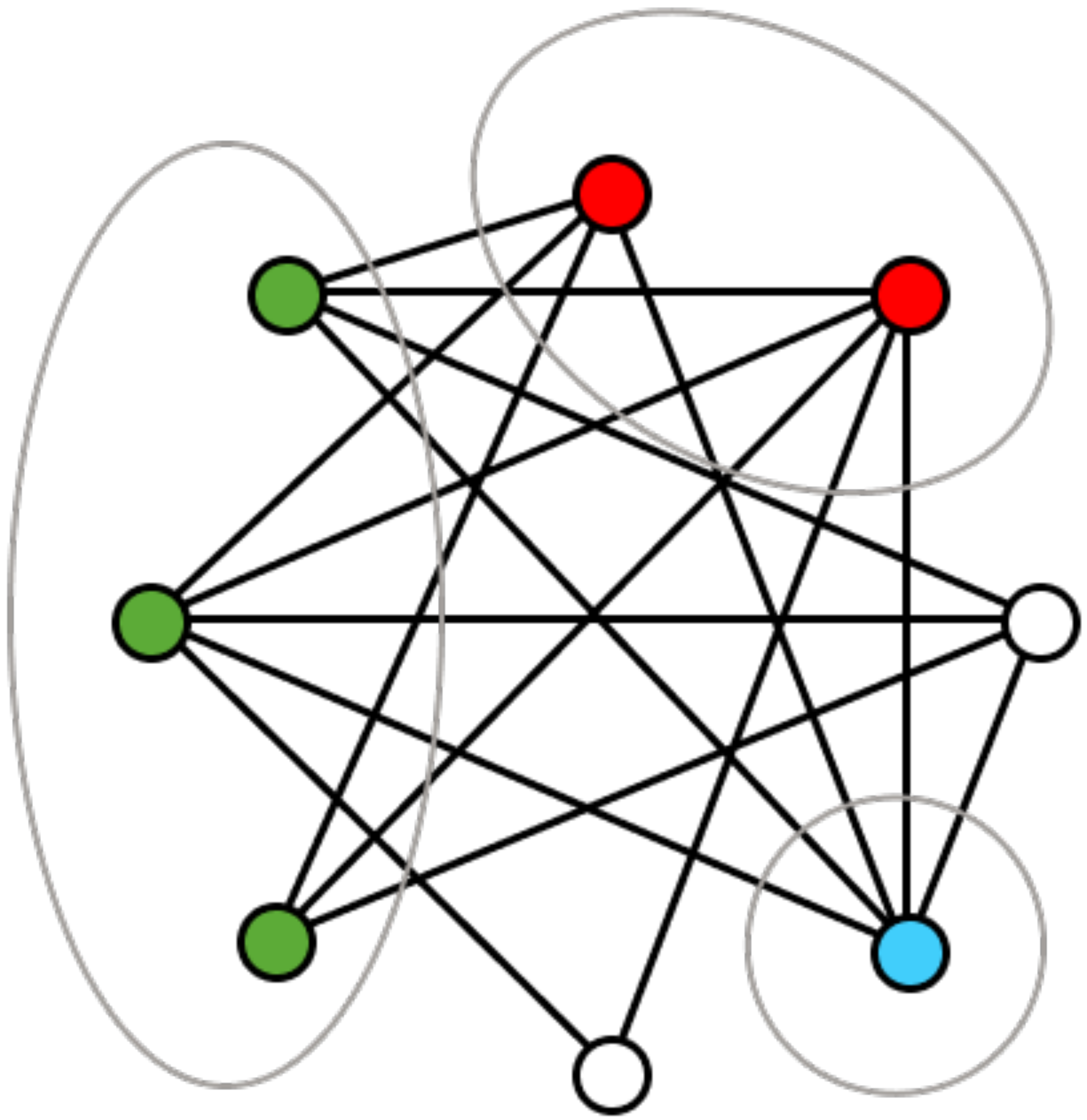
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



What is the Container Method?

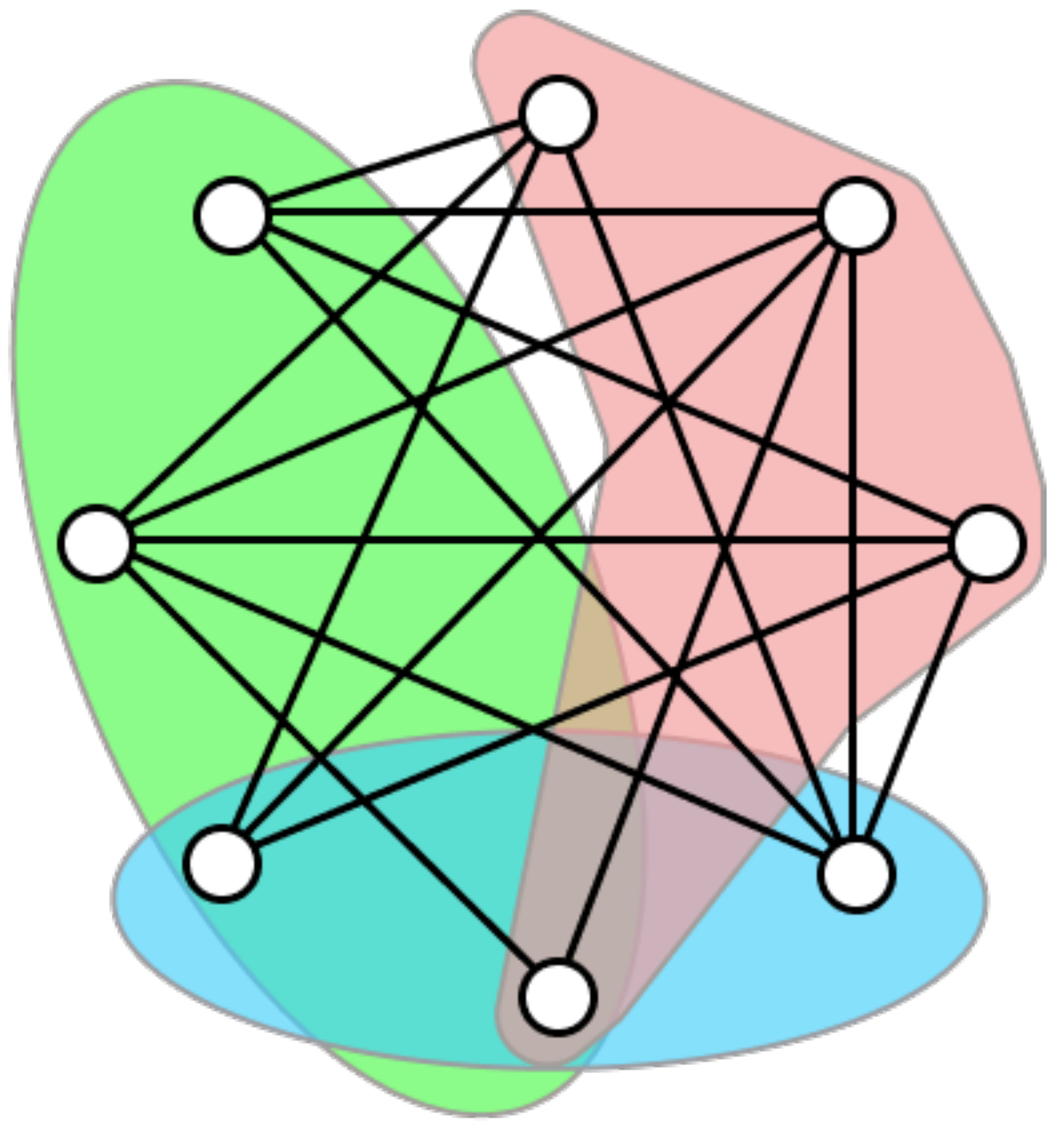
Answer: A tool for characterizing independent sets in some graphs.



Informal Idea: For any graph satisfying some “nice” conditions, all independent sets in the graph can be covered by a small number of containers (each container is a subset of vertices).

What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



Informal Idea: For any graph satisfying some “nice” conditions, all independent sets in the graph can be covered by a small number of containers (each container is a subset of vertices).

An Initial Graph Container Lemma

Lemma: For any ϵ let $G = (V, E)$ be a graph with at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

[Kleitman, Winston '82] - simple case

An Initial Graph Container Lemma

Lemma: For any ϵ let $G = (V, E)$ be a graph with at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

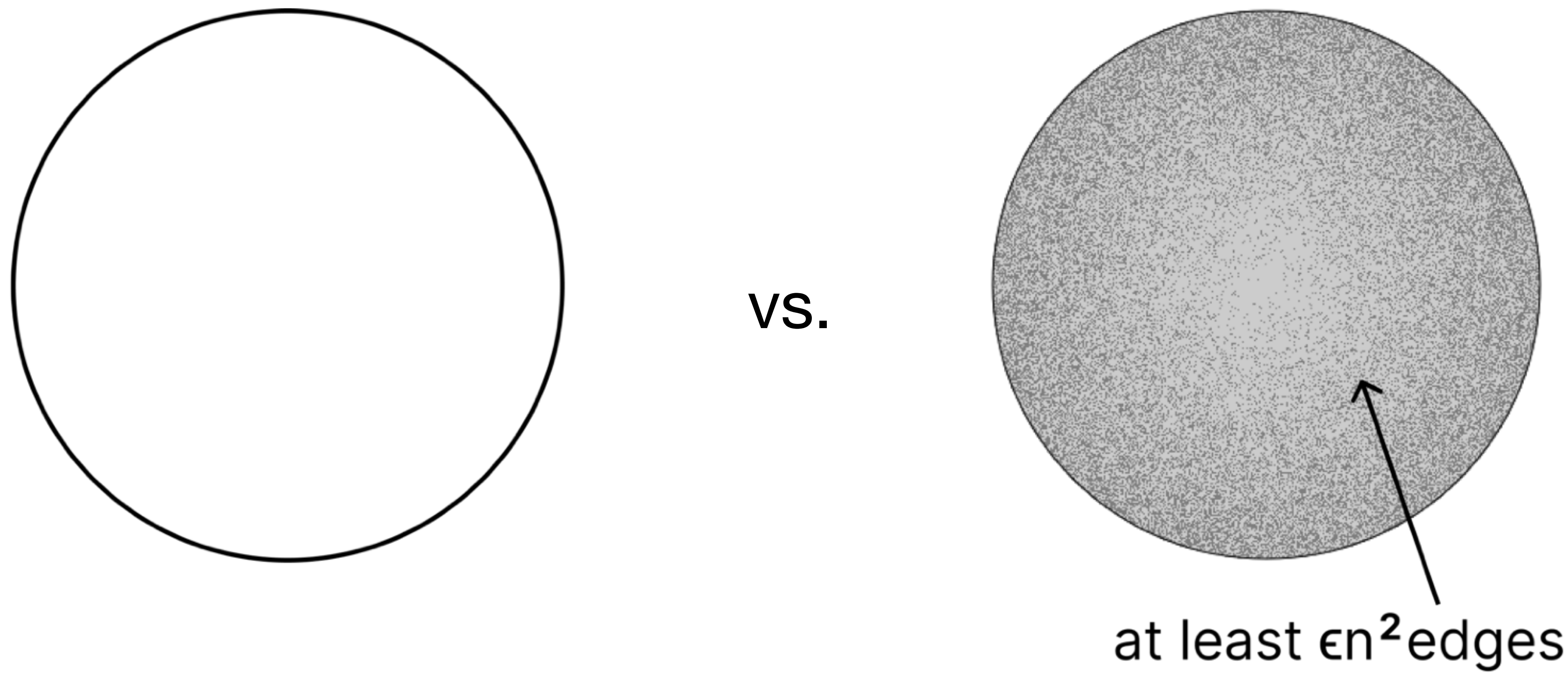
[Kleitman, Winston ’82] - simple case

Note: for survey of combinatorial applications see "Counting Indepedent Sets in Graphs" by Samotij or "The method of hypergraph containers" by Balogh, Morris, and Samotij.

How to use Container Lemmas for Property Testing

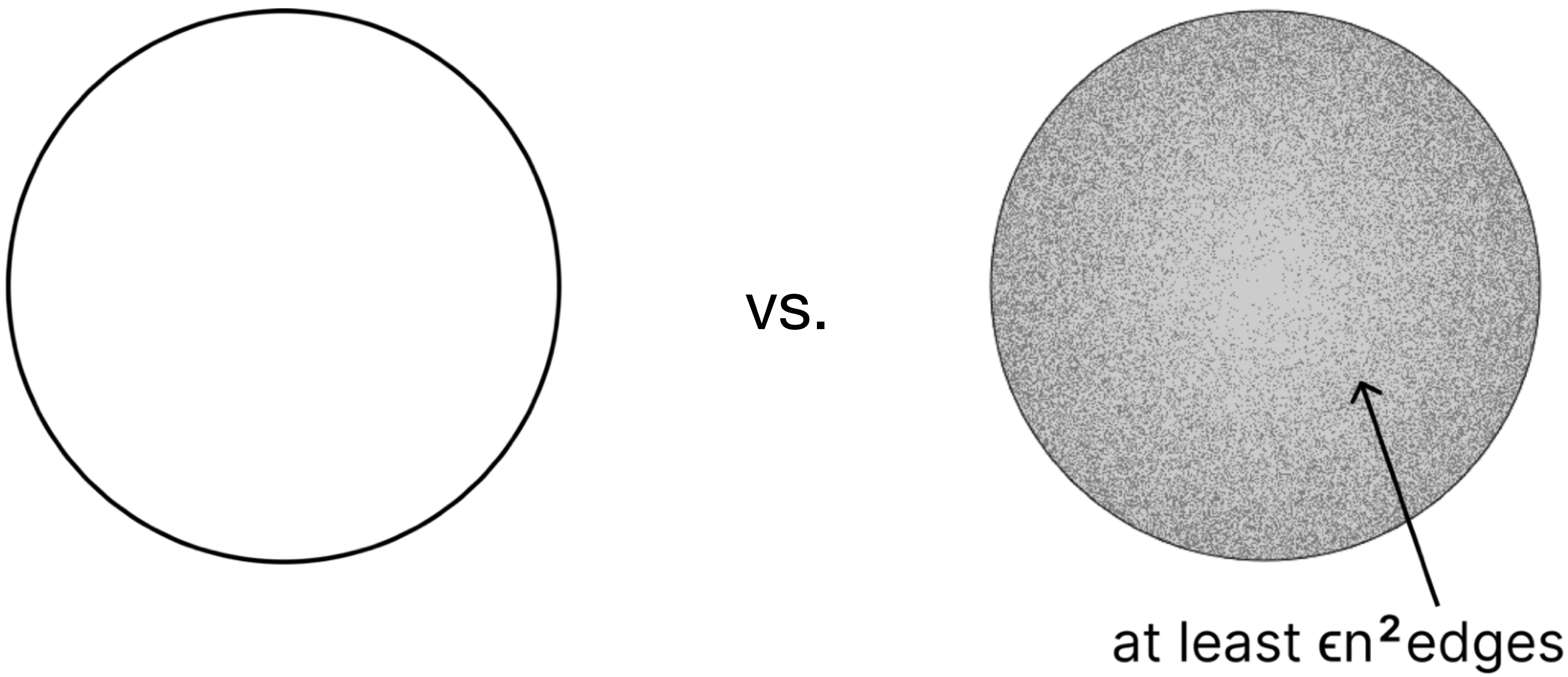
Example: Testing Emptiness with a Container Lemma

Problem: Distinguish empty vs at least ϵn^2 edges



Example: Testing Emptiness with a Container Lemma

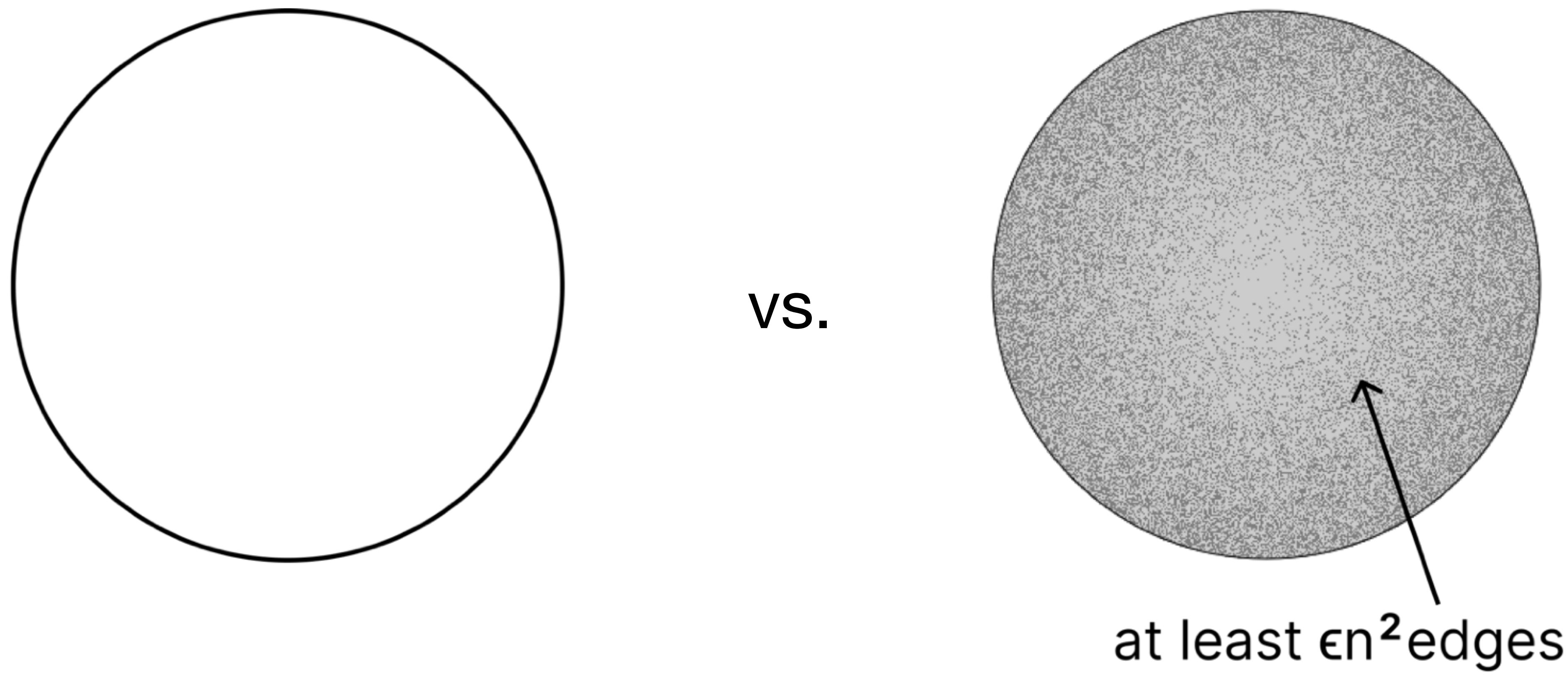
Problem: Distinguish empty vs at least ϵn^2 edges



Theorem: There is a tester for the emptiness property with sample complexity $O(\log n/\epsilon^2)$.

