On Testing Properties in Directed Graphs

Artur Czumaj

DIMAP and Department of Computer Science

University of Warwick

Joint work with Pan Peng and Christian Sohler (TU Dortmund)

Dealing with "BigData" in Graphs

- We want to process graphs quickly
  - Detect basic properties
  - Analyze their structure

- For large graphs, by "quickly" we often would mean: in time *constant* or *sublinear* in the size of the graph

Dealing with "BigData" in Graphs

One approach:

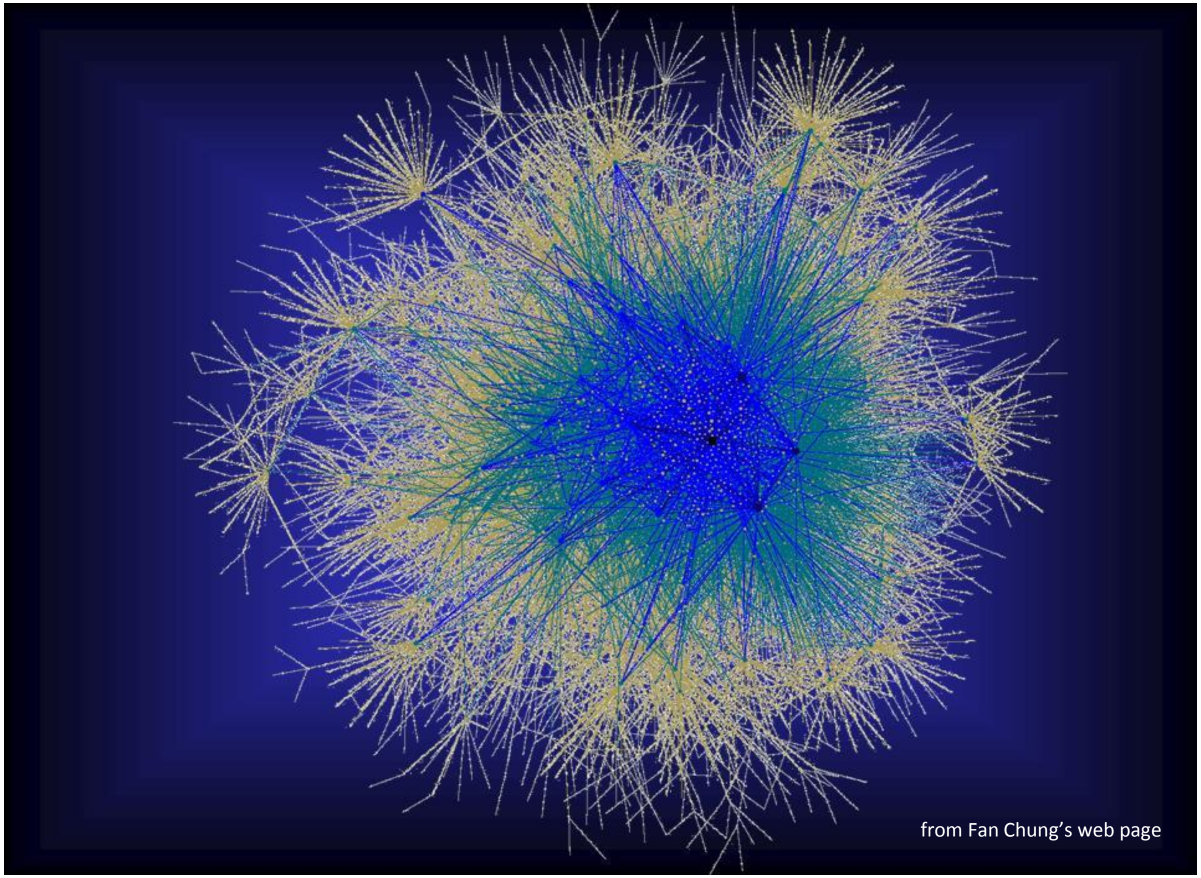
- How to test basic properties of graphs
  in the framework of **property testing**

Framework of property testing

- We cannot quickly give 100% precise answer
- We need to approximate

- Distinguish graphs that have specific property from those that are far from having the property

