STACS 2007, RWTH Aachen University, Germany

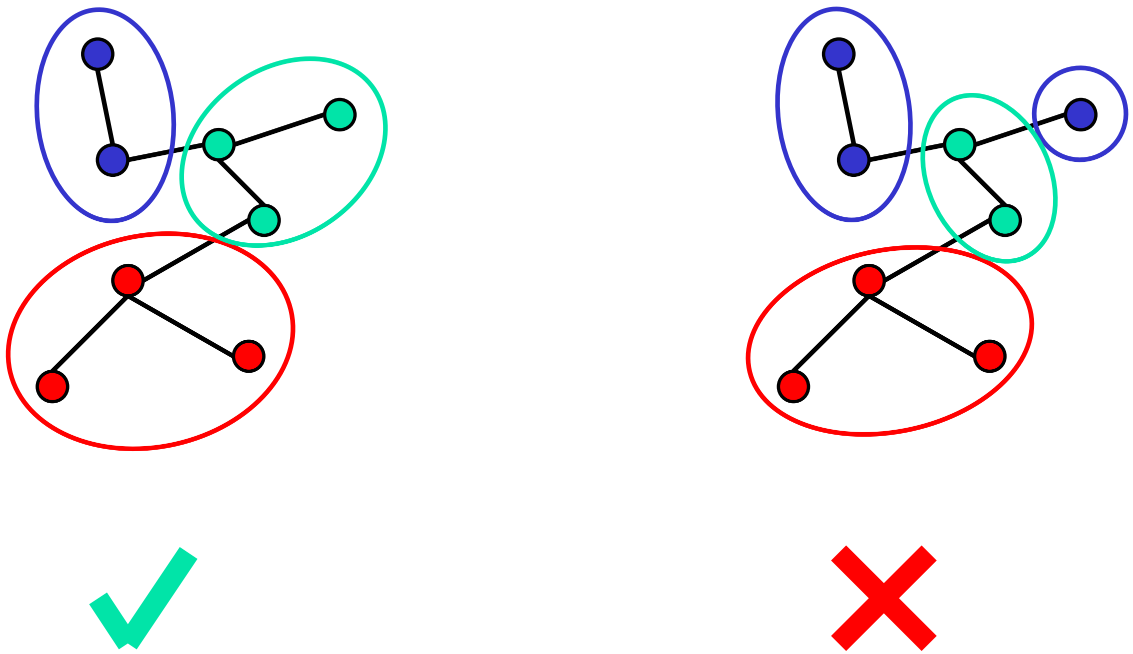
Testing Convexity Properties of Tree Colorings

Eldar Fischer and Orly Yahalom

Technion – IIT, Haifa, Israel

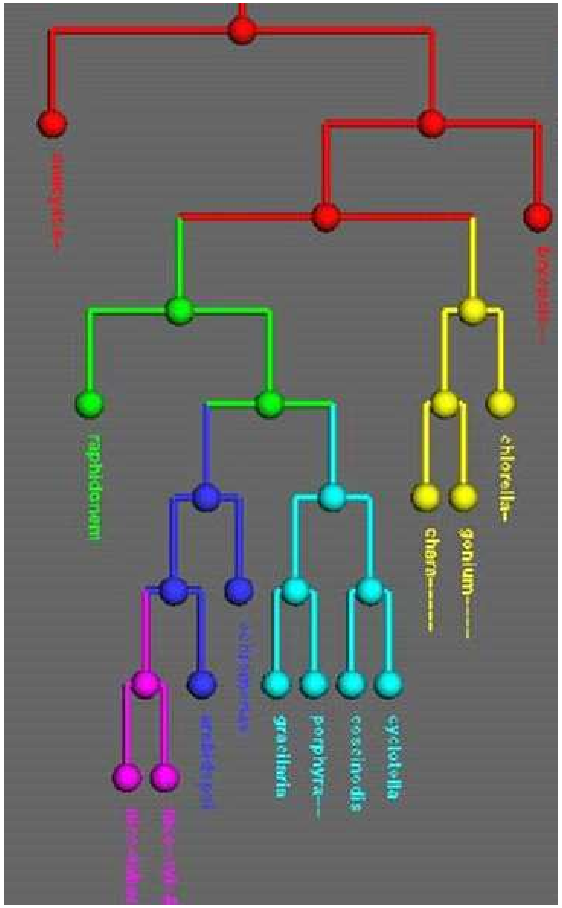
Convex colorings

- A tree coloring is convex if it induces connected color components.



Application: Phylogenetic trees

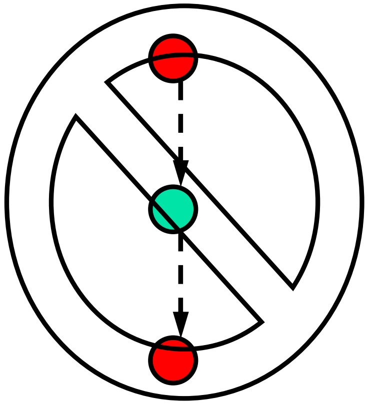
- A Phylogenetic tree describes the genetic relations between species.
- Colorings represent characters.

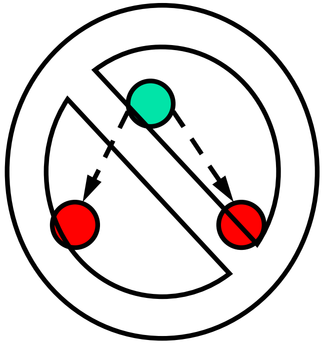