Example: Testing Emptiness with a Container Lemma

Problem: Distinguish empty vs at least ϵn^2 edges

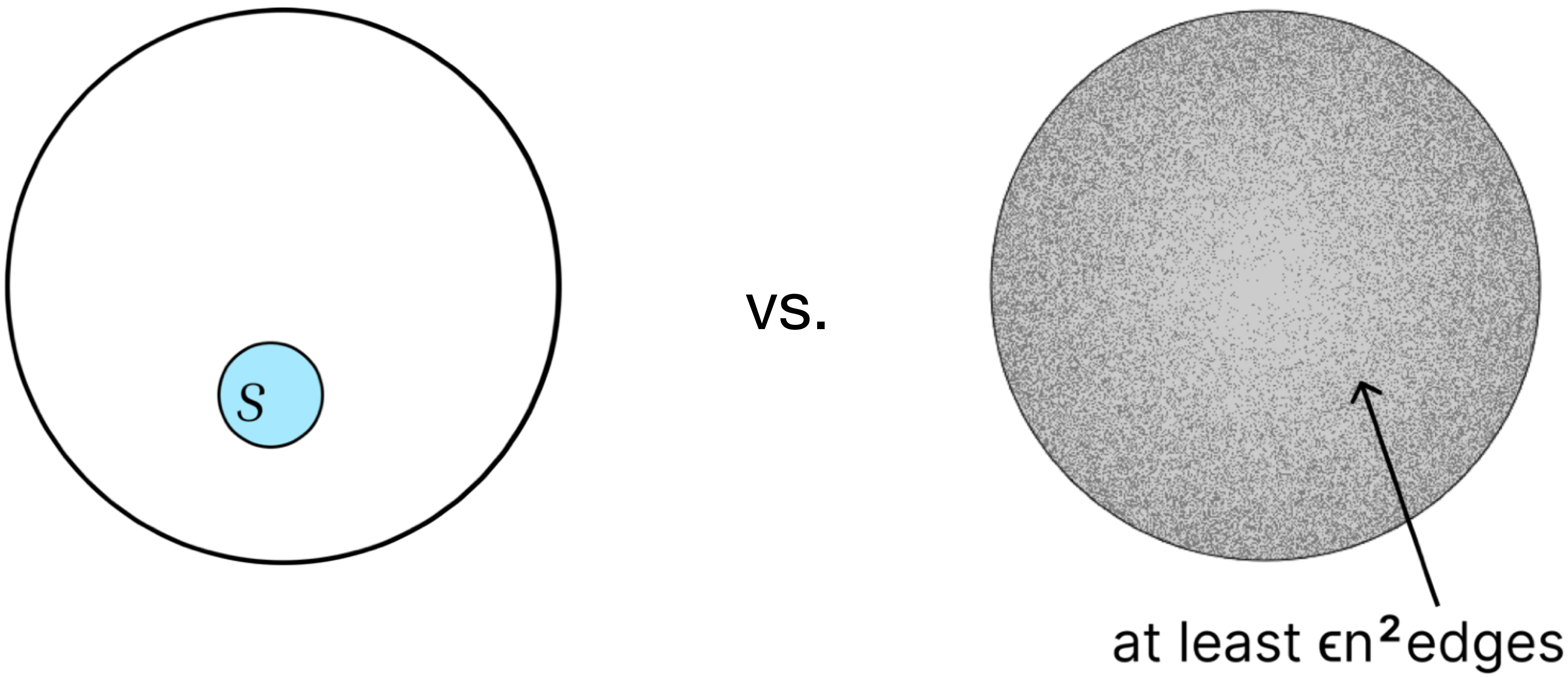


Theorem: There is a tester for the emptiness property with sample complexity $O(\log n/\epsilon^2)$.

Testing Algorithm: Take a random sample S of $O(\log n/\epsilon^2)$ vertices, check if the induced subgraph $G[S]$ is empty.

Example: Testing Emptiness with a Container Lemma

Problem: Distinguish empty vs at least ϵn^2 edges

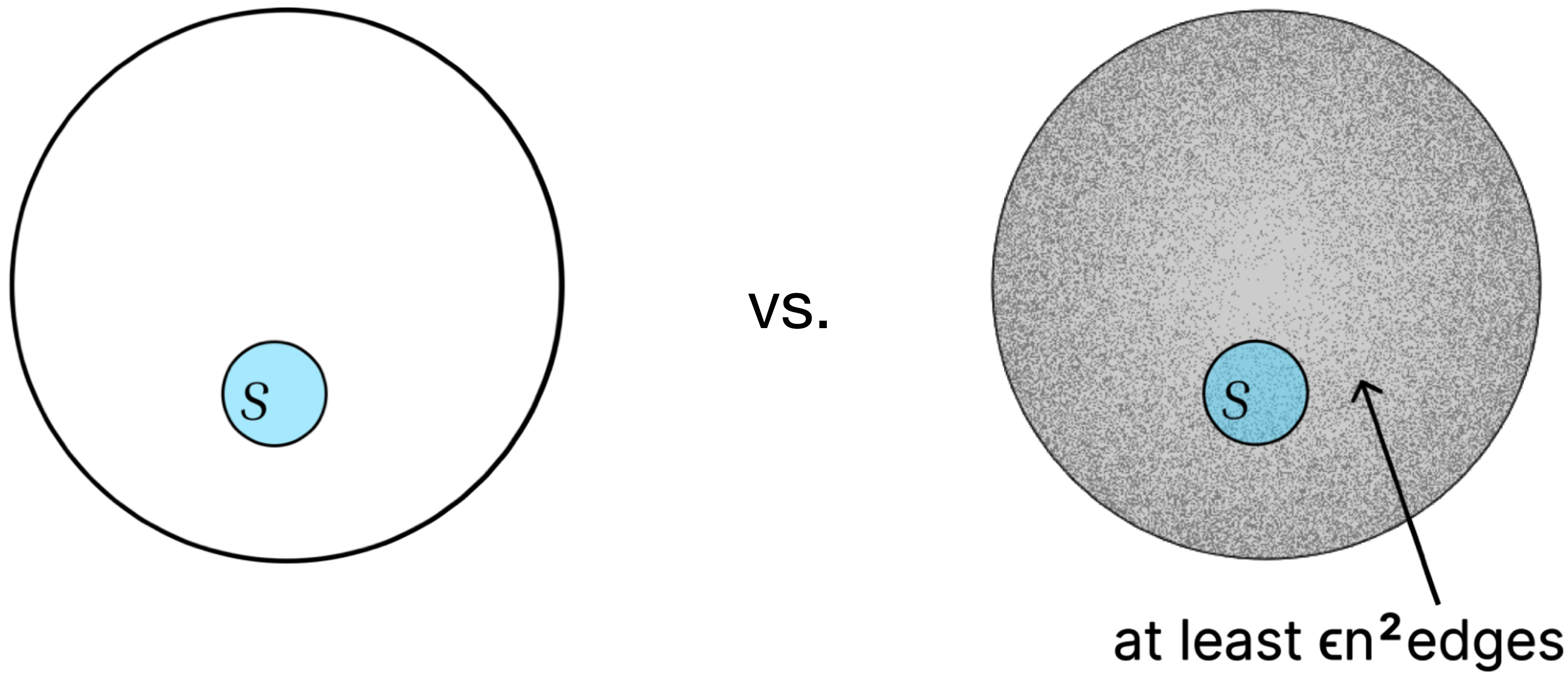


Theorem: There is a tester for the emptiness property with sample complexity $O(\log n/\epsilon^2)$.

Testing Algorithm: Take a random sample S of $O(\log n/\epsilon^2)$ vertices, check if the induced subgraph $G[S]$ is empty.

Example: Testing Emptiness with a Container Lemma

Problem: Distinguish empty vs at least ϵn^2 edges



Theorem: There is a tester for the emptiness property with sample complexity $O(\log n/\epsilon^2)$.

Testing Algorithm: Take a random sample S of $O(\log n/\epsilon^2)$ vertices, check if the induced subgraph $G[S]$ is empty.

Application of Container Lemma

Key Challenge: If G is ϵ-far from empty, show that S is an independent set with only small probability.

Application of Container Lemma

Key Challenge: If G is ϵ-far from empty, show that S is an independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1 - \epsilon/2)n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

Application of Container Lemma

Key Challenge: If G is ϵ-far from empty, show that S is an independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1 - \epsilon/2)\, n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$\Pr[S$ is an independent set$]$

Application of Container Lemma

Key Challenge: If G is ϵ-far from empty, show that S is an independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1 - \epsilon/2)\, n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ is an independent set}] \leq \sum_{C \in \mathscr{C}} \Pr[S \subseteq C]$$

Application of Container Lemma

Key Challenge: If G is ϵ-far from empty, show that S is an independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1 - \epsilon/2)\, n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ is an independent set}] \leq \sum_{C \in \mathscr{C}} \Pr[S \subseteq C]$$

$$\leq \binom{n}{1/\epsilon} \left(1 - \frac{\epsilon}{2}\right)^{|S|}$$

Application of Container Lemma

Key Challenge: If G is ϵ-far from empty, show that S is an independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1 - \epsilon/2)\, n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ is an independent set}] \leq \sum_{C \in \mathscr{C}} \Pr[S \subseteq C]$$

$$\leq \binom{n}{1/\epsilon} \left(1 - \frac{\epsilon}{2}\right)^{|S|}$$

$$\leq \exp\left(\frac{1}{\epsilon}\log(n) - \frac{\epsilon\,|S|}{2}\right)$$

Application of Container Lemma

Key Challenge: If G is ϵ-far from empty, show that S is an independent set with only small probability.

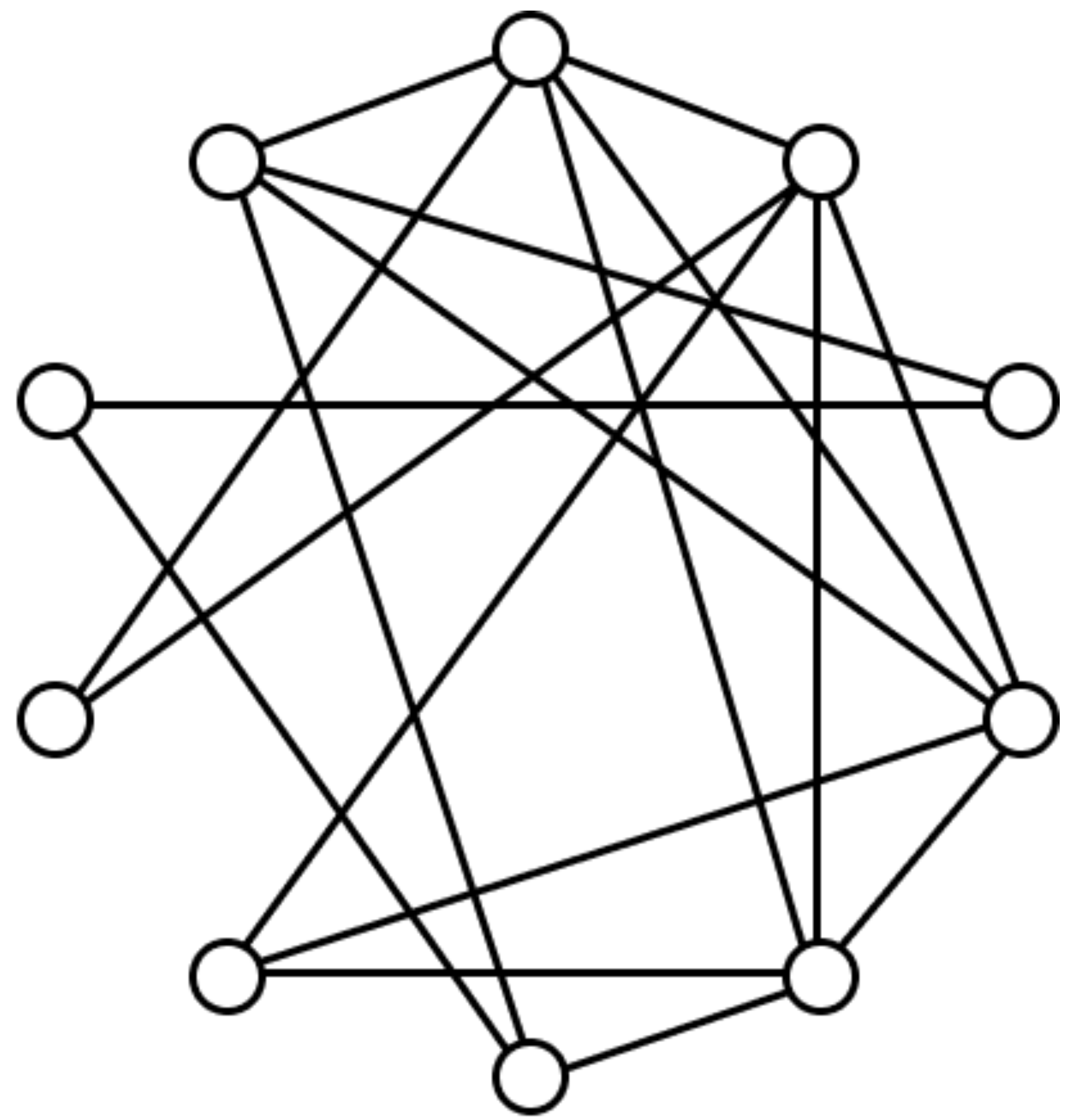
Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1 - \epsilon/2)\,n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\begin{aligned}
\Pr[S \text{ is an independent set}] \quad &\leq \sum_{C \in \mathscr{C}} \Pr[S \subseteq C] \\
&\leq \binom{n}{1/\epsilon} \left(1 - \frac{\epsilon}{2}\right)^{|S|} \\
&\leq \exp\left(\frac{1}{\epsilon}\log(n) \;-\; \frac{\epsilon\,|S|}{2}\right) \\
&\leq 1/3, \text{ as long as } s \gtrsim \frac{\log n}{\epsilon^2}.
\end{aligned}$$

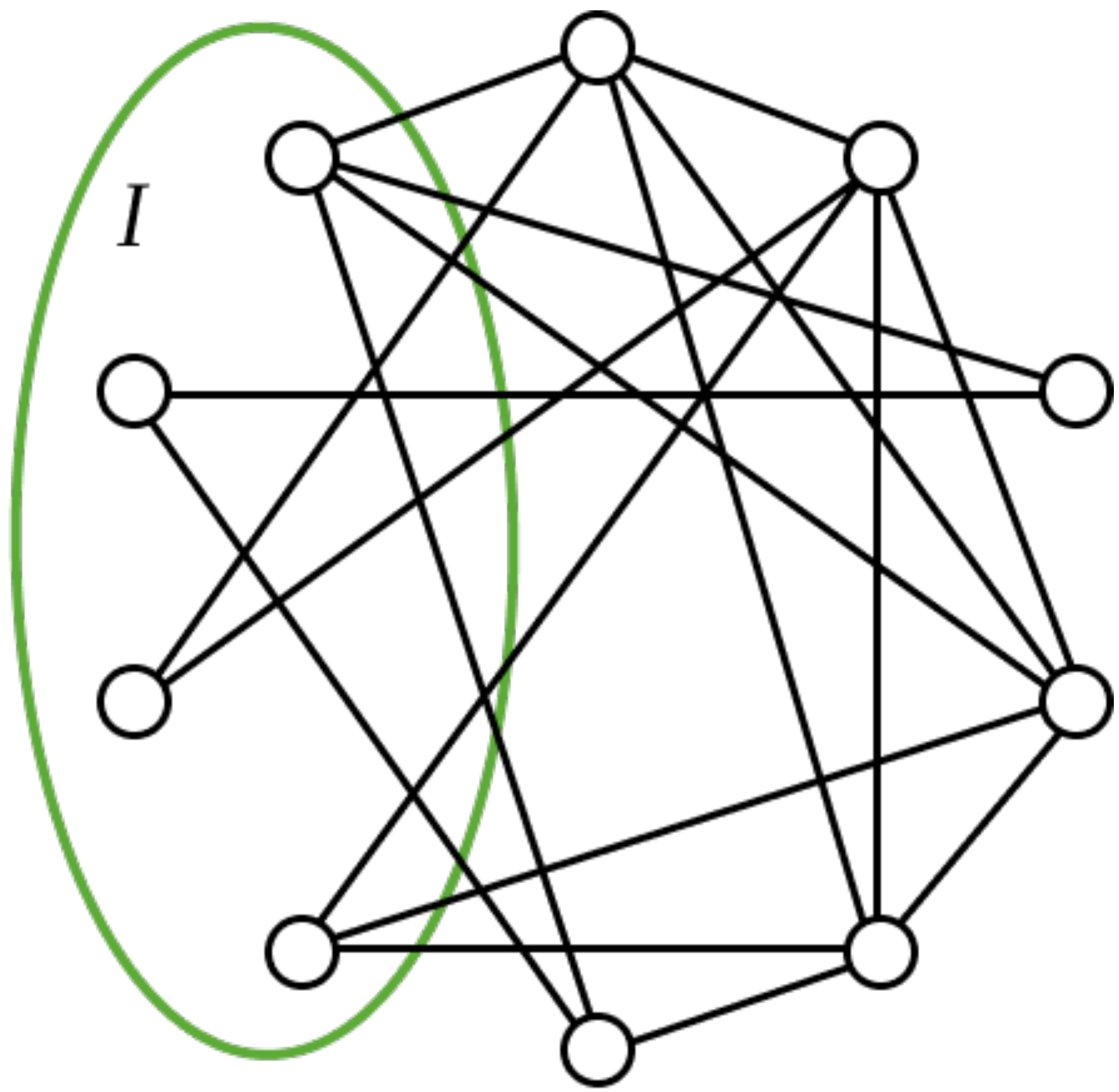
How to prove a Container Lemma - An Encoding Argument