Fast Testing of Graph Properties



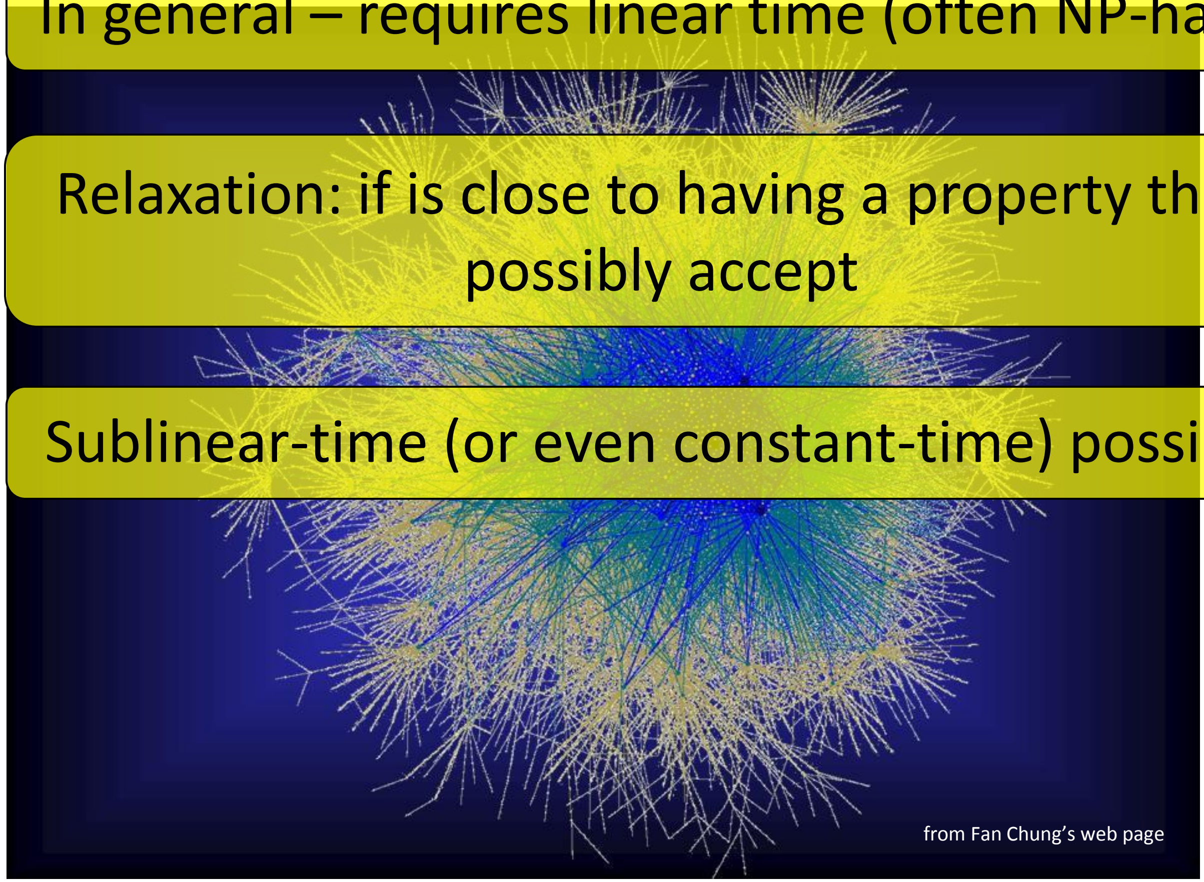
- Does this graph have a clique of size 11?
- Does it have a given H as its subgraph?
- Is this graph planar?
- Is it bipartite?
- Is it k-colorable?
- Does it have good expansion?
- Does it have good clustering?

Fast Testing of Graph Properties

In general – requires linear time (often NP-hard)

Relaxation: if is close to having a property then possibly accept

Sublinear-time (or even constant-time) possible



- Does this graph have a clique of size 11?
- Does it have a given H as its subgraph?
- Is this graph planar?
- Is it bipartite?
- Is it k-colorable?
- Does it have good expansion?
- Does it have good clustering?

Testing properties of graphs

Input:

- graph property P;
- proximity parameter ε;
- input graph $G = (V, E)$ of maximum degree d.

Output:

- if G satisfies property P then ACCEPT
- if G is ε–far from having property P then REJECT

Testing properties of graphs

Input:

- graph property P;
- proximity parameter ε;
- input graph $G = (V, E)$ of maximum degree d.

Output:

- if G satisfies property P then ACCEPT
- if G is ε–far from having property P then REJECT

G is ε–far from satisfying P if one has to modify $\leq d|V|$ edges of G to obtain a graph satisfying P

Testing properties of graphs

Input:

- graph property P;
- proximity parameter ε;
- input graph $G = (V, E)$ of maximum degree d.

