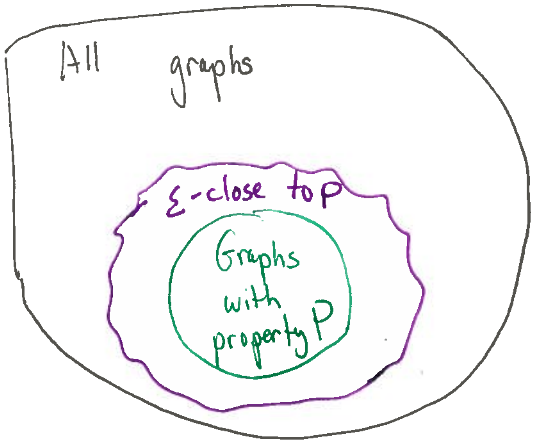
## Today

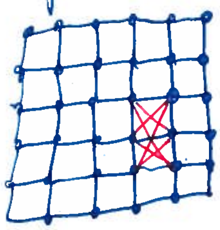
- Property testing
- testing Planarity (or any minor-free property)
- using partition oracles (also useful for other applications)

# Property Testing




Can we distinguish? in sublinear time?

e.g. $P$ = "planar"



**Compromise**

Can we distinguish graphs with prop $P$ from far from $P$?

e.g. $G$ is $\varepsilon$-far from planar if must remove $\geq \varepsilon \cdot d_{max} \cdot n$ edges to make it planar

Today: Test planarity in time independent of $n$ (but exponential in $d_{max}$, $\varepsilon$)

# Testing H-minor freeness

all graphs have max degree $\leq d$

def. • $H$ is "minor" of $G$
if can obtain $H$ from $G$ via
vertex removals, edge removals, edge contractions

• $G$ is "$H$-minor-free" if $H$ not minor of $G$

• $G$ is "$\varepsilon$-close to $H$-minor-free" if
can remove $\leq \varepsilon d n$ edges to make it
$H$-minor-free

• minor closed property $P$ –
if $G \in P$ then all minors of $G$ are in $P$

**Really Cool Theorem** [Robertson + Seymour]
Every minor-closed property is expressible
as a constant # of excluded minors.

Some minor-closed properties: $K_{3,3}$ or $K_5$ no
planar graph, bounded tree width, ...

**Goal**: Testing H-minor freeness
Pass H-minor free graphs
Fail if far from H-minor free

## more definitions

- $G$ is "$(\varepsilon, k)$-hyperfinite" if
  can remove $\leq \varepsilon n$ edges
  + remain with connected components of size $\leq k$

- $G$ is "$\rho$-hyperfinite" if
  $\forall\ \varepsilon > 0$, $G$ is $(\varepsilon, \rho(\varepsilon))$-hyperfinite
  $\Big\}$ gives # conn comp in terms of $\varepsilon$

## Useful Thm

Given $H$
$\exists\ C_H$ st. $\forall\ 0 < \varepsilon < 1$, every $H$-minor free graph of deg $\leq d$
is $(\varepsilon d, C_H^2/\varepsilon^2)$-hyperfinite.

constant that depends only on $H$

(ie. remove $\leq \varepsilon d n$ edges + components of size $O(1/\varepsilon^2)$)

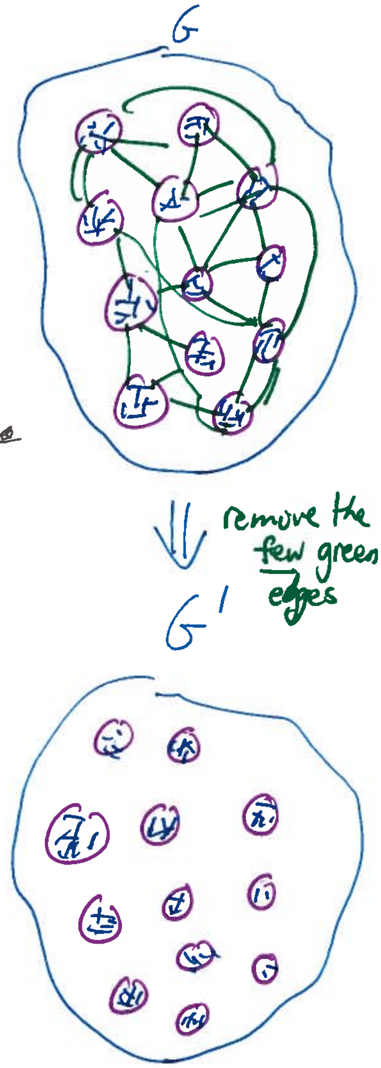
**note**

Subgraphs of $H$-minor free graphs also $H$-minor free
+ so <u>also</u> hyperfinite
but, only remove #edges in proportion to #node in subgraph

## Why is hyperfiniteness useful?

Partition graph $G$ into $G'$

how in sublinear time?
- only const size connected components remain
- removed only few edges ($\le \varepsilon d n$)
- if can't do this, $G$ is <u>not</u> $H$-minor free




If $G'$ is close to having property, so is $G$

Constant time: so test $G'$ by picking random components & seeing if they have the property

Assume we have "partition oracle" $P$
(with parameters $\frac{\varepsilon d}{4}$, $k$)

fraction edges removed ↗ ↖ component size

<u>input</u>: vertex $v$
<u>output</u>: $P[v]$ ($v$'s partition name)

s.t. $\forall v \in V$
(1) $|P[v]| \le k$
(2) $P(v)$ connected

& if $G$ is $H$-minor free
with prob $\ge \frac{9}{10}$ $\quad |\{(u,v) \in E \mid P(u) \ne P(v)\}| \le \frac{\varepsilon d n}{4}$