Encoding Independent Sets

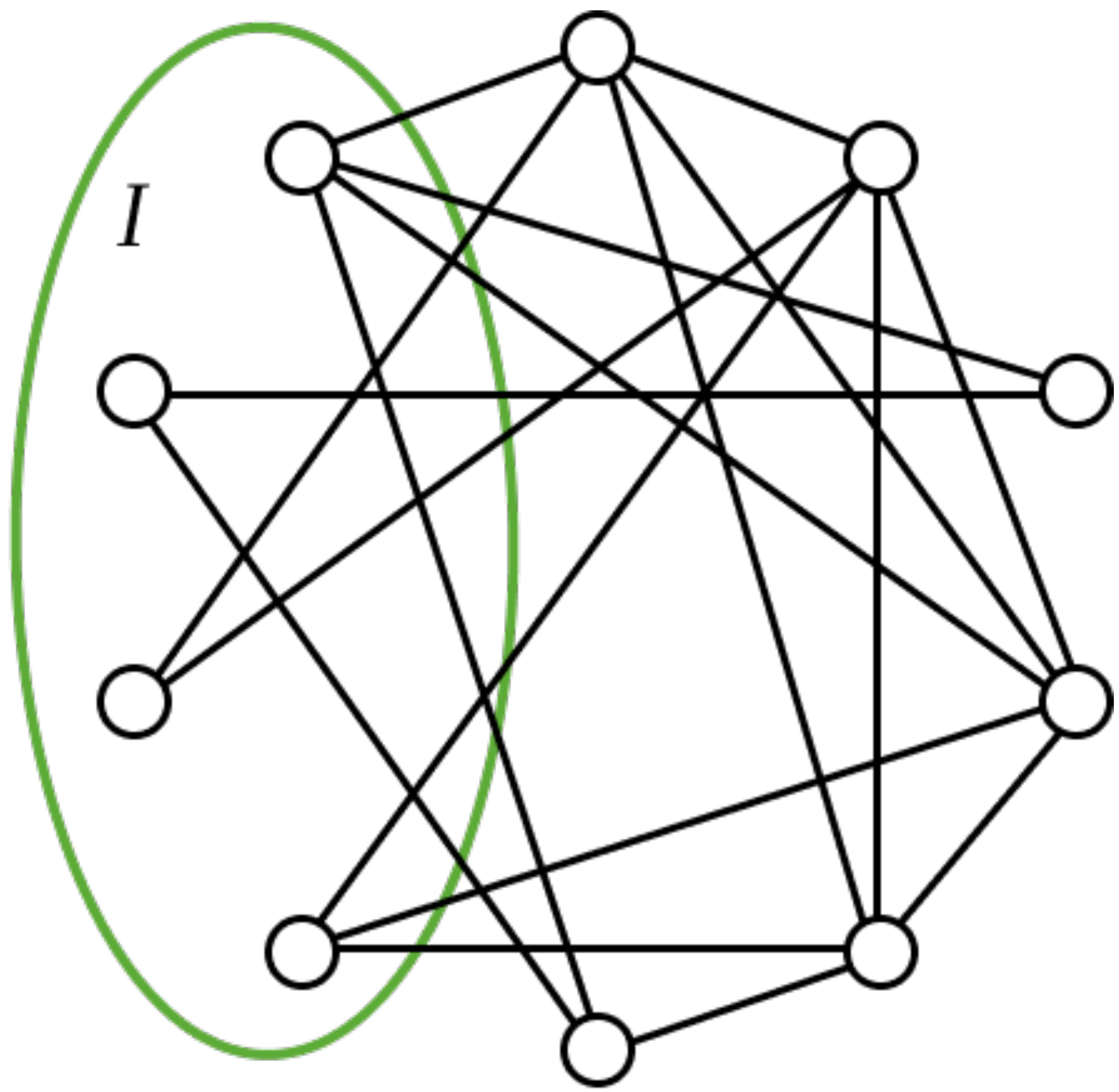
Encoding Independent Sets



Encoding Independent Sets

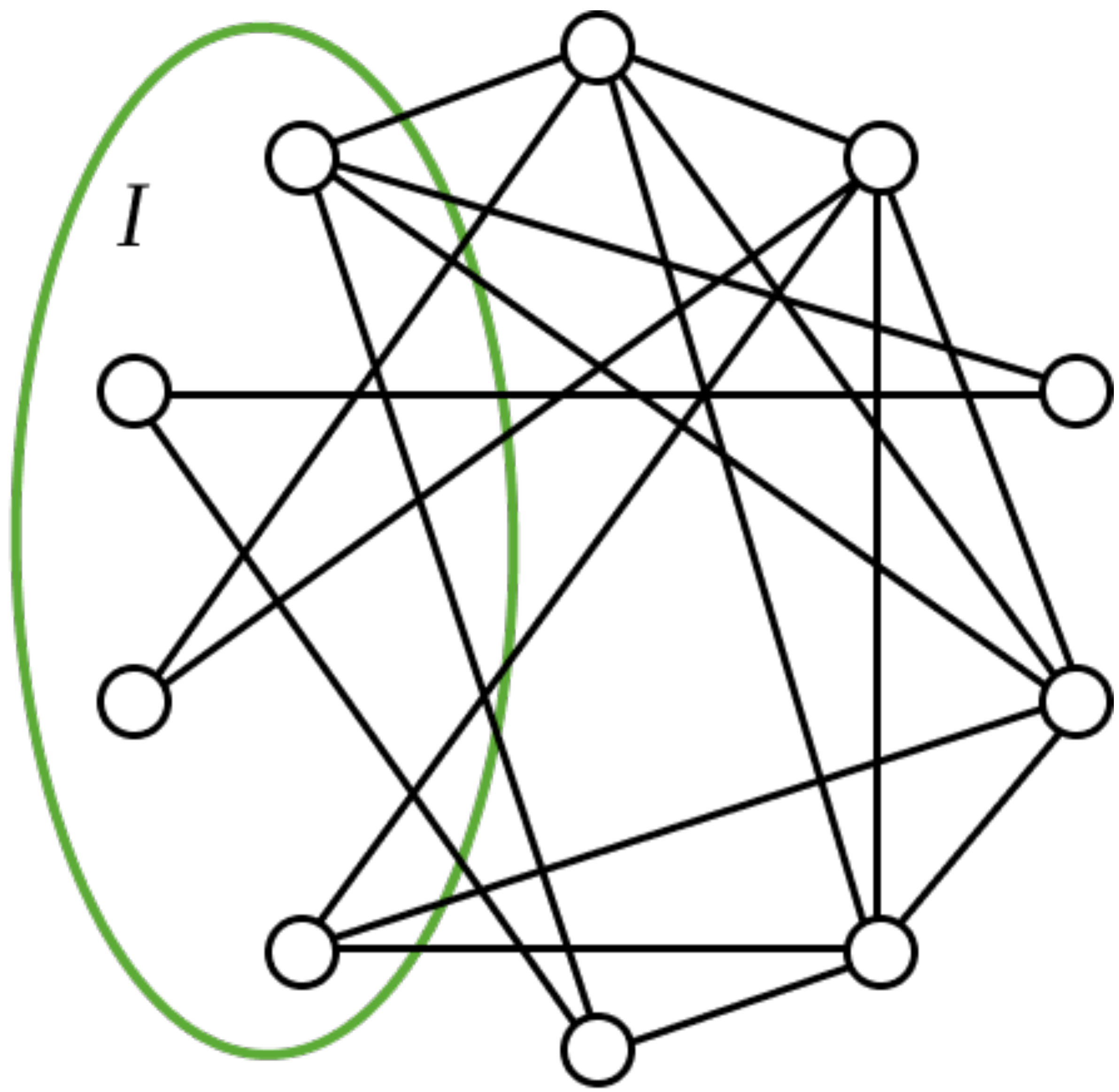


Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

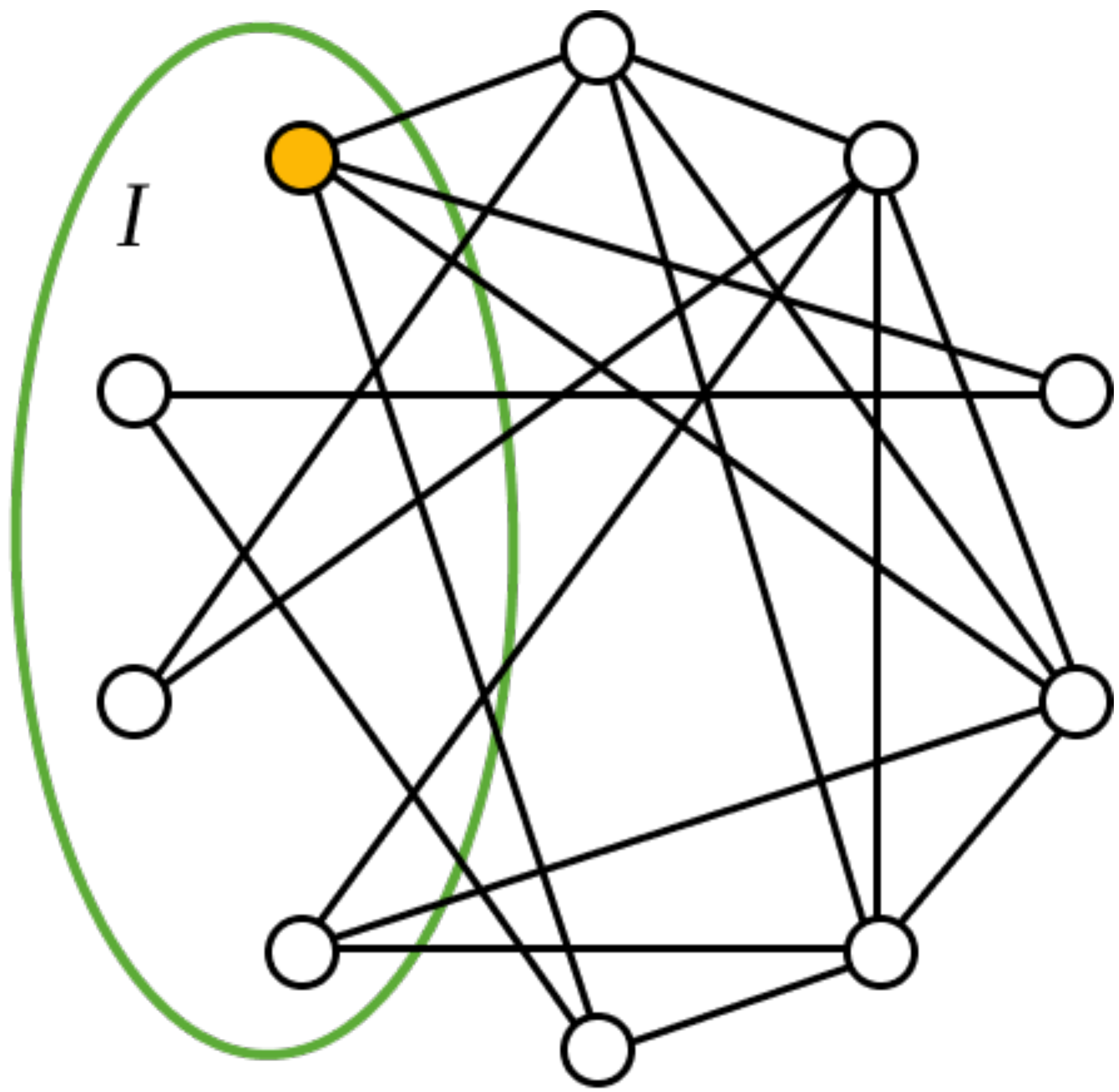
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

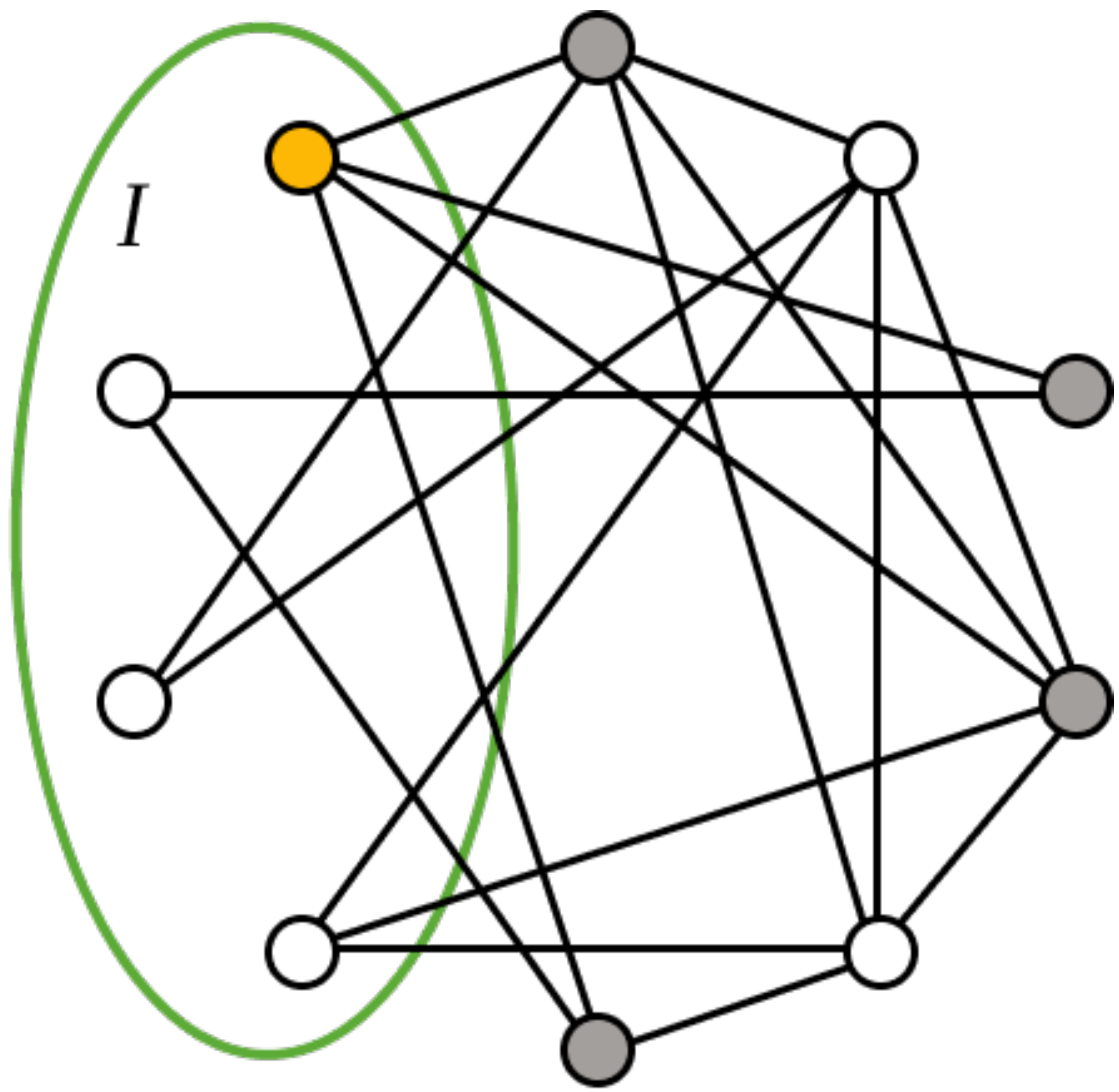
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

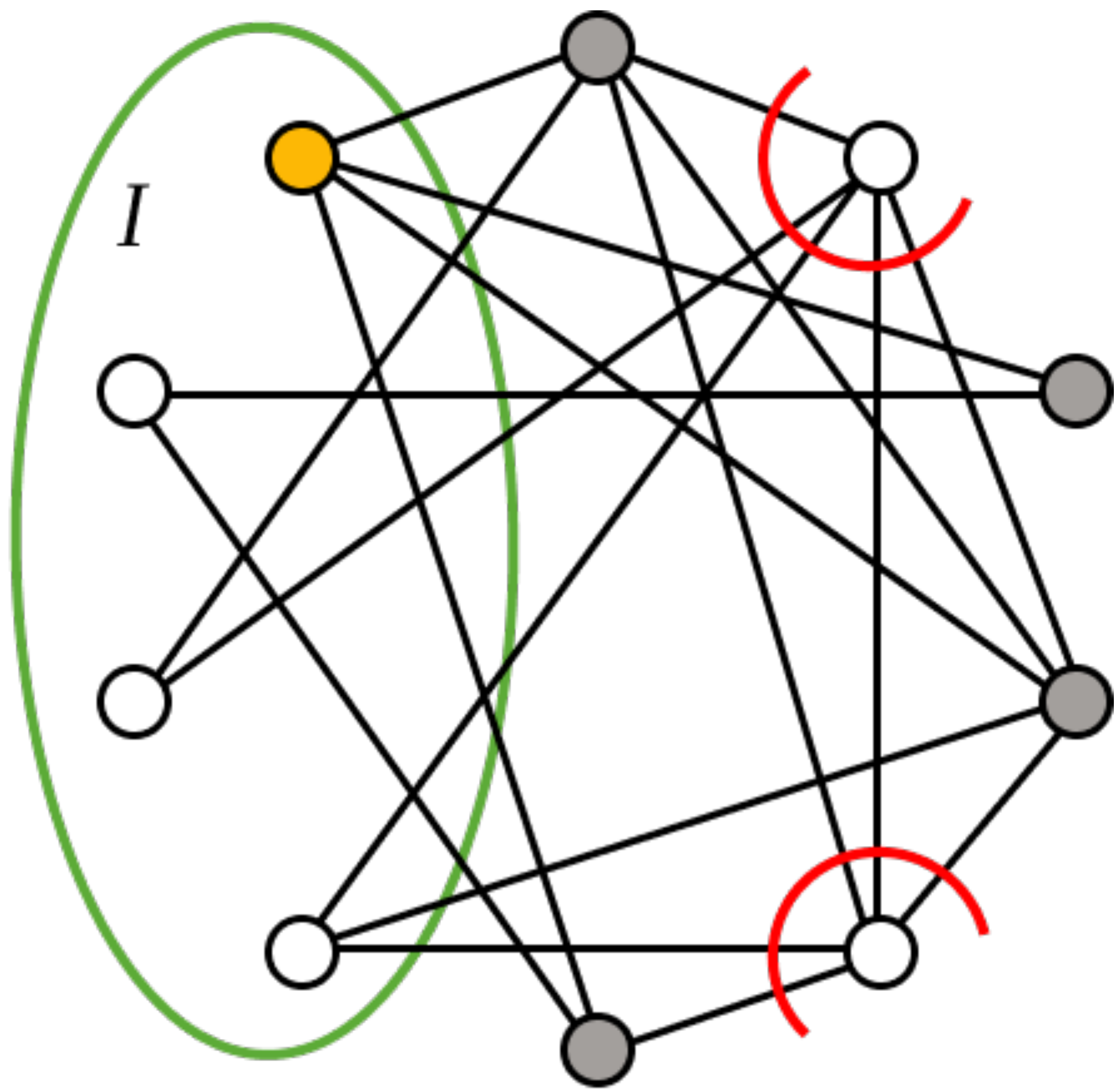
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

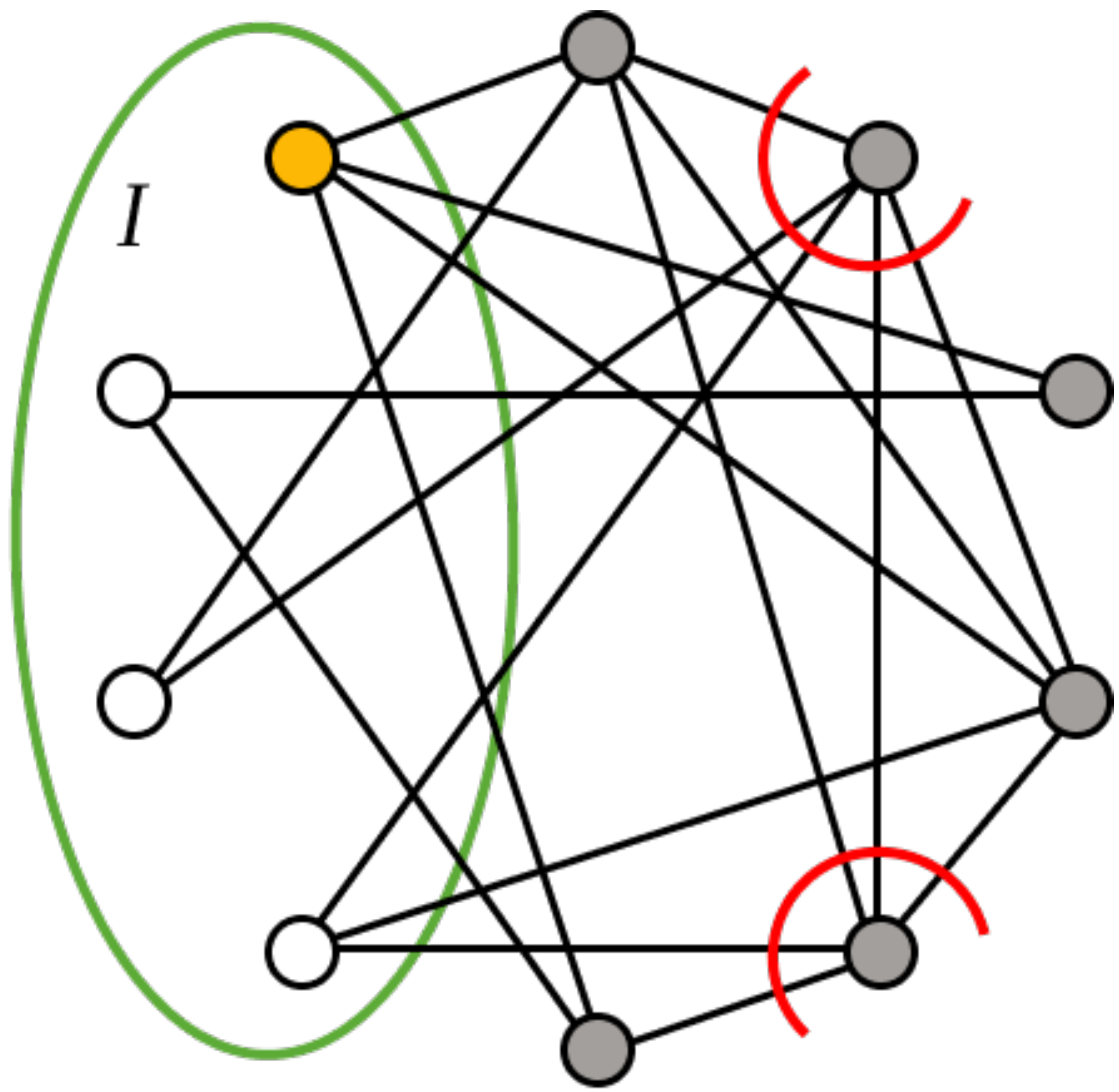
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