Output:

- if G satisfies property P then ACCEPT
- if G is ε–far from having property P then REJECT

- if we can err only for REJECTION then one-sided error
- if we can also err for ACCEPTs then two-sided error

Fast Testing of Graph Properties

- Started with Rubinfeld-Sudan (1996) and Goldreich-Goldwasser-Ron (1998)
- Now we know a lot
  - If G is dense, given as an oracle to adjacency matrix, then every hereditary property can be tested in constant time
  - If G is sparse, given as an oracle to adjacency list, then many properties can be tested in constant time, many can be tested in sublinear time
  - If G is directed then … essentially nothing is known
    - unless there is a trivial reduction to undirected graphs

Fast Testing of Digraph Properties

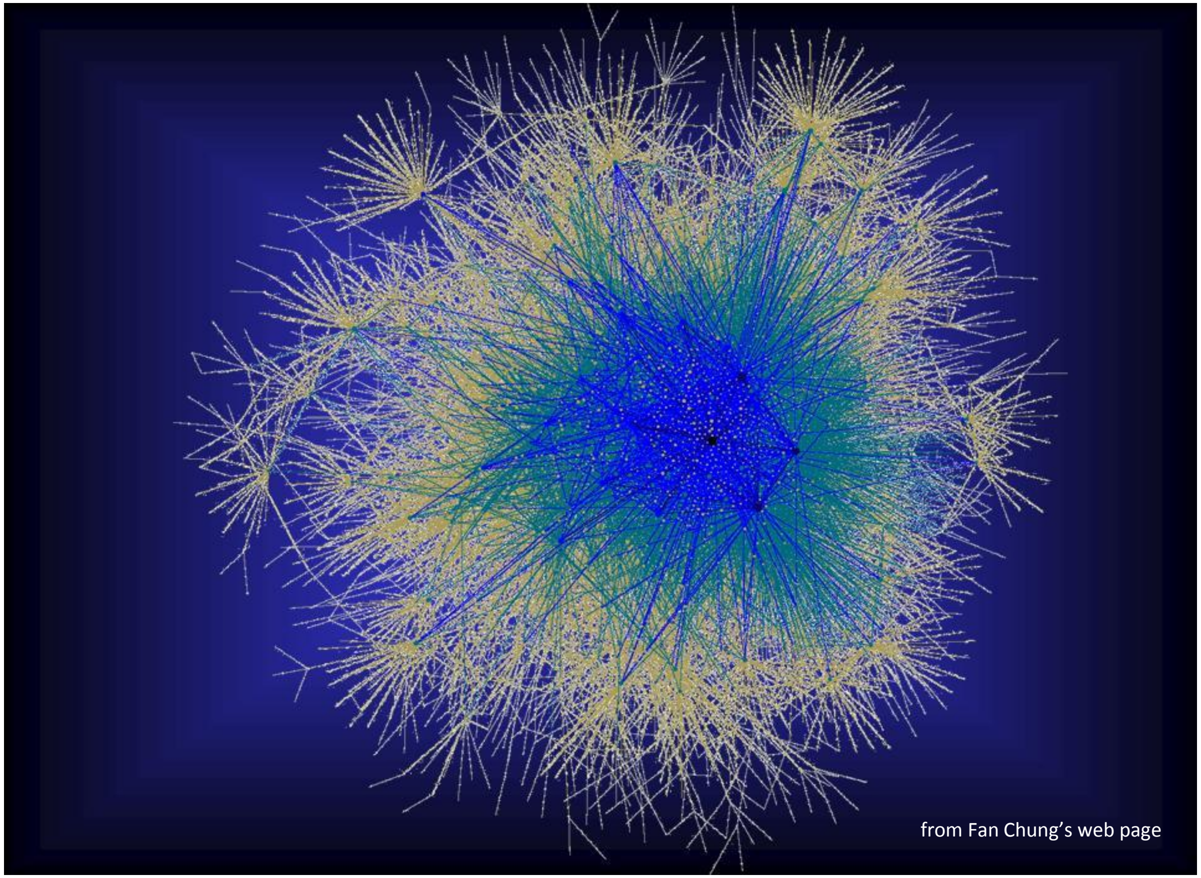
Models introduced by Bender-Ron (2002):

- Digraphs with bounded maximum in- and out-degrees
- Oracle with access to adjacency list
- Two main models:
  - Bidirectional: outgoing and incoming edges
    - shares properties of undirected graphs;
    - not suitable in many scenarios/applications
  - One-directional: access to outgoing edges only
    - major difference wrt undirected graphs
    - more natural in many scenarios/applications