## Algorithm given partition oracle $P$:

- estimate number $\hat{f}$ of edges $(u,v)$ s.t. $P[u] \neq P[v]$ to additive error $\leq \frac{\varepsilon dn}{8}$ with prob failure $\leq \frac{1}{10}$
- if $\hat{f} \geq \frac{3}{8}\varepsilon dn$, output "fail" + halt
- else choose $S' = O(1/\varepsilon)$ random nodes
  if for any $s \in S'$ $P[s]$ not $H$-minor free, reject & halt
- Accept

**Analysis** (assume $P$ always correct)

if $G$ $H$-minor free:

$E[\hat{f}] \leq \frac{\varepsilon dn}{4}$

Sampling bounds $\Rightarrow$ $\hat{f} \leq \frac{\varepsilon dn}{4} + \frac{\varepsilon dn}{8} = \frac{3}{8}\varepsilon dn$

with prob $\geq 9/10$

$\forall s \in V$, $P[s]$ is $H$-minor free

if $G$ $\varepsilon$-far from $H$-minor free:

Case 1 $P$'s output doesn't satisfy $|\{(u,v) \in E : P(u) \neq P(v)\}| \leq \frac{\varepsilon dn}{2}$

Sampling bnds $\Rightarrow$ $\hat{f} \geq \frac{\varepsilon dn}{2} - \frac{\varepsilon dn}{8} \geq \frac{3}{8}\varepsilon dn$

$\Rightarrow$ output "fail" with $\underbrace{\text{prob} \geq 9/10}_{\text{make mistake only if additive estimate is off by} \geq \frac{\varepsilon dn}{8}}$

Case 2   P's output satisfies $|\{(u,v)\in E : P(u)\neq P(v)\}| \leq \frac{\epsilon d n}{2}$

$G' \leftarrow G$ with "cross" edges removed

$(u,v)$ s.t. $P[u]\neq P[v]$

if $G'$ is $\frac{\epsilon}{2}$-far from having property, third step likely to fail ✱ why?

else, $G'$ is $\frac{\epsilon}{2}$-close to property & $G$ is $\frac{\epsilon}{2}$-close to $G'$, so $G$ is $\epsilon$-close to having property ∎

✱ if $G'$ is $\frac{\epsilon}{2}$-far, need to remove at least $\epsilon d n$ edges. $\epsilon d n$ edges touch at least $\epsilon n$ nodes. Therefore, with prob $\geq \epsilon$, will pick a node in a component which is not minor-free.

So the main remaining issue is

how do we implement the partitioning oracle?

Before implementing, let's consider the following (not sublinear time, just a mental thought process)

## Global Partitioning Algorithm

let $\pi_1 \cdots \pi_n$ be a random labelling of nodes s.t. $\pi_i \neq \pi_j$ $\forall \pi_i \in [n]$
(random permutation of nodes)

$P \leftarrow \phi$

For $i = 1..n$ do

  if $\pi_i$ still in graph then

    if $\exists$ $(k,\delta)$-isolated neighborhood of $\pi_i$ in remaining graph

    then $S \leftarrow$ this nbhd

    else $S \leftarrow \{\pi_i\}$

    $P \leftarrow P \cup \{S\}$

    Remove $S$ from graph

def. of $(k,\delta)$-isol. nbhd of $\pi_i$:
1) $\pi_i \in S$
2) $S$ connected
3) $|S| \leq k$
4) $e_G(s) \leq \delta |s|$

# edges in $S$

Note order of parameters got switched from here onwards

S is just one node, hopefully this doesn't happen often!

For hyperfinite graphs, most nodes have $(k,\delta)$-isolated nbhds:

**Lemma** if $G'$ subgraph of $G$ (h.f.) with $\geq \delta n$ nodes
$\leq \frac{\varepsilon}{30}|V_{G'}|$ nodes don't have $\left(\rho\left(\frac{\varepsilon^2\delta}{1800}\right), \frac{\varepsilon}{30}\right)$-isolated nbhd.

Pf idea
- $G' \subseteq G$ h.f. so $G'$ can be broken in into components of size $\leq \rho\left(\frac{\varepsilon^3\delta}{1800}\right)$ by removing few edges

Using lemma $\Rightarrow$ $\leq \frac{\varepsilon d n}{8}$ edges cross with prob $\geq \frac{9}{10}$

## local simulation of oracle:

- assign random number $\in [0,1)$ to $v$ when first see it, use rank orders to define $\pi$
- to compute $P[v]$
  - recursively compute $P[w]$ $\forall$ $w$ of rank $< v$ within distance $\leq k$ of $v$
  - if $\exists w$ s.t. $v \in P[w]$ then $P[v] = P[w]$
    
    else look for $(k, \delta)$-isolated nbhd of $v$ which is in (ignoring any node with smaller rank) $P[w]$ for any $w$ with
    
    if find it, $P[v] \leftarrow$ this nbhd
    
    else $P[v] \leftarrow \{v\}$

## Query complexity:

Can do much much better!!

- $d^k$ nodes w/in distance $k$
- $2^{d^{O(k)}}$ using [NO] analysis

  & $k \approx \rho(\varepsilon^3/\text{big constant})$