Container Generator Algorithm

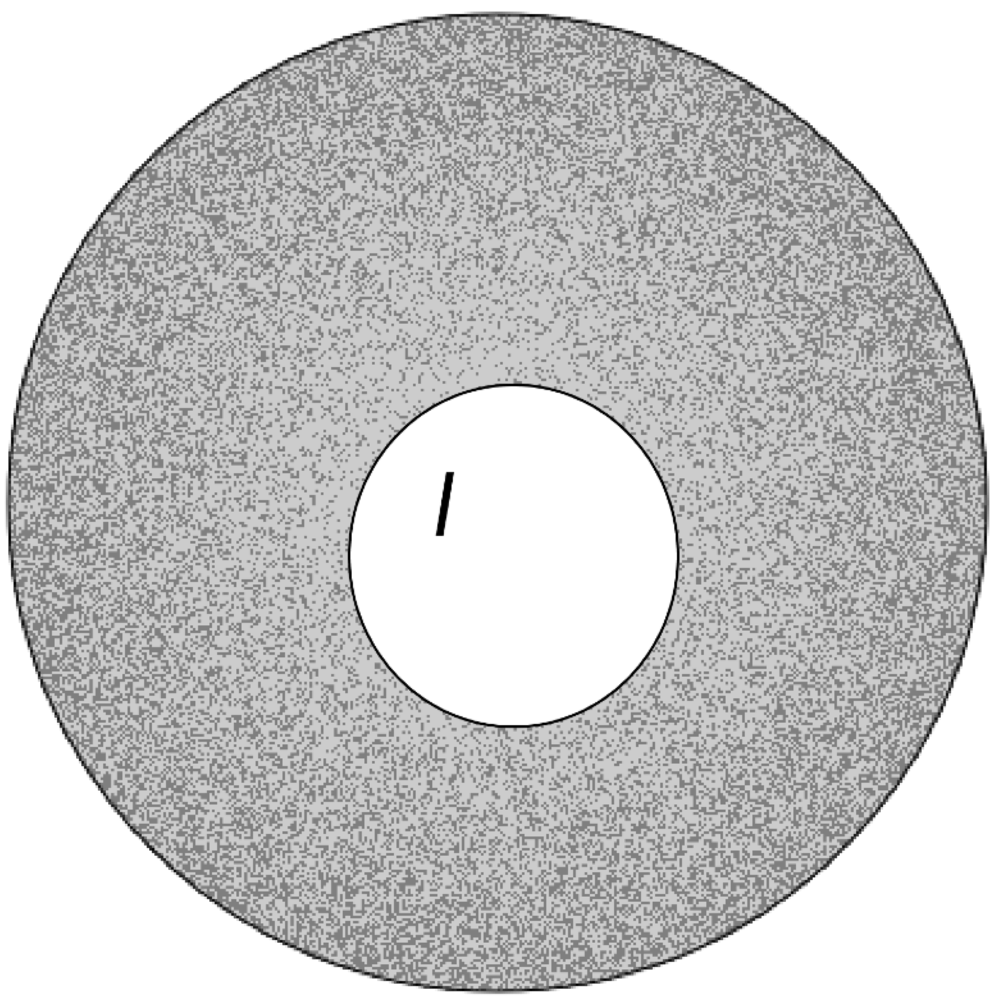
Input: Graph G and an independent set I

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

Container Generator Algorithm

Input: Graph G and an independent set I

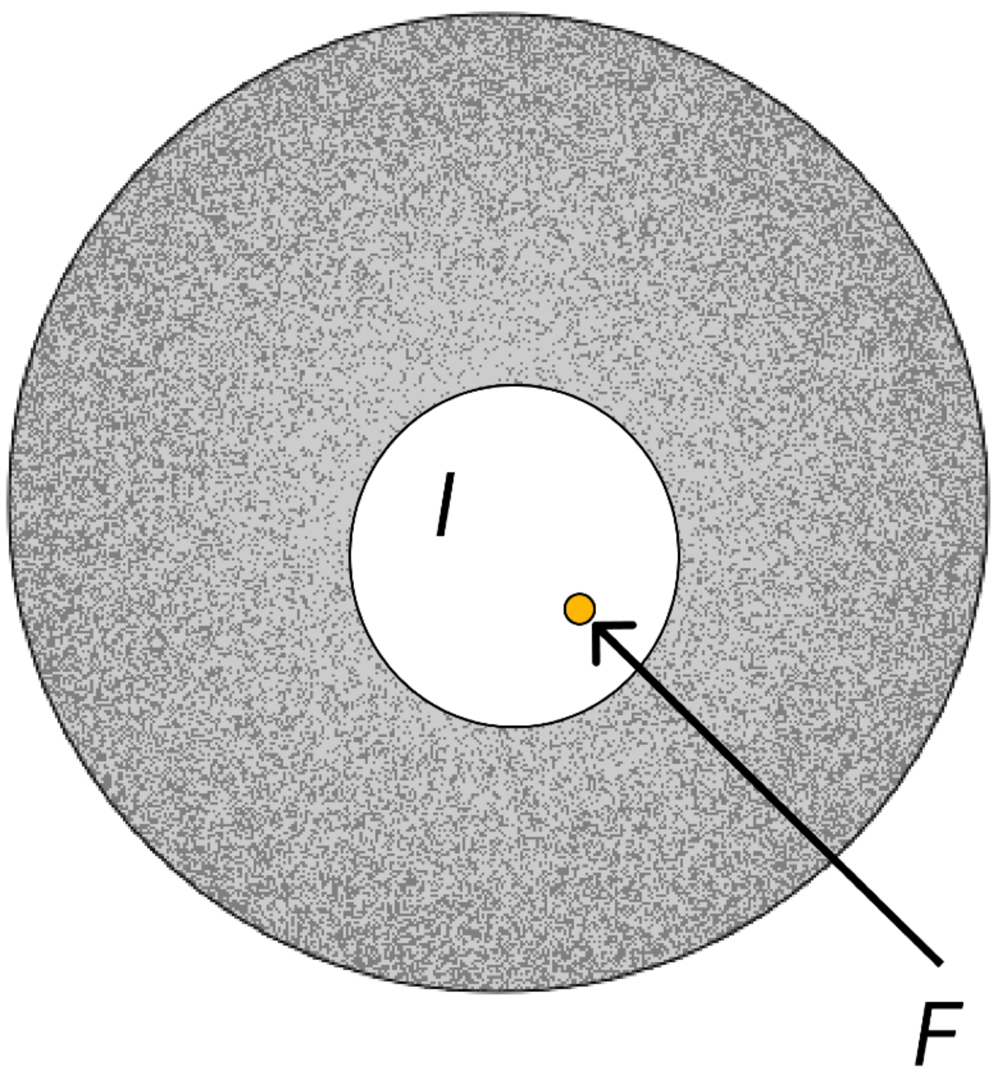
- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.



Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

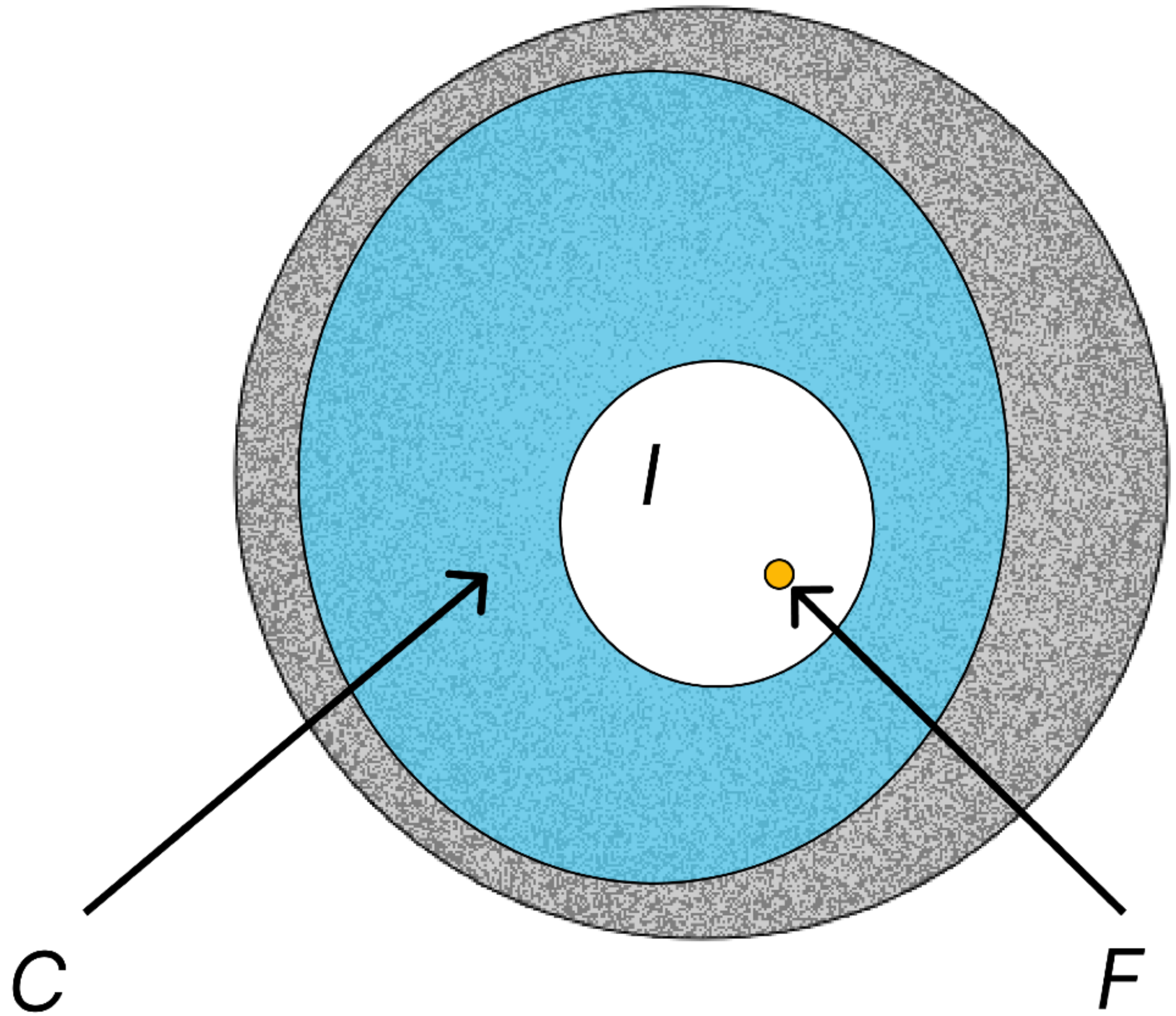


1st Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

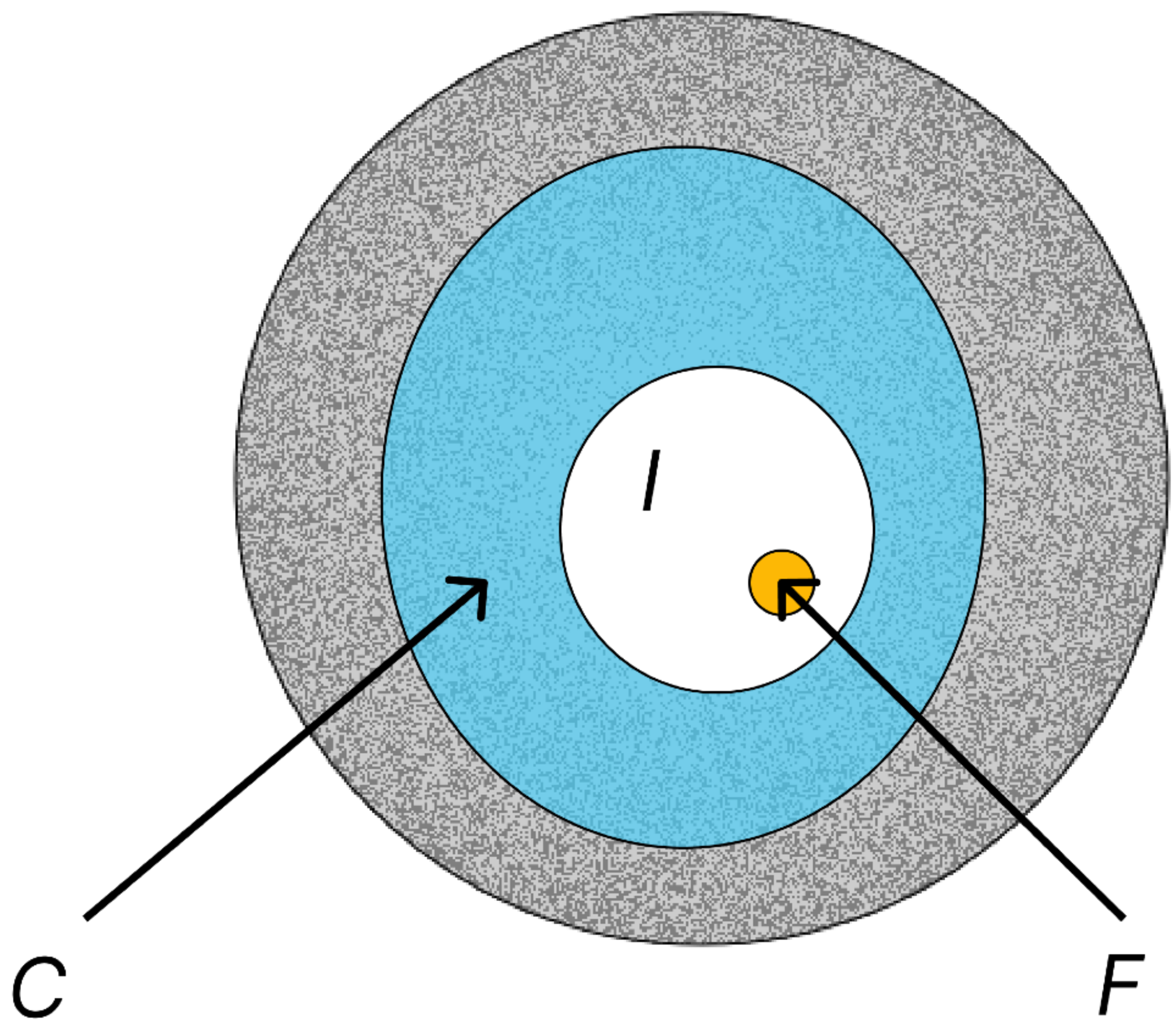


1st Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

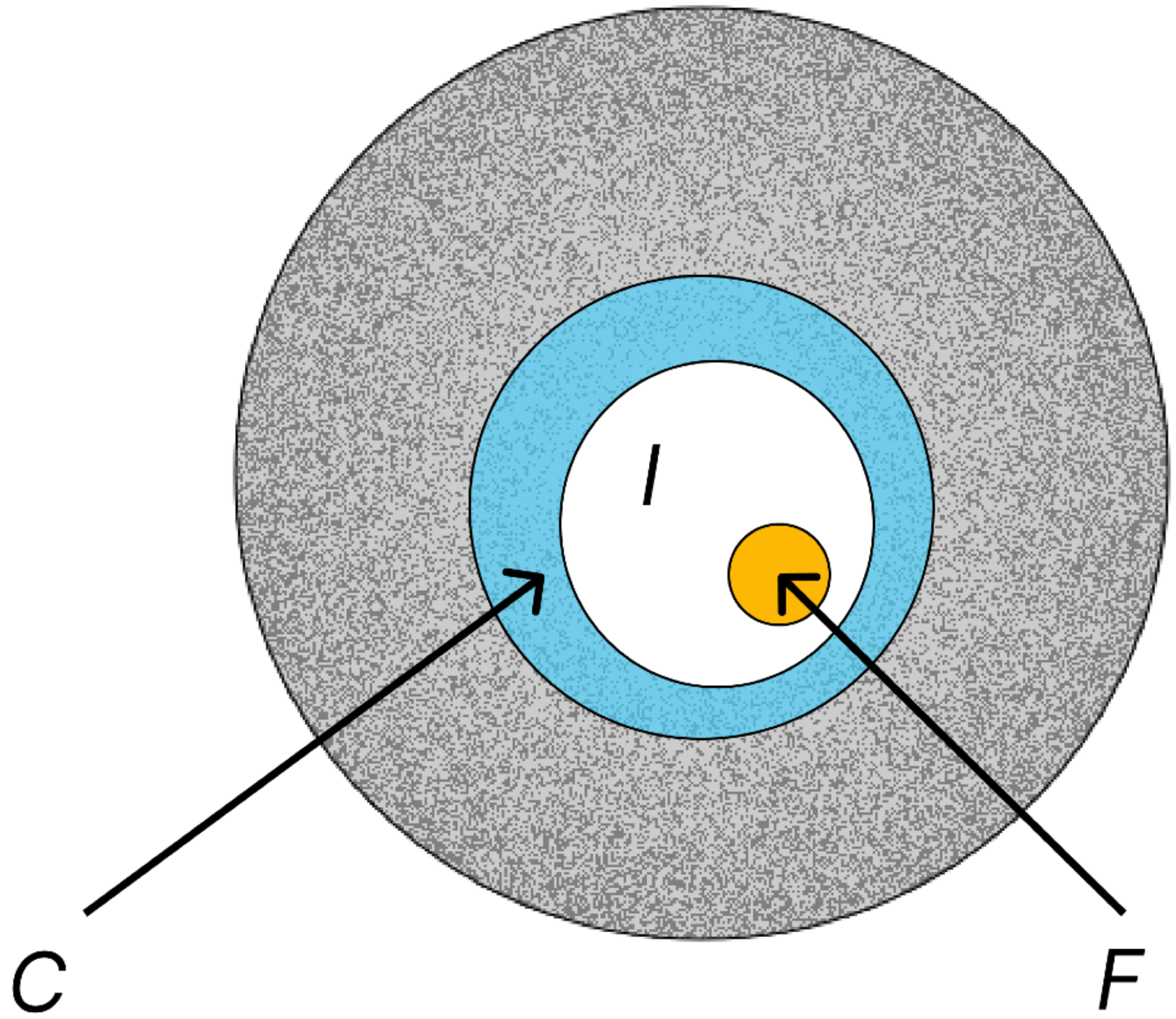


2nd Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

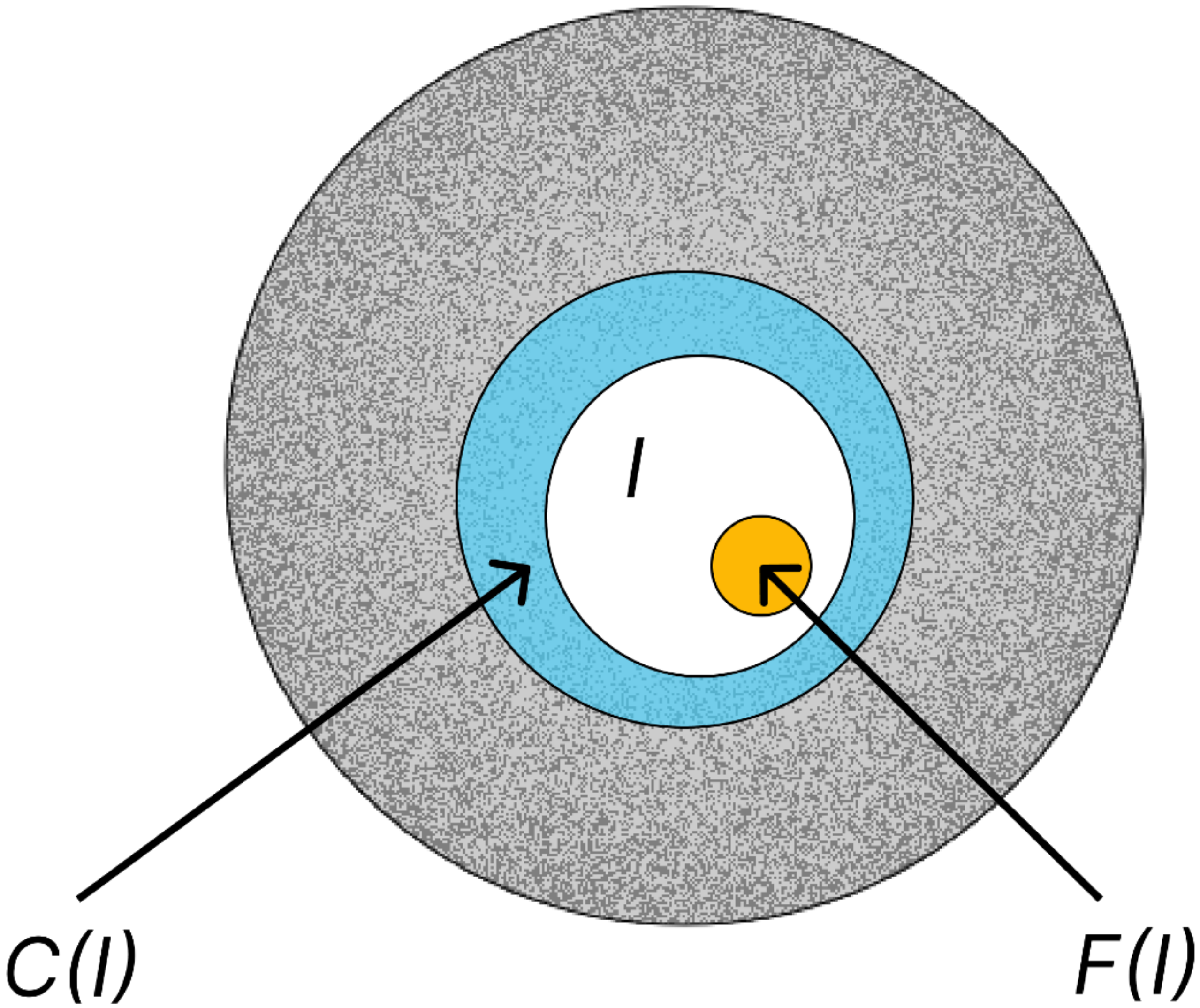


Final Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

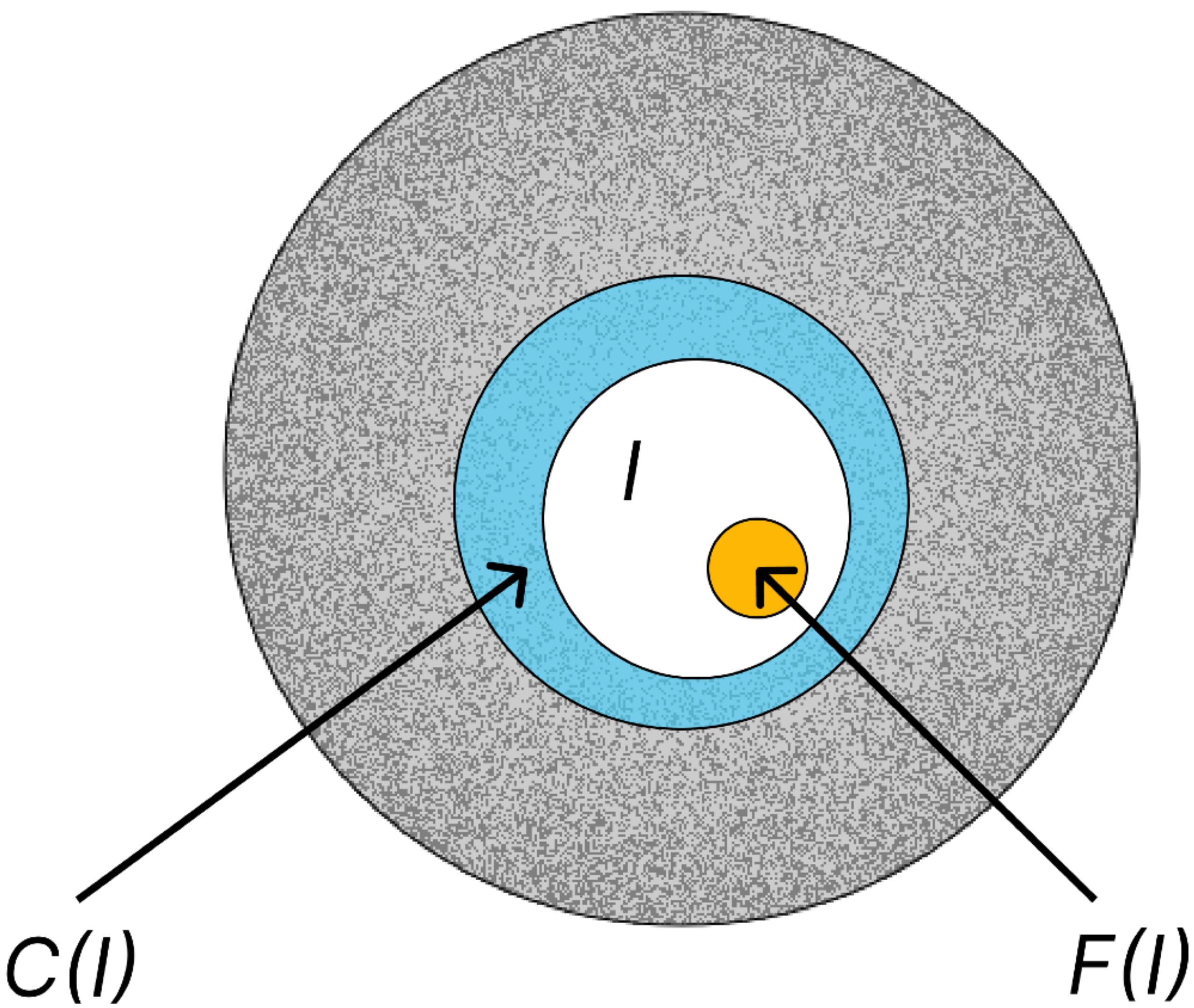


Final Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.



Final Iteration

Key Observations:

- $I \subseteq C(I)$
- $C(I) = C(F(I))$

Proving the Simple Graph Container Lemma

Lemma: For any ϵ let $G = (V, E)$ be a graph with at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. $|\mathscr{C}| \leq \binom{n}{4/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

[Kleitman, Winston '82] - simple case

Proving the Simple Graph Container Lemma

Lemma: For any ϵ let $G = (V, E)$ be a graph with at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. $|\mathscr{C}| \leq \binom{n}{4/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

[Kleitman, Winston '82] - simple case

Proof: Run container procedure on every independent for $4/\epsilon$ steps

Let $\mathscr{C} = \{C(I) : I \text{ is independent set in } G\}$

$= \{C(F) : F = F(I) \text{ for some independent set } I \text{ in } G\}$

Proving the Simple Graph Container Lemma

Lemma: For any ϵ let $G = (V, E)$ be a graph with at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. $|\mathscr{C}| \leq \binom{n}{4/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

[Kleitman, Winston '82] - simple case

Proof: Run container procedure on every independent for $4/\epsilon$ steps

Let $\mathscr{C} = \{C(I) : I \text{ is independent set in } G\}$

$= \{C(F) : F = F(I) \text{ for some independent set } I \text{ in } G\}$

Observe that in each step, if there are more than $\epsilon n^2/2$ edges then at least $\epsilon n/4$ vertices are removed. Hence after $4/\epsilon$ steps the final container has less than $\epsilon n^2/2$ edges. And so $|C| \leq (1 - \epsilon/2)n$ $\square$

Restating our Theorem

Restating Informal Version of Theorem

Theorem: Let Π be a hereditary graph property. Then

Π is ϵ-testable with one sided error and sample complexity s

$$\Updownarrow$$

Π has an "ϵ-container lemma" parameterized by $\mathrm{poly}(s)$

Restating Informal Version of Theorem

Theorem: Let Π be a hereditary graph property. Then

Π is ϵ-testable with one sided error and sample complexity s

$$\Updownarrow$$

Π has an "ϵ-container lemma" parameterized by $\mathrm{poly}(s)$

Defining ϵ-Container Lemma: An Example

Example: An ϵ-container lemma for the $\Pi =$ emptiness property

Lemma: For any ϵ let $G = (V, E)$ be ϵ-far from empty. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

[Kleitman, Winston '82] - simple case

Defining ϵ-Container Lemma: An Example

Example: An ϵ-container lemma for the $\Pi =$ emptiness property

Lemma: For any ϵ let $G = (V, E)$ be ϵ-far from **empty**. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every **independent set I**, there exists $C \in \mathscr{C}$ with $I \subseteq C$,
2. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

[Kleitman, Winston '82] - simple case

Defining ϵ-Container Lemma Informally

Definition (informal): An ϵ-container lemma for Π is the following:

Lemma: For any ϵ let $G = (V, E)$ be ϵ-far from Π. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every S with $G[S] \in \Pi$, there exists $C \in \mathscr{C}$ with $S \subseteq C$,
2. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

Defining ϵ-Container Lemma Informally

Definition (informal): An ϵ-container lemma for Π is the following:

Lemma: For any ϵ let $G = (V, E)$ be ϵ-far from Π. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. for every S with $G[S] \in \Pi$, there exists $C \in \mathscr{C}$ with $S \subseteq C$,
2. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
3. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)n$.

The containers are covering Π-subgraphs instead of independent sets!

Defining ϵ-Container Lemma

Definition: Let Π be a graph property. We say Π has an

"ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$"

when the following holds. For any $G = (V, E)$ that is ϵ-far from Π there exists a set $\mathcal{F} \subseteq 2^V$ of fingerprints and container function $C : \mathcal{F} \to 2^V$ that satisfies:

1. For every $S \subseteq V$ for which $G[S] \in \Pi$, there exists $F \in \mathcal{F}$ such that $F \subseteq S \subseteq C(F)$,

2. for every $F \in \mathcal{F}$, $|F| \leq f$, $|C(F)| \leq (1-\alpha)n$ and $G[C(F)]$ can be made to satisfy Π by modifying at most ϵn^2 edges.

Theorem Restated

Theorem Restated

Theorem*: Let Π be a hereditary graph property. Then

Π is ϵ-testable with one sided error and sample complexity s

$$\Updownarrow$$

Π has an ϵ-container lemma with fingerprint size $\mathrm{poly}(s)$ and container size $\left(1 - \frac{1}{\mathrm{poly}(s)}\right)$

* Omitting some details

Proof Sketch of Theorem

Proof Sketch (testability => container lemma)

Proof Sketch (testability => container lemma)

Proof sketch with the specific example $\Pi =$ 3-colorability.

Recall that 3-colorability is testable with sample complexity $s \sim 1/\epsilon$.

[Sohler '12]

We want to prove an ϵ-container lemma like: if G is ϵ-far then there is a collection of containers that covers all the 3-colorable subgraphs.

Proof Sketch (testability => container lemma)

Proof sketch with the specific example $\Pi =$ 3-colorability.
Recall that 3-colorability is testable with sample complexity $s \sim 1/\epsilon$.

[Sohler '12]

We want to prove an ϵ-container lemma like: if G is ϵ-far then there is a collection of containers that covers all the 3-colorable subgraphs.

What does it mean to be testable with sample complexity $s \sim 1/\epsilon$?

Proof Sketch (testability => container lemma)

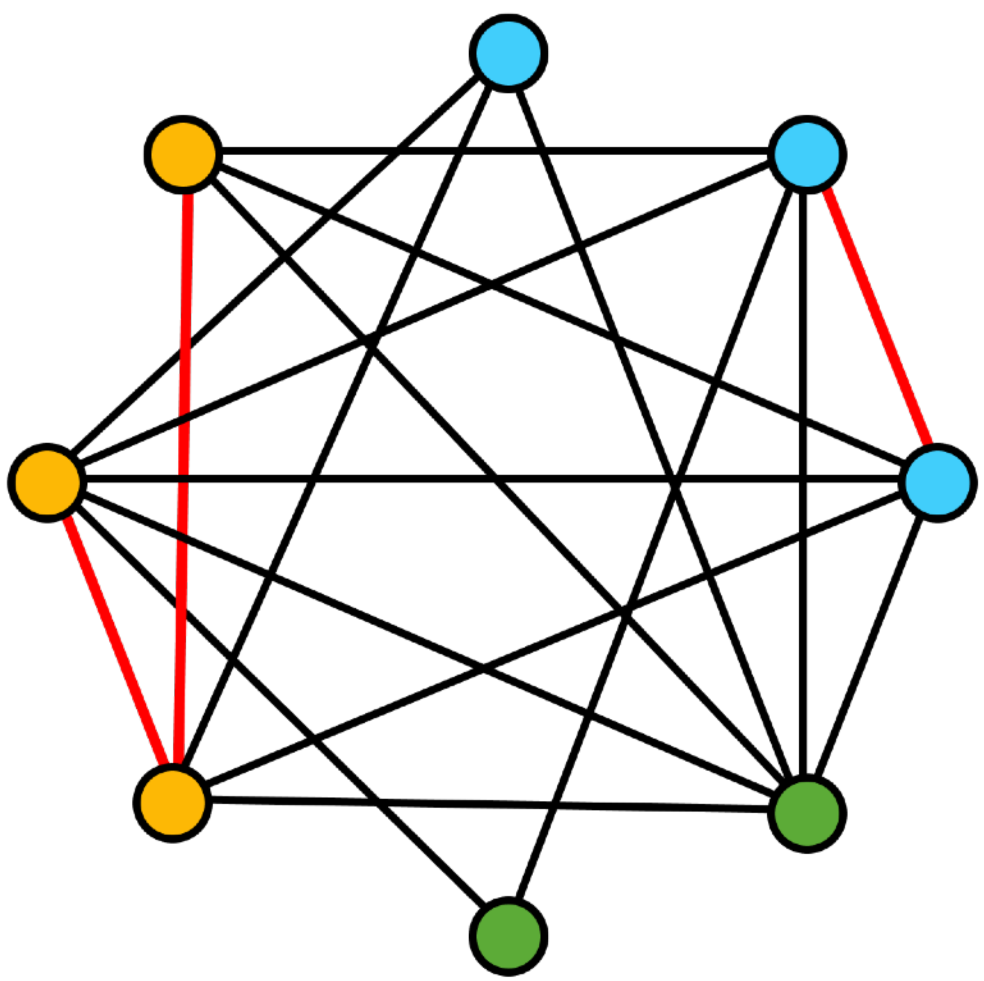
Proof sketch with the specific example $\Pi = $ 3-colorability.

Recall that 3-colorability is testable with sample complexity $s \sim 1/\epsilon$.

[Sohler '12]

We want to prove an ϵ-container lemma like: if G is ϵ-far then there is a collection of containers that covers all the 3-colorable subgraphs.

What does it mean to be testable with sample complexity $s \sim 1/\epsilon$?



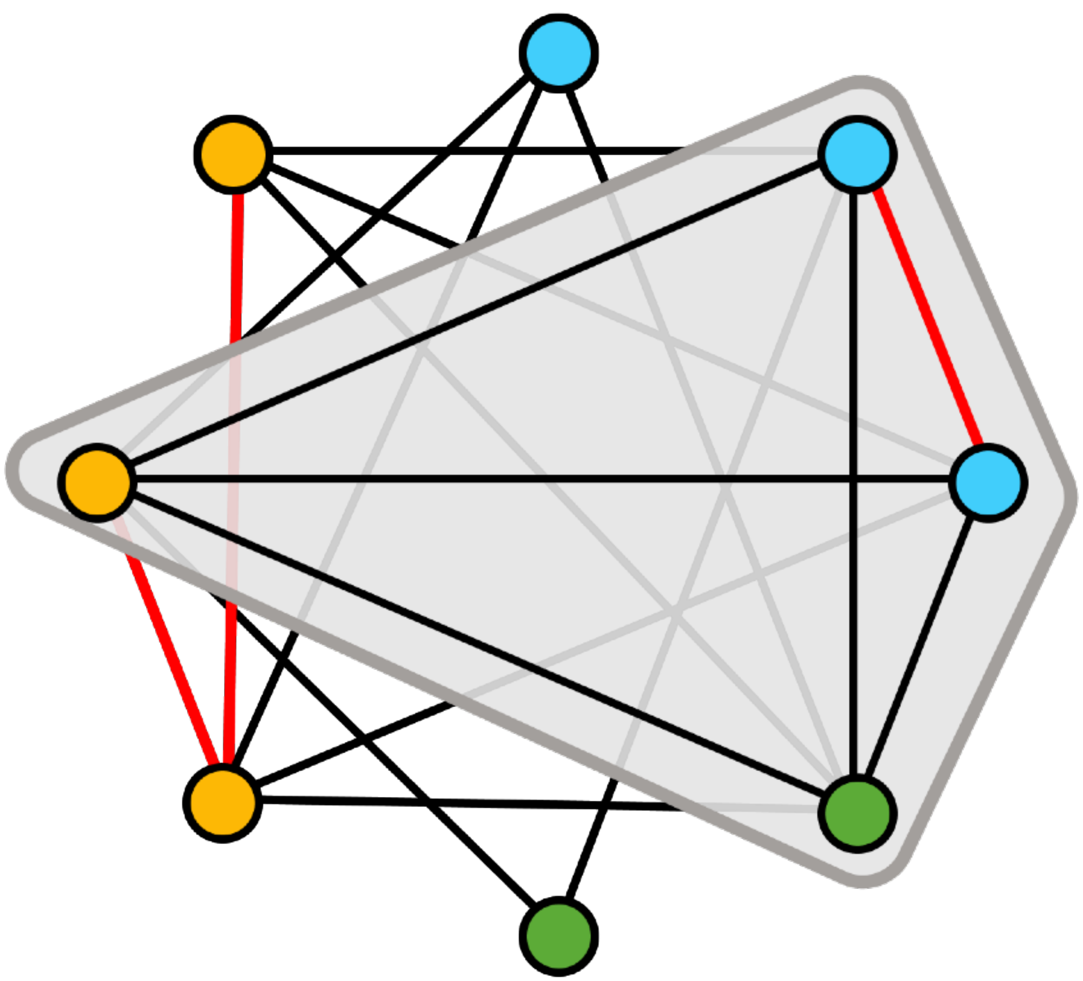
If G is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

Proof Sketch (testability => container lemma)

Proof sketch with the specific example $\Pi =$ 3-colorability.
Recall that 3-colorability is testable with sample complexity $s \sim 1/\epsilon$.
[Sohler '12]

We want to prove an ϵ-container lemma like: if G is ϵ-far then there is a collection of containers that covers all the 3-colorable subgraphs.