Sometimes very fast

More challenging

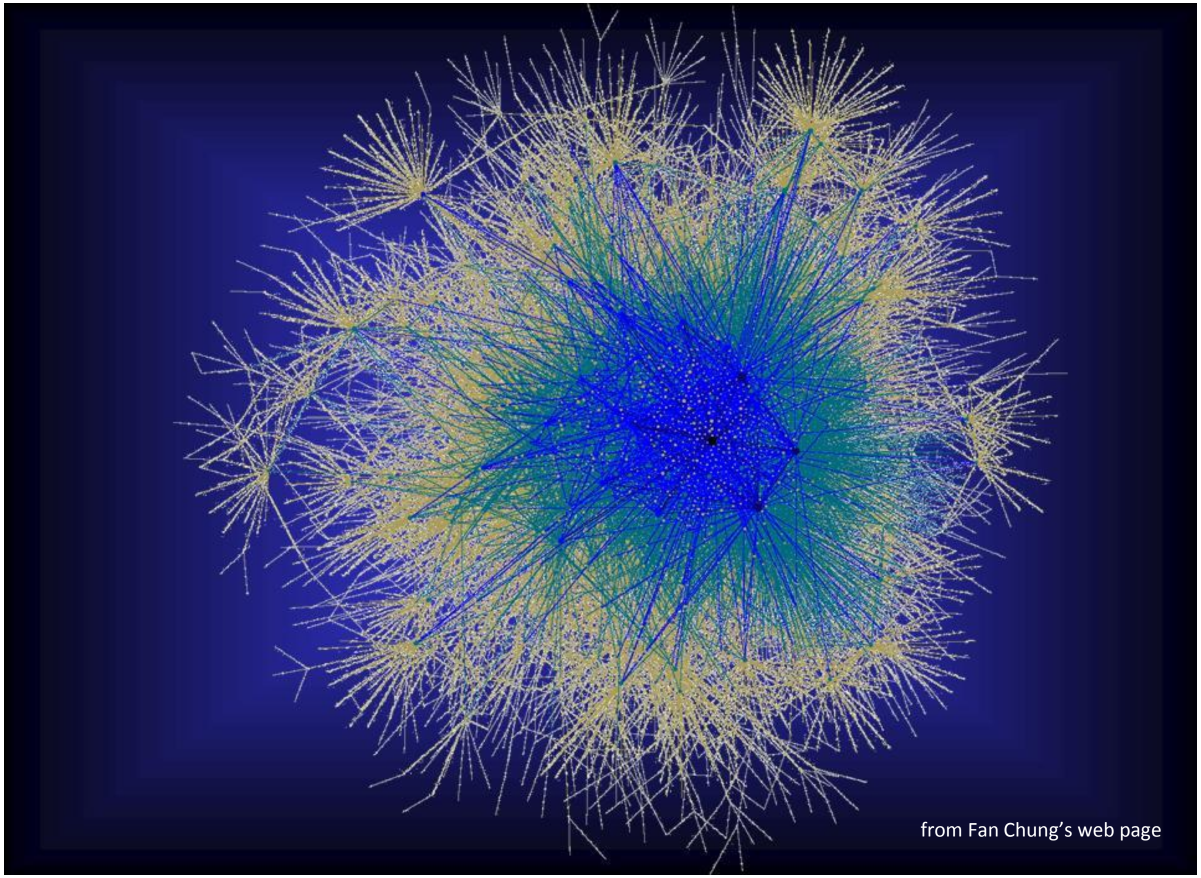
Big networks



- Is it weakly connected?
  (or close to it)
- Is it planar?
  (or close to it)

If we have access to both directional edges then this reduces to a problem in undirected graphs (which we understand well)

Big networks



- Is it strongly connected?
  (or close to it)
- Is it acyclic?
  (or close to it)
- Is it C_{33}-free?
  (or close to it)

Highly non-trivial if we have no access to incoming edges
For example: we cannot easily check if a node has in-degree 0

OBJECTIVE: Study the dependency between the models

There is a tester for property P with constant query time in bidirectional model

⬇

We can test P in one-directional model with sublinear $n^{1-\Omega_{\varepsilon,d}(1)}$ query time (in two-sided error model)

OBJECTIVE: Study the dependency between the models

There is a tester for property P with constant query time in bidirectional model

⇓

We can test P in one-directional model with sublinear $n^{1-\Omega_{\varepsilon,d}(1)}$ query time (in two-sided error model)

Application:
Every hyperfinite property can be tested with sublinear complexity in one-directional model

What is known for digraphs

Not much

What is known for digraphs

Strong connectivity

- Constant complexity in bidirectional model (Bender-Ron'02)
- One-directional queries:
  - requires $\Omega(\sqrt{n})$ complexity (Bender-Ron'02)
  - can be done with $n^{1-\Omega_{\varepsilon,d}(1)}$ complexity (Goldreich'11, Hellweg-Sohler'12)
  - requires $\Omega(n)$ complexity in one-sided-error model (Goldreich'11, Hellweg-Sohler'12)

What is known for digraphs

Bidirectional model:

- testing Eulerianity (Orenstein-Ron'11)
- testing k-edge-connectivity (Orenstein-Ron'11 ,Yoshida-Ito'10)
- testing k-vertex connectivity (Orenstein-Ron'11)
- acyclicity requires $\Omega(n^{1/3})$ queries (Bender-Ron'02)

- Testing H-freeness
  - constant complexity in bidirectional model (folklore)
  - $O(n^{1-1/k})$ complexity, where k is # of connected components of H with no incoming edge from another part of H (Hellweg-Sohler'12)
- 3-star-freeness:
  - requires $\Omega(n^{2/3})$ complexity (Hellweg-Sohler'12)

OBJECTIVE: Study the dependency between the models

There is a tester for property P with constant query time in bidirectional model

⇓

We can test P in one-directional model with sublinear $n^{1-\Omega_{\varepsilon,d}(1)}$ query time (in two-sided error model)

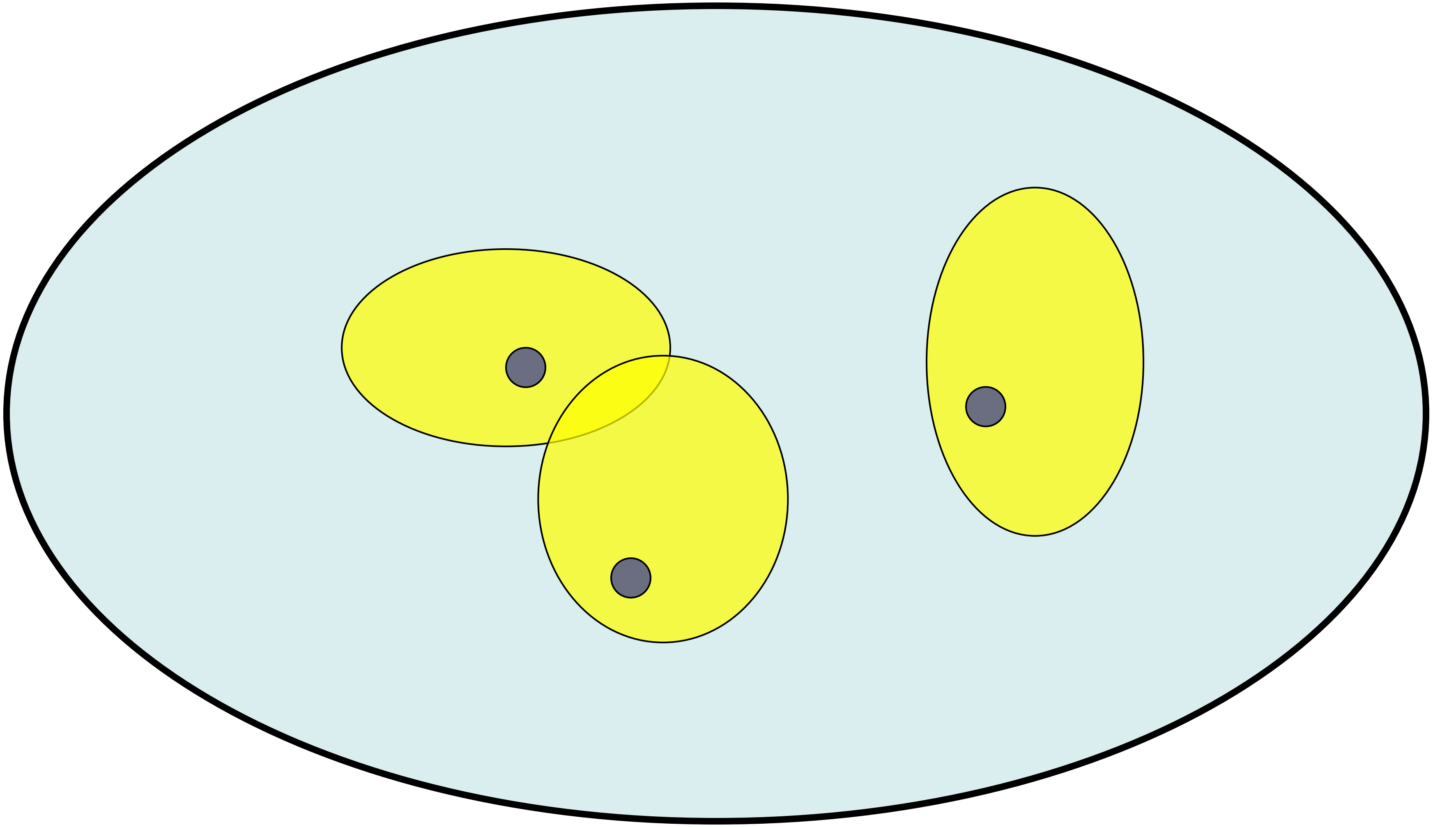
This cannot be improved much:

- two-sided error is required (cf. strong connectivity)
- $\Omega(n^{2/3})$ "simulation" slowdown is required (cf. 3-star-freeness)

Conjecture: bound is tight

Key ideas

What a constant-complexity tester in bidirectional model can do?



What a constant-complexity tester in bidirectional model can do?

Tester of complexity $q = q(\varepsilon, d, n)$

Cannot do more than

- Randomly sample q vertices
- Explore q neighborhood of the sampled vertices
  - neighborhood = using edges of either direction
- Accept or reject on the basis of the explored digraph

Key ideas

- We can characterize properties testable with constant number of queries ➔ canonical testers
- Canonical tester will do the following:
  - Samples a constant number of random vertices
  - Explores bounded-radius discs rooted at sampled vertices
  - Decides whether to accept or reject on the basis of a check if the explored digraph is isomorphic to any digraph from a forbidden collection of rooted discs

Key ideas

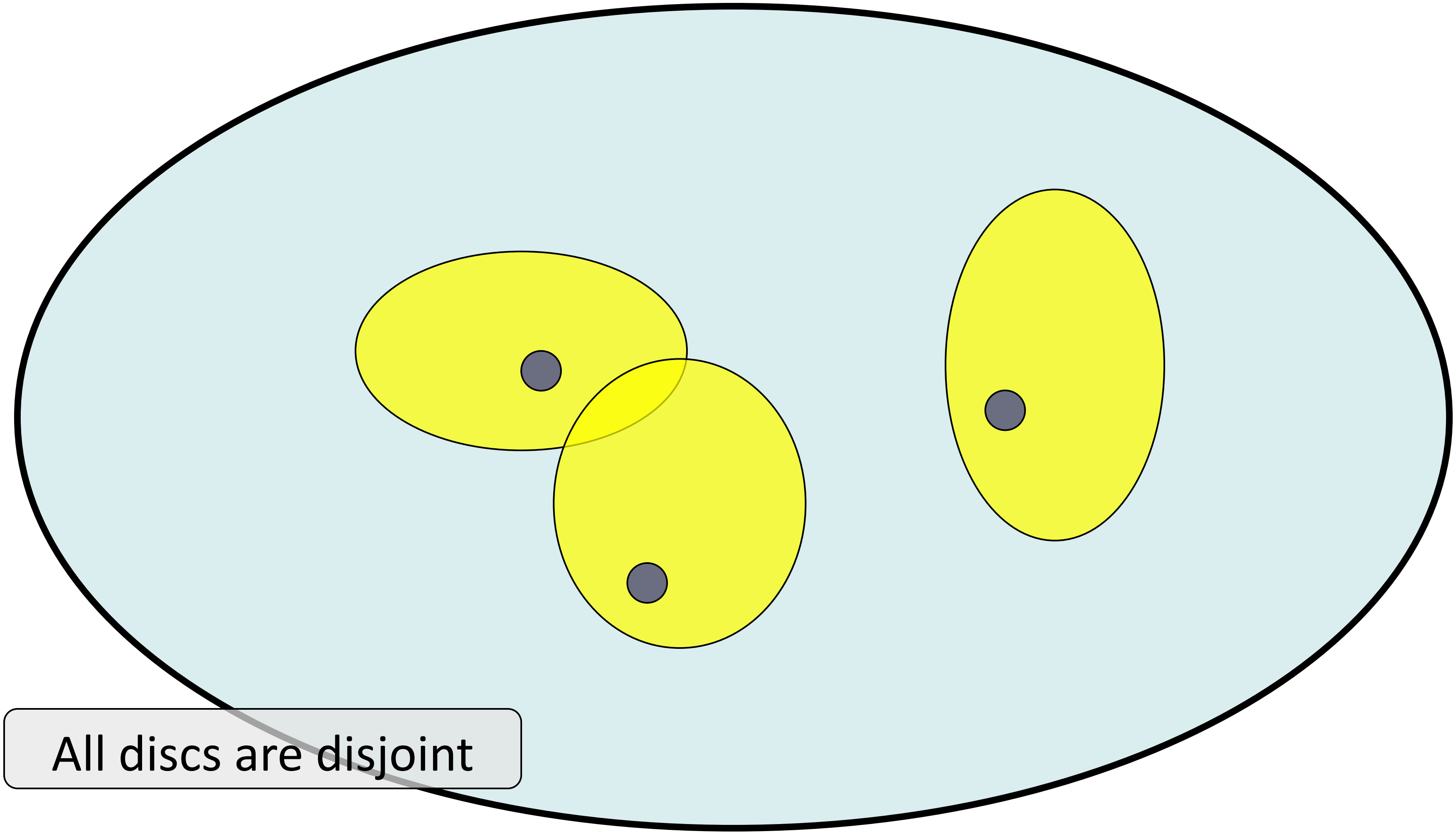
- We can characterize properties testable with constant number of queries ➔ canonical testers
- Canonical tester will do the following:
  - Samples a constant number of random vertices
  - Explores bounded-radius discs rooted at sampled vertices
  - Decides whether to accept or reject on the basis of a check if the explored digraph is isomorphic to any digraph from a forbidden collection of rooted discs

Further property:
* If G satisfies P then bounded-radius discs at randomly sampled vertices will be isomorphic to any element from the forbidden collection with prob $\leq 1/3$
* If G is ε–far, then the discs will be isomorphic with prob $\geq 2/3$

Key ideas

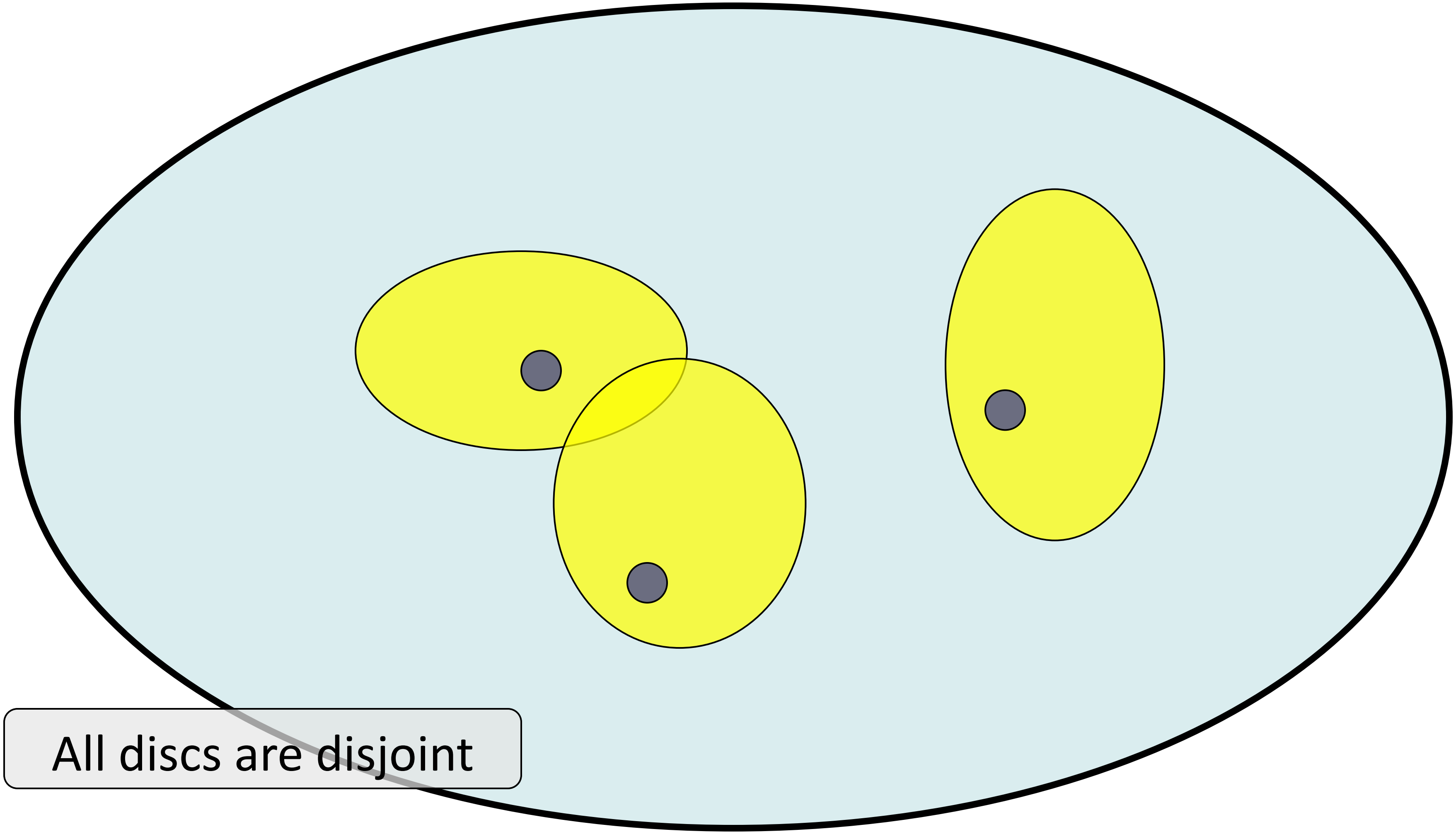
- We can characterize properties testable with constant number of queries ➔ canonical testers
- Goal of one-directional tester
  - Simulate canonical bidirectional testers
  - We want to "estimate" the structure of random q discs of (bidirectional) radius q

What a constant-complexity tester in bidirectional model can do?



All discs are disjoint

What a ~~constant-complexity~~ tester in ~~bidirectional~~ one-directional model can do?



All discs are disjoint

Key ideas

- We can characterize properties testable with constant number of queries ➔ canonical testers
- Goal of one-directional tester
  - Simulate canonical bidirectional testers
  - We want to "estimate" the structure of random q discs of (bidirectional) radius q
  - Let $H_{q,d}$ be the set of q rooted digraphs of (bidirectional) radius q of maximum in-/out-degree d
    - Note: $|H_{q,d}| = f(q, d, \varepsilon)$, and $q = q(\varepsilon, d)$ ➔ $|H_{q,d}| = O_{\varepsilon,d}(1)$
  - We can approximate the number of copies of any $H \in H_{q,d}$ in the input digraph G

Key ideas

- We can characterize properties testable with constant number of queries ➔ canonical testers
- Goal of one-directional tester
  - Simulate canonical bidirectional testers
  - We want to “estimate” the structure of random q discs of (bidirectional) radius q
  - By randomly sampling $n^{1-\Omega_{\varepsilon,d}(1)}$ edges, we can approximate well the number of occurrences of any $H \in H_{q,d}$ in the input digraph G

Key ideas

- We can characterize properties testable with constant number of queries ➔ canonical testers
- Goal of one-directional tester
  - Simulate canonical bidirectional testers
  - We want to "estimate" the structure of random q discs of (bidirectional) radius q
  - By randomly sampling $n^{1-\Omega_{\varepsilon,d}(1)}$ edges, we can approximate well the number of occurrences of any $H \in H_{q,d}$ in the input digraph G

  ➔ We can simulate canonical bidirectional tester

OBJECTIVE: Study the dependency between the models

There is a tester for property P with constant query time in bidirectional model

⇓

We can test P in one-directional model with sublinear $n^{1-\Omega_{\varepsilon,d}(1)}$ query time (in two-sided error model)

Application:
Every hyperfinite property can be tested with sublinear complexity in one-directional model

Hyperfinite graphs and properties

- Graph is hyperfinite if we can remove small fraction of edges to split it into small connected components
  - E.g. bounded degree planar graphs, bounded degree graphs defined by a finite collection of forbidden minors

- Property is hyperfinite if it contains only hyperfinite graphs
  - E.g. planarity

Hyperfinite graphs and properties

Newman-Sohler (2013) proved that every (undirected) graph property of a hyperfinite graph is testable with constant complexity. Also: every hyperfinite property is testable with constant query complexity.

We can extend this to digraphs (in bidirectional model)

This extends the claims to one-directional model, giving two-sided error testers with query complexity $n^{1-\Omega_{\varepsilon,d}(1)}$

Conclusions

While testing of undirected graphs is rather well understood, we know little about directed graphs

In this talk: progress towards our understanding of testing digraph properties in one-directional model