What does it mean to be testable with sample complexity $s \sim 1/\epsilon$?



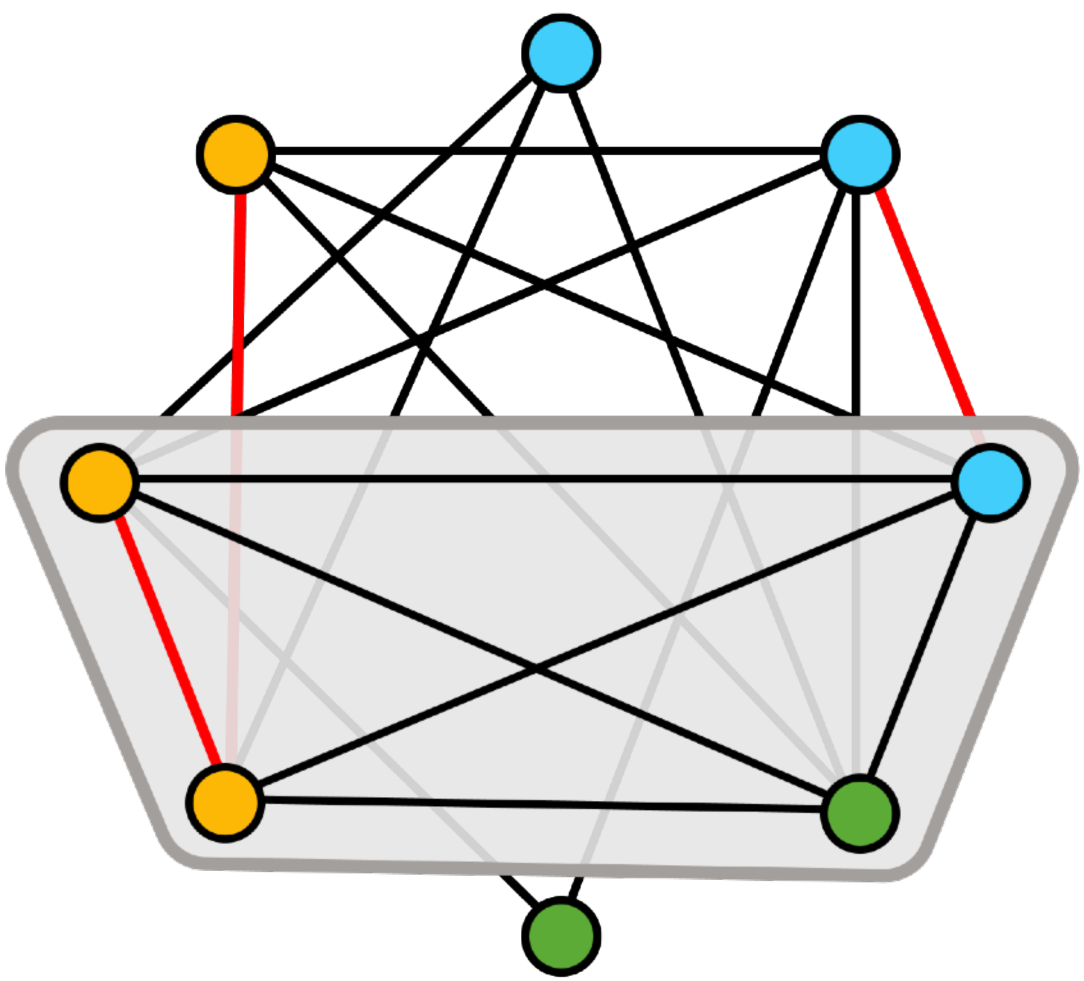
If G is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

Proof Sketch (testability => container lemma)

Proof sketch with the specific example $\Pi =$ 3-colorability.
Recall that 3-colorability is testable with sample complexity $s \sim 1/\epsilon$.
[Sohler '12]

We want to prove an ϵ-container lemma like: if G is ϵ-far then there is a collection of containers that covers all the 3-colorable subgraphs.

What does it mean to be testable with sample complexity $s \sim 1/\epsilon$?



If G is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

Proof Sketch (testability => container lemma)

Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

Proof Sketch (testability => container lemma)

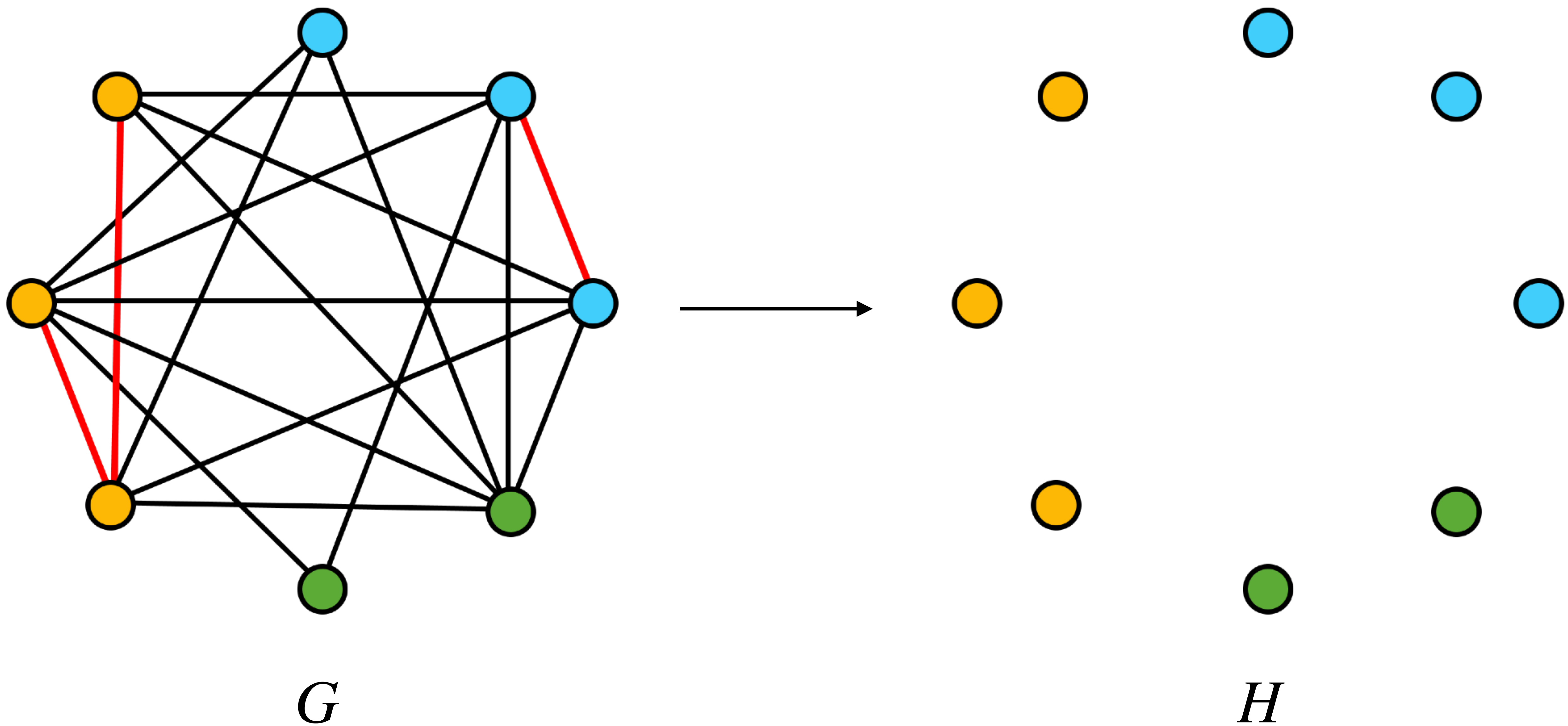
If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs

Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

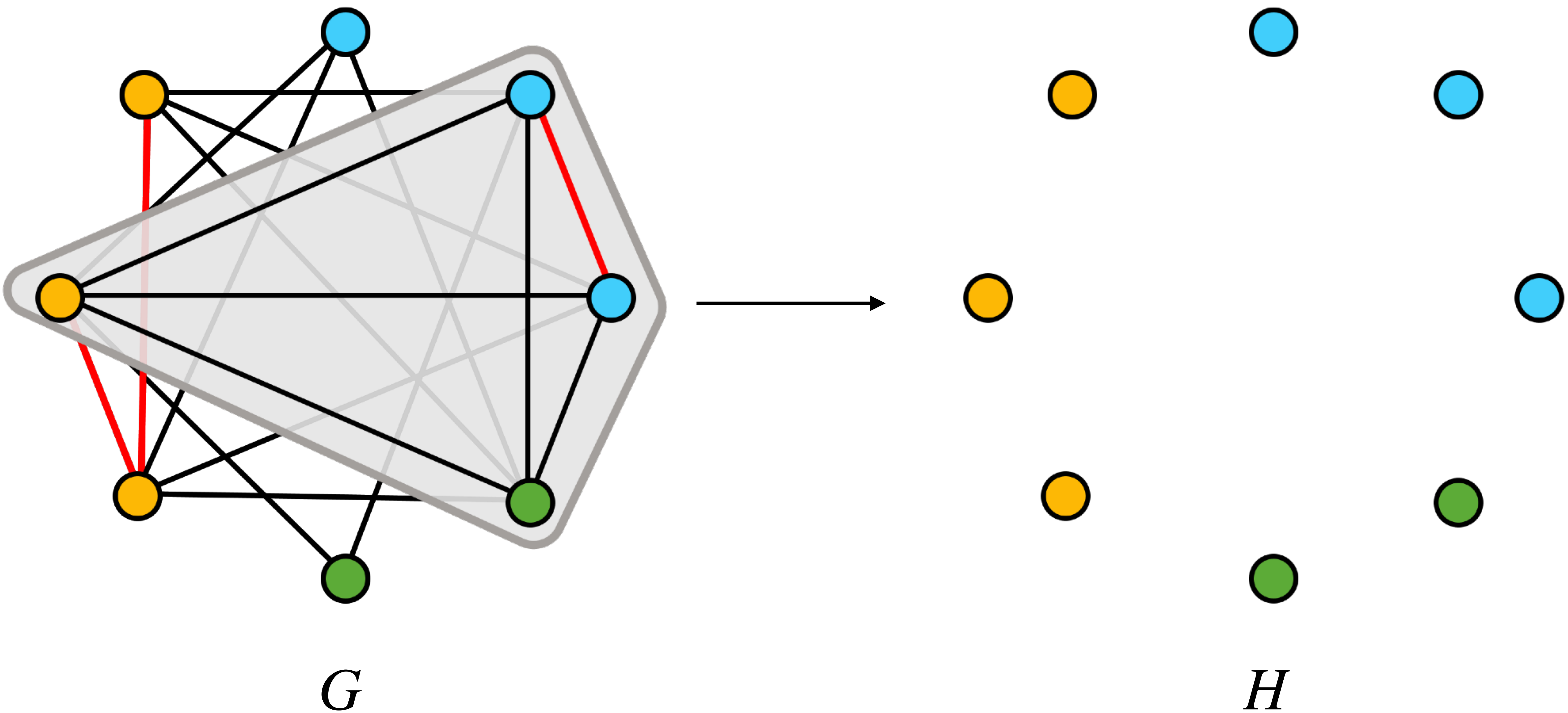
Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs



Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

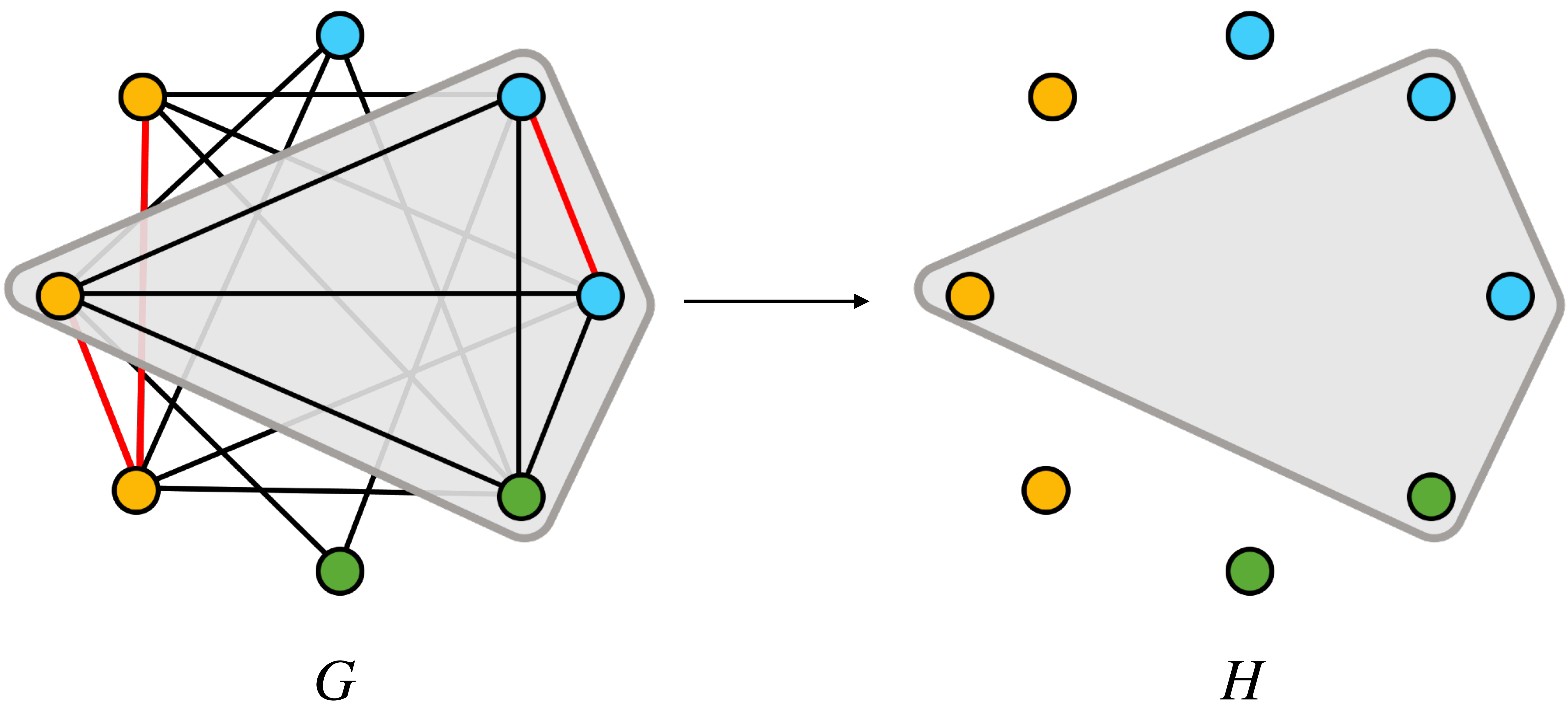
Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs



Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

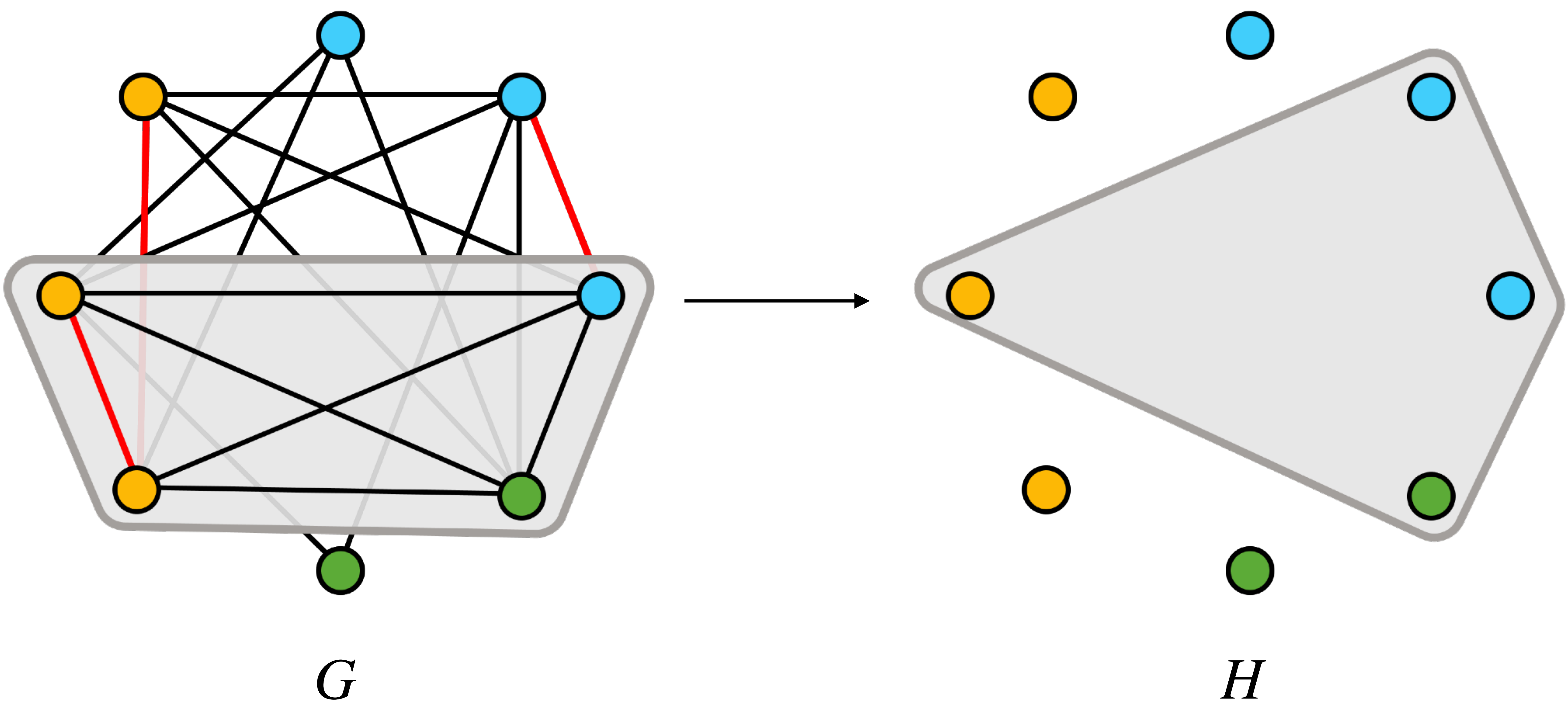
Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs



Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

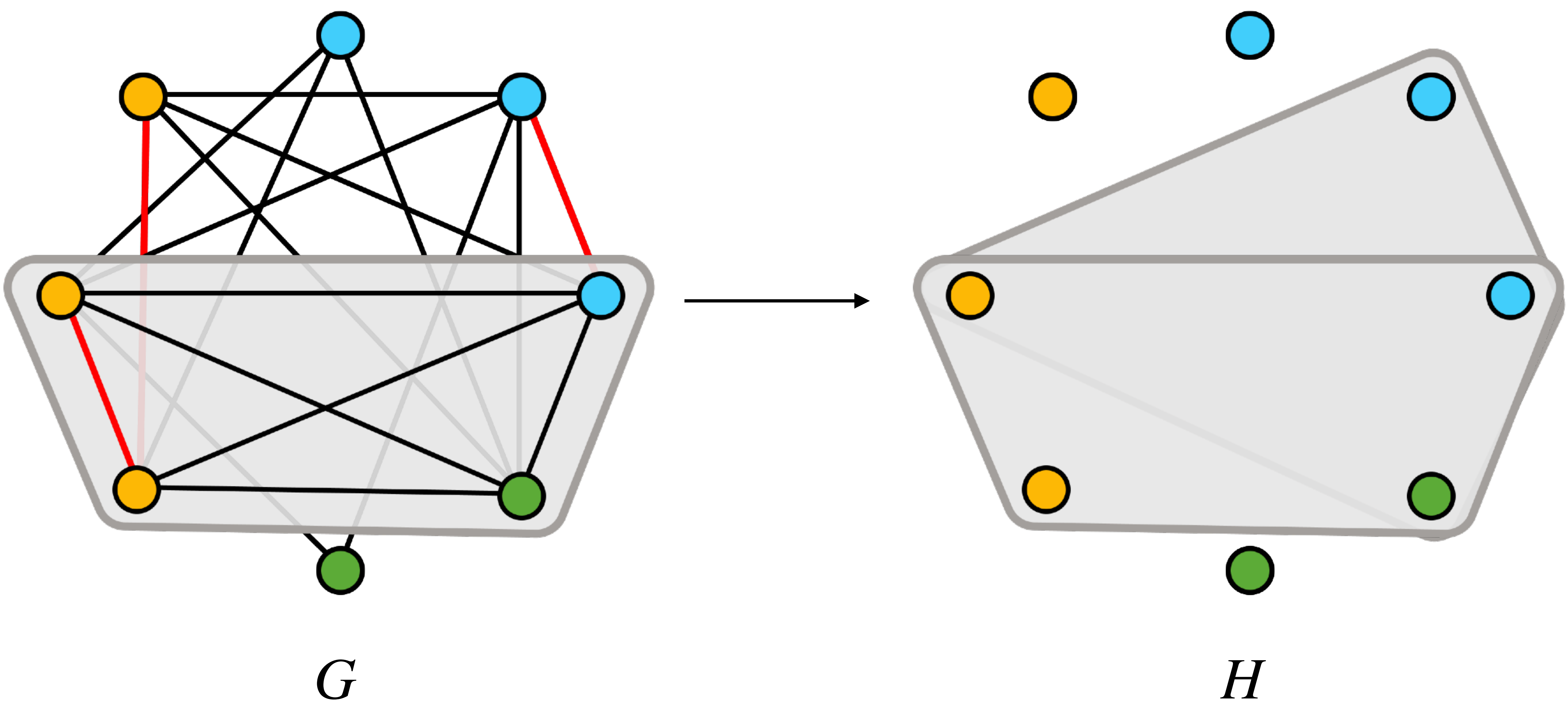
Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs



Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

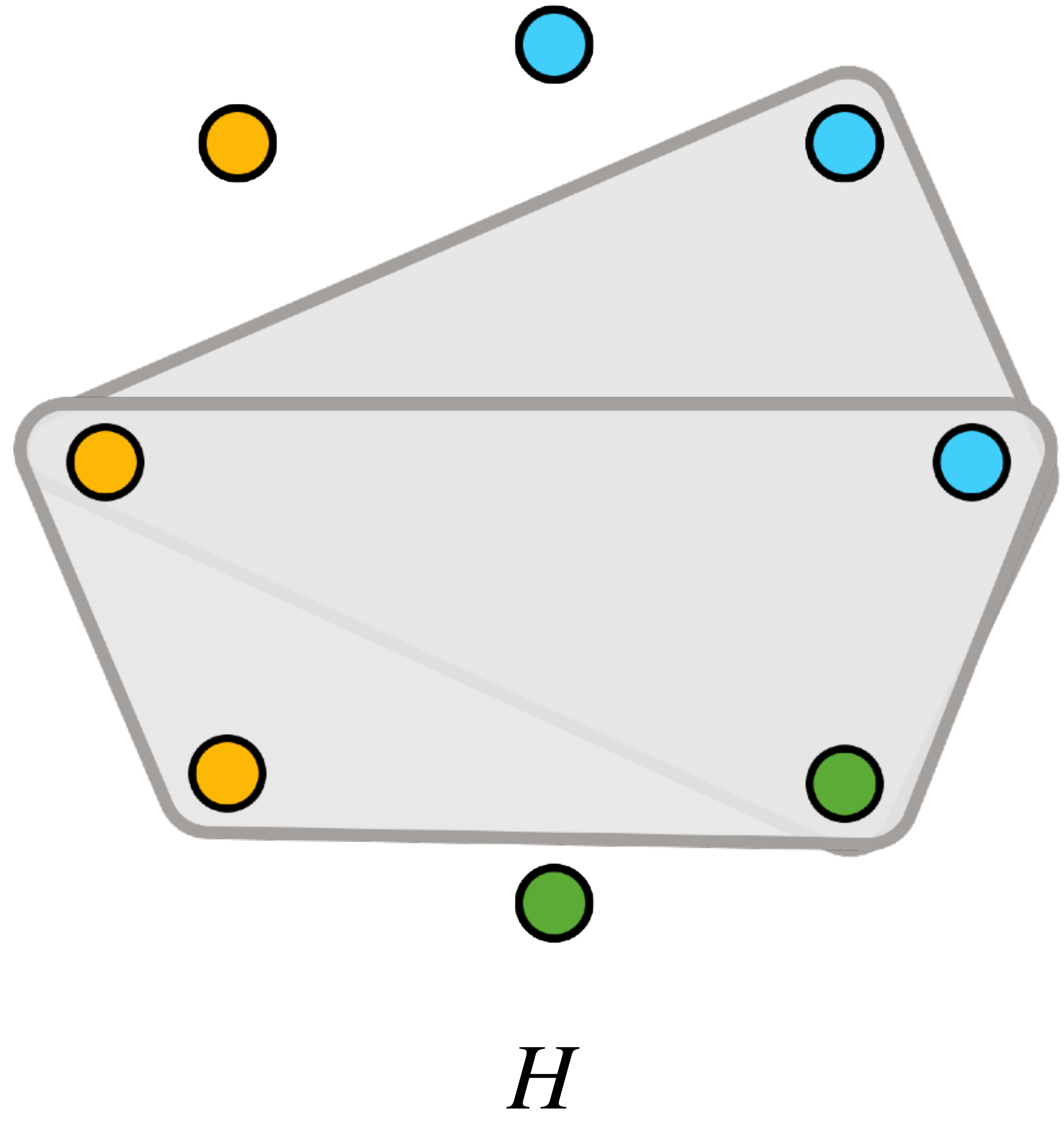
Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs



Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs

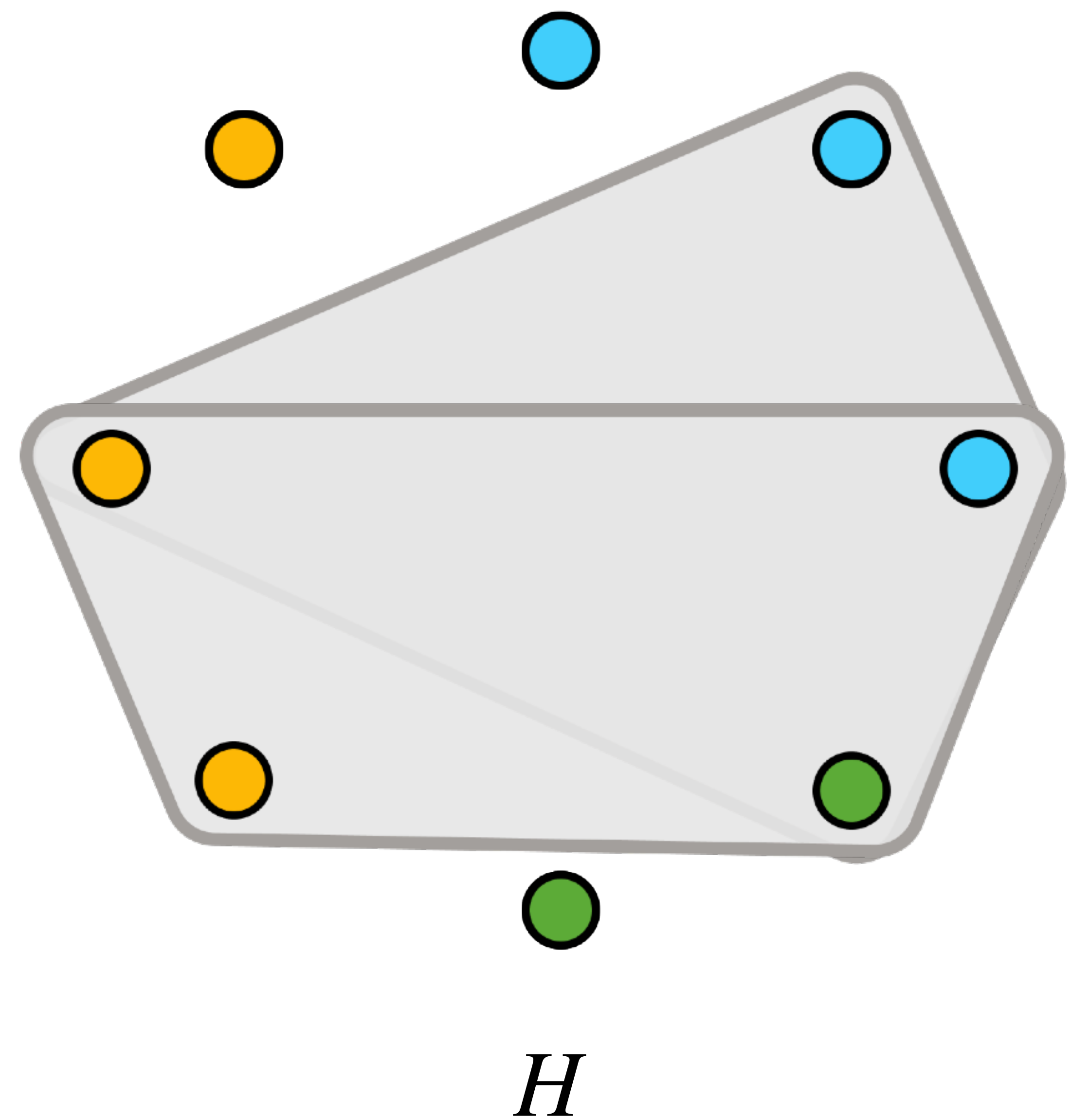


Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs

Observation: 3-colorable subgraphs in G are independent sets in H



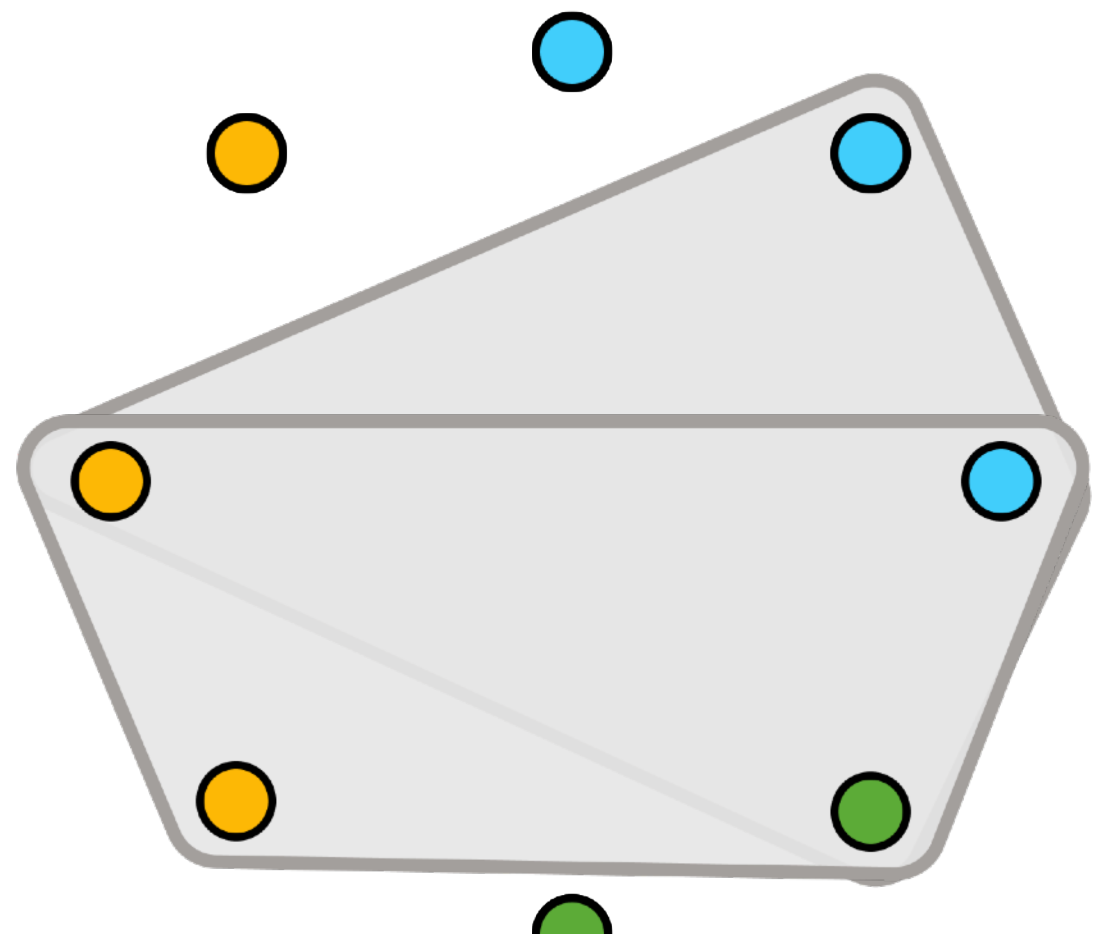
Proof Sketch (testability => container lemma)

If $G = (V, E)$ is ϵ-far from 3-colorable then a random induced subgraph on s vertices is not 3-colorable whp.

Idea: construct s-uniform hypergraph H on vertex set V with edges corresponding to sets of s vertices that form non-3-colorable subgraphs

Observation: 3-colorable subgraphs in G are independent sets in H

A container lemma for independent sets in H gives a container lemma for 3-colorable subgraphs in G



H

Proof Sketch (container lemma => testability)

Proof Sketch (container lemma => testability)

Assume Π has an ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$.

Proof Sketch (container lemma => testability)

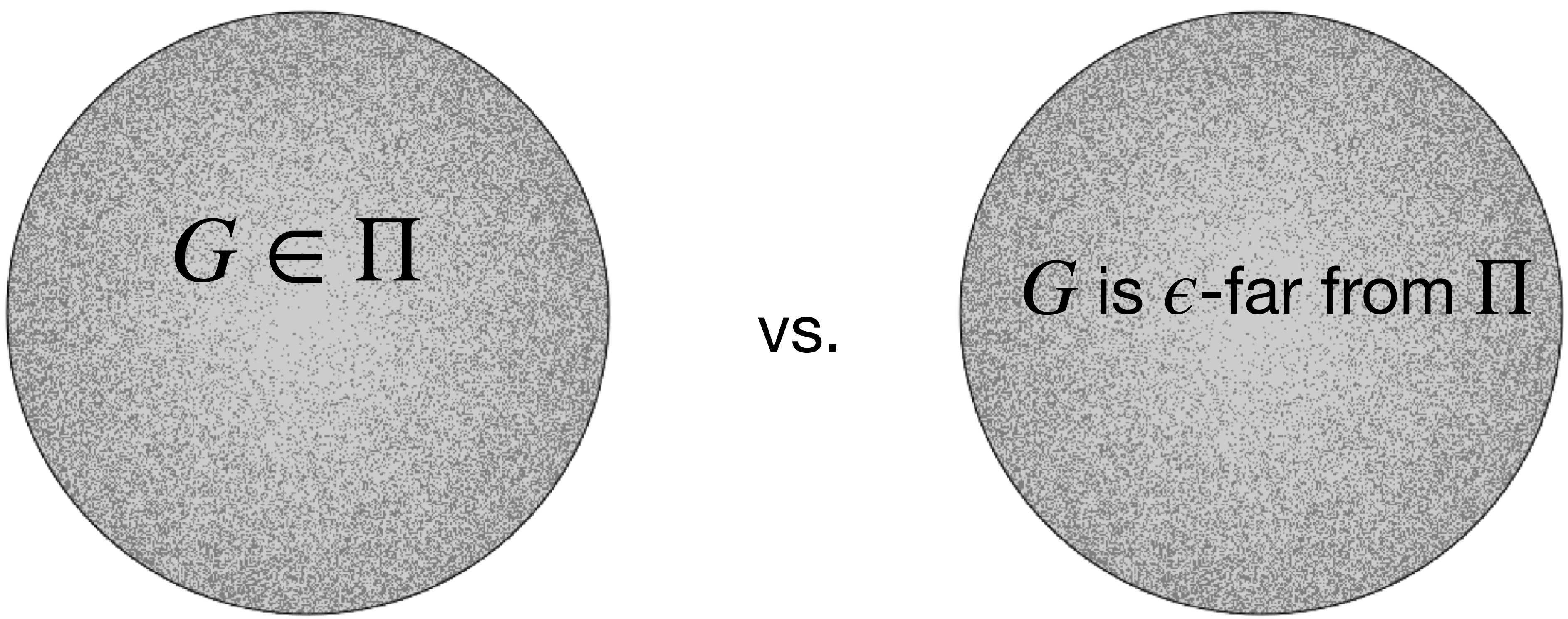
Assume Π has an ϵ-container lemma with fingerprint size f and container size $(1 - \alpha)$.

Consider the tester that takes a random sample S of s vertices and checks if $G[S]$ satisfies Π.

Proof Sketch (container lemma => testability)

Assume Π has an ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$.

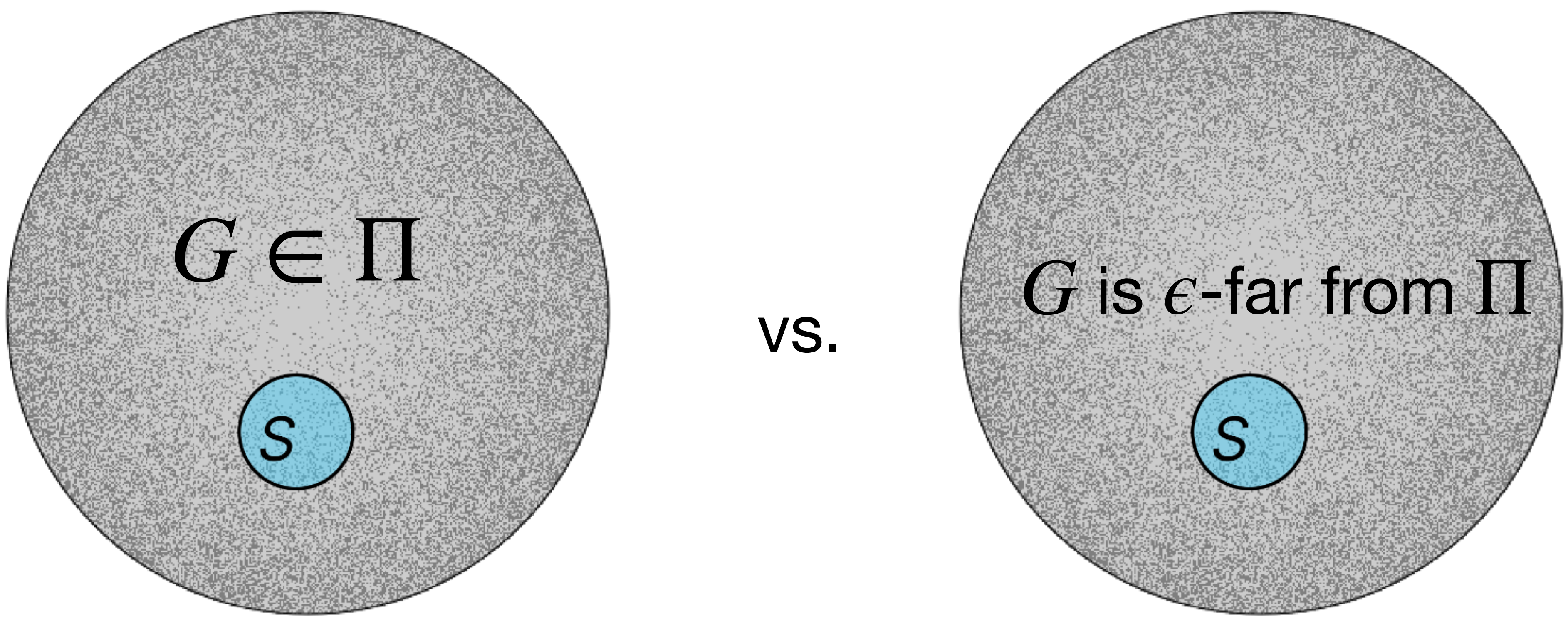
Consider the tester that takes a random sample S of s vertices and checks if $G[S]$ satisfies Π.



Proof Sketch (container lemma => testability)

Assume Π has an ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$.

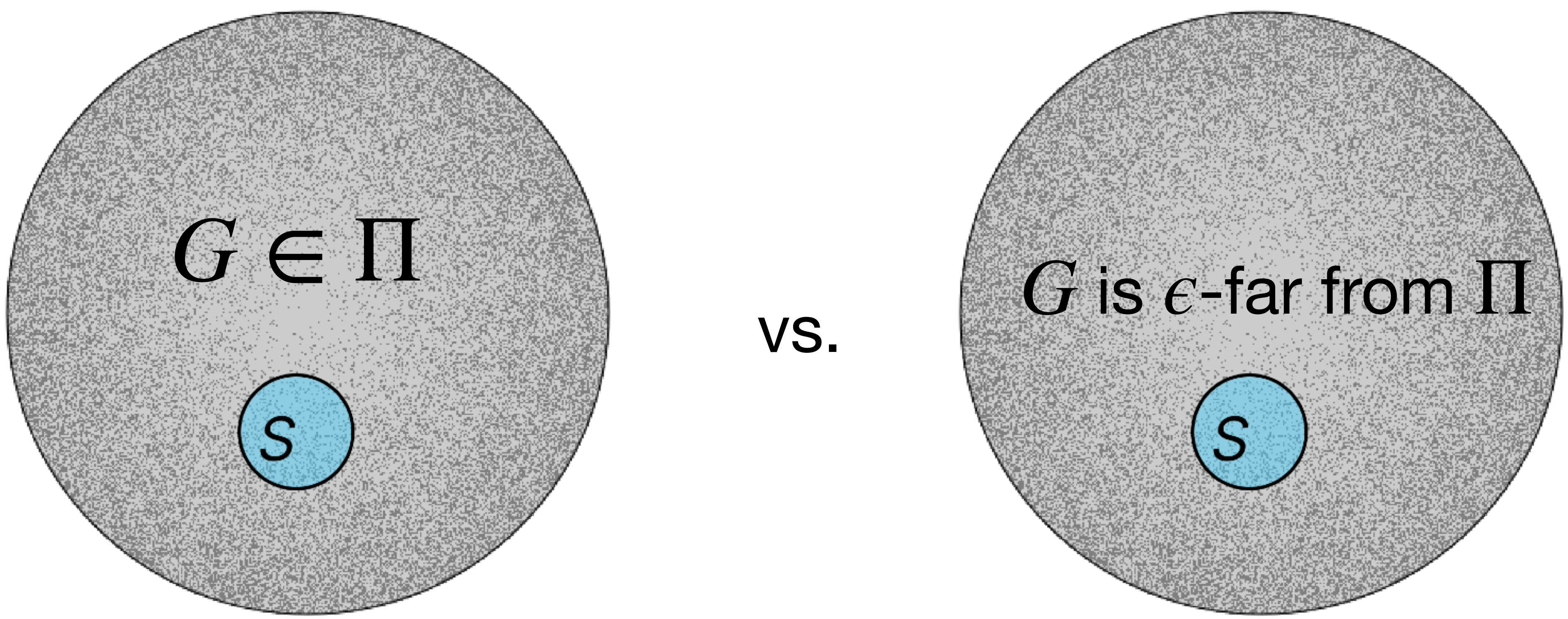
Consider the tester that takes a random sample S of s vertices and checks if $G[S]$ satisfies Π.



Proof Sketch (container lemma => testability)

Assume Π has an ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$.

Consider the tester that takes a random sample S of s vertices and checks if $G[S]$ satisfies Π.



Key challenge: show that if G is ϵ-far from Π then $G[S] \in \Pi$ with only small probability.

Proof Sketch (container lemma => testability)

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1 - \alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

First Attempt: same union bound as emptiness testing example

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

First Attempt: same union bound as emptiness testing example

$$\Pr[G[S] \in \Pi] \leq \sum_{F \in \mathcal{F}} \Pr[S \subseteq C(F)]$$

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

First Attempt: same union bound as emptiness testing example

$$\Pr[G[S] \in \Pi] \leq \sum_{F \in \mathcal{F}} \Pr[S \subseteq C(F)]$$
$$\leq \sum_{F \in \mathcal{F}} \left(\frac{|C(F)|}{n} \right)^{|S|}$$

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

First Attempt: same union bound as emptiness testing example

$$\begin{aligned}\Pr[G[S] \in \Pi] &\le \sum_{F \in \mathcal{F}} \Pr[S \subseteq C(F)] \\ &\le \sum_{F \in \mathcal{F}} \left(\frac{|C(F)|}{n}\right)^{|S|} \\ &\lesssim \binom{n}{f}(1-\alpha)^{|S|}\end{aligned}$$

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1 - \alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

First Attempt: same union bound as emptiness testing example

$$\begin{aligned}
\Pr[G[S] \in \Pi] &\leq \sum_{F \in \mathcal{F}} \Pr[S \subseteq C(F)] \\
&\leq \sum_{F \in \mathcal{F}} \left(\frac{|C(F)|}{n} \right)^{|S|} \\
&\lesssim \binom{n}{f} (1-\alpha)^{|S|} \\
&\leq \exp(f \log(n) - \alpha |S|)
\end{aligned}$$

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

First Attempt: same union bound as emptiness testing example

$$
\begin{aligned}
\Pr[G[S] \in \Pi] &\leq \sum_{F \in \mathcal{F}} \Pr[S \subseteq C(F)] \\
&\leq \sum_{F \in \mathcal{F}} \left(\frac{|C(F)|}{n} \right)^{|S|} \\
&\lesssim \binom{n}{f} (1-\alpha)^{|S|} \\
&\leq \exp(f \log(n) - \alpha |S|) \\
&= o(1) \qquad \text{as long as } |S| = \Omega(f \log(n)/\alpha)
\end{aligned}
$$

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

Second Attempt:

$$\Pr[G[S] \in \Pi] \leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S \subseteq C(F)]$$

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

Second Attempt:

$$\Pr[G[S] \in \Pi] \leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S \subseteq C(F)]$$

$$\leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S] \cdot \Pr[S \subseteq C(F) \,|\, F \subseteq S]$$

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

Second Attempt:

$$\Pr[G[S] \in \Pi] \leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S \subseteq C(F)]$$

$$\leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S] \cdot \Pr[S \subseteq C(F) \mid F \subseteq S]$$

$$\leq \sum_{r=1}^{f} \sum_{\substack{F \in \mathcal{F} \\ |F| = r}} \frac{\binom{n-r}{|S|-r}}{\binom{n}{|S|}} \cdot \left(\frac{|C(F)|}{n}\right)^{|S|-r}$$

Proof Sketch (container lemma => testability)

Assume that Π has ϵ-container lemma with fingerprint size f and container size $(1-\alpha)$

Key challenge: show that if G is ϵ-far then $G[S] \in \Pi$ with only small probability

Approach: every Π-subgraph is in a container, so union bound over containers

Second Attempt:

$$\Pr[G[S] \in \Pi] \leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S \subseteq C(F)]$$

$$\leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S] \cdot \Pr[S \subseteq C(F) \,|\, F \subseteq S]$$

$$\leq \sum_{r=1}^{f} \sum_{\substack{F \in \mathcal{F} \\ |F| = r}} \frac{\binom{n-r}{|S|-r}}{\binom{n}{|S|}} \cdot \left(\frac{|C(F)|}{n}\right)^{|S|-r}$$

$$\leq \sum_{r=1}^{f} \binom{|S|}{r} (1-\alpha)^{|S|-r} \leq \sum_{r=1}^{f} \exp(r \log(|S|) - \alpha |S|) = o(1)$$

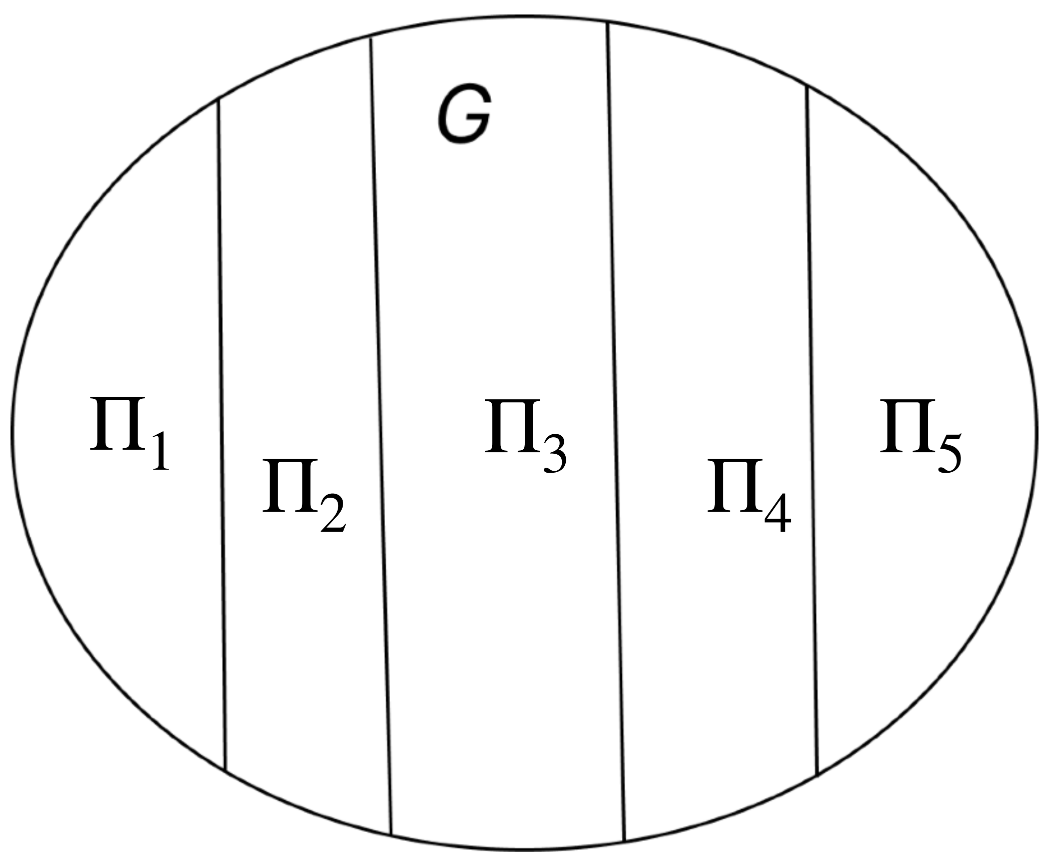
As long as $|S| = \tilde{\Omega}(f/\alpha)$

An Application of the Theorem

Application: Testing Partitions of Hereditary Properties

Corollary: Let $\Pi_1, \ldots, \Pi_k$ be hereditary graph properties that are ϵ-testable with one sided error with sample complexity $s_1(\epsilon), \ldots, s_k(\epsilon)$ respectively. Let Π be the graph property of having a partition into k parts such that the i-th part satisfies Π_i. Then Π is ϵ-testable with sample complexity roughly

$$\max(s_i(\epsilon/k))^2 \cdot k/\epsilon$$



Application: Testing Partitions of Hereditary Properties

Corollary: Let $\Pi_1, \ldots, \Pi_k$ be hereditary graph properties that are ϵ-testable with one sided error with sample complexity $s_1(\epsilon), \ldots, s_k(\epsilon)$ respectively. Let Π be the graph property of having a partition into k parts such that the i-th part satisfies Π_i. Then Π is ϵ-testable with sample complexity roughly

$$\max(s_i(\epsilon/k))^2 \cdot k/\epsilon$$

Application: Testing Partitions of Hereditary Properties

Corollary: Let $\Pi_1, \ldots, \Pi_k$ be hereditary graph properties that are ϵ-testable with one sided error with sample complexity $s_1(\epsilon), \ldots, s_k(\epsilon)$ respectively. Let Π be the graph property of having a partition into k parts such that the i-th part satisfies Π_i. Then Π is ϵ-testable with sample complexity roughly

$$\max(s_i(\epsilon/k))^2 \cdot k/\epsilon$$

Example 1: If Π_i is the empty property for all $i \in [k]$ then Π is the property of being k-colorable. Corollary gives sample complexity bound of k^3/ϵ^3 since emptiness property is testable with $1/\epsilon$ samples

Application: Testing Partitions of Hereditary Properties

Corollary: Let $\Pi_1, \ldots, \Pi_k$ be hereditary graph properties that are ϵ-testable with one sided error with sample complexity $s_1(\epsilon), \ldots, s_k(\epsilon)$ respectively. Let Π be the graph property of having a partition into k parts such that the i-th part satisfies Π_i. Then Π is ϵ-testable with sample complexity roughly

$$\max(s_i(\epsilon/k))^2 \cdot k/\epsilon$$

Example 1: If Π_i is the empty property for all $i \in [k]$ then Π is the property of being k-colorable. Corollary gives sample complexity bound of k^3/ϵ^3 since emptiness property is testable with $1/\epsilon$ samples

Example 2: If Π_i is the property of being a DAG (directed acyclic graph) for all $i \in [k]$ then Π is the property of having dichromatic number k. Corollary gives sample complexity bound of k^3/ϵ^3 since DAG property is testable with $1/\epsilon$ samples ([Bender, Ron '02])

Application: Testing Partitions of Hereditary Properties

Application: Testing Partitions of Hereditary Properties

Proof:

We will show that Π has an ϵ-container lemma with fingerprint size $k \max \left(s_i \left(\frac{\epsilon}{2k} \right) \right)^2$ and container size $(1 - \epsilon/2)$.

Application: Testing Partitions of Hereditary Properties

Proof:

We will show that Π has an ϵ-container lemma with fingerprint size $k \max \left(s_i \left(\frac{\epsilon}{2k} \right) \right)^2$ and container size $(1 - \epsilon/2)$.

- Each Π_i has ϵ-container lemma with fingerprint size bounded by $s_i^2(\epsilon)$ - by Theorem
- If G ϵ-far from Π then G is ϵ-far from each Π_i, and so G is $\epsilon/(2k)$-far from each Π_i.
- So, applying the $\epsilon/(2k)$-container lemmas, we get containers for the Π_i subgraphs, and each container is $\epsilon/(2k)$-close to Π_i.

Application: Testing Partitions of Hereditary Properties

Proof:

We will show that Π has an ϵ-container lemma with fingerprint size $k \max \left(s_i \left(\frac{\epsilon}{2k} \right) \right)^2$ and container size $(1 - \epsilon/2)$.

- Each Π_i has ϵ-container lemma with fingerprint size bounded by $s_i^2(\epsilon)$ - by Theorem
- If G ϵ-far from Π then G is ϵ-far from each Π_i, and so G is $\epsilon/(2k)$-far from each Π_i.
- So, applying the $\epsilon/(2k)$-container lemmas, we get containers for the Π_i subgraphs, and each container is $\epsilon/(2k)$-close to Π_i.
- Take the union of one container from each lemma. For each k-tuple of containers the union of the containers is close to Π because each of the containers is $\epsilon/(2k)$ -close to Π_i. Hence the container size is at most $(1 - \epsilon/2)n$.
- The fingerprints are of size $k \max \left(s_i \left(\frac{\epsilon}{2k} \right) \right)^2$.

Application: Testing Partitions of Hereditary Properties

Proof:

We will show that Π has an ϵ-container lemma with fingerprint size $k \max\left(s_i\left(\frac{\epsilon}{2k}\right)\right)^2$ and container size $(1 - \epsilon/2)$.

- Each Π_i has ϵ-container lemma with fingerprint size bounded by $s_i^2(\epsilon)$ - by Theorem
- If G ϵ-far from Π then G is ϵ-far from each Π_i, and so G is $\epsilon/(2k)$-far from each Π_i.
- So, applying the $\epsilon/(2k)$-container lemmas, we get containers for the Π_i subgraphs, and each container is $\epsilon/(2k)$-close to Π_i.
- Take the union of one container from each lemma. For each k-tuple of containers the union of the containers is close to Π because each of the containers is $\epsilon/(2k)$ -close to Π_i. Hence the container size is at most $(1 - \epsilon/2)n$.
- The fingerprints are of size $k \max\left(s_i\left(\frac{\epsilon}{2k}\right)\right)^2$.
- Again applying the Theorem, we get testability with sample complexity roughly

$$k \max\left(s_i\left(\frac{\epsilon}{2k}\right)\right)^2 / \epsilon$$

Summary and Open Questions

Open Questions

Summary: We show that for any hereditary graph property Π:

Π is ϵ-testable with one sided error and sample complexity s $\quad\Longleftrightarrow\quad$ Π has an ϵ-container lemma parameterized by $\mathrm{poly}(s)$

And used this to show the testability of partitions of hereditary properties

Open Questions

Summary: We show that for any hereditary graph property Π:

Π is ϵ-testable with one sided error and sample complexity s $\Leftrightarrow$ Π has an ϵ-container lemma parameterized by $\mathrm{poly}(s)$

And used this to show the testability of partitions of hereditary properties

---

Question 1: What about general (non-hereditary) graph properties? (Similar to how [Alon, Fischer, Newman, Shapira '09] generalized the [Alon, Shapira '08] result)

Open Questions

Summary: We show that for any hereditary graph property Π:

Π is ϵ-testable with one sided error and sample complexity s

$\Leftrightarrow$

Π has an ϵ-container lemma parameterized by $\mathrm{poly}(s)$

And used this to show the testability of partitions of hereditary properties

---

Question 1: What about general (non-hereditary) graph properties? (Similar to how [Alon, Fischer, Newman, Shapira '09] generalized the [Alon, Shapira '08] result)

Question 2: Can we remove the polynomial loss in the theorem?

Open Questions

Summary: We show that for any hereditary graph property Π:

Π is ϵ-testable with one sided error and sample complexity s

$\Longleftrightarrow$

Π has an ϵ-container lemma parameterized by $\mathrm{poly}(s)$

And used this to show the testability of partitions of hereditary properties

---

Question 1: What about general (non-hereditary) graph properties? (Similar to how [Alon, Fischer, Newman, Shapira '09] generalized the [Alon, Shapira '08] result)

Question 2: Can we remove the polynomial loss in the theorem?

Question 3: Can we use this to test general graph partition properties? (The challenging part is testing densities)

Open Questions

Summary: We show that for any hereditary graph property Π:

Π is ϵ-testable with one sided error and sample complexity s

$\Longleftrightarrow$

Π has an ϵ-container lemma parameterized by $\mathrm{poly}(s)$

And used this to show the testability of partitions of hereditary properties

---

Question 1: What about general (non-hereditary) graph properties? (Similar to how [Alon, Fischer, Newman, Shapira '09] generalized the [Alon, Shapira '08] result)

Question 2: Can we remove the polynomial loss in the theorem?

Question 3: Can we use this to test general graph partition properties? (The challenging part is testing densities)

Thank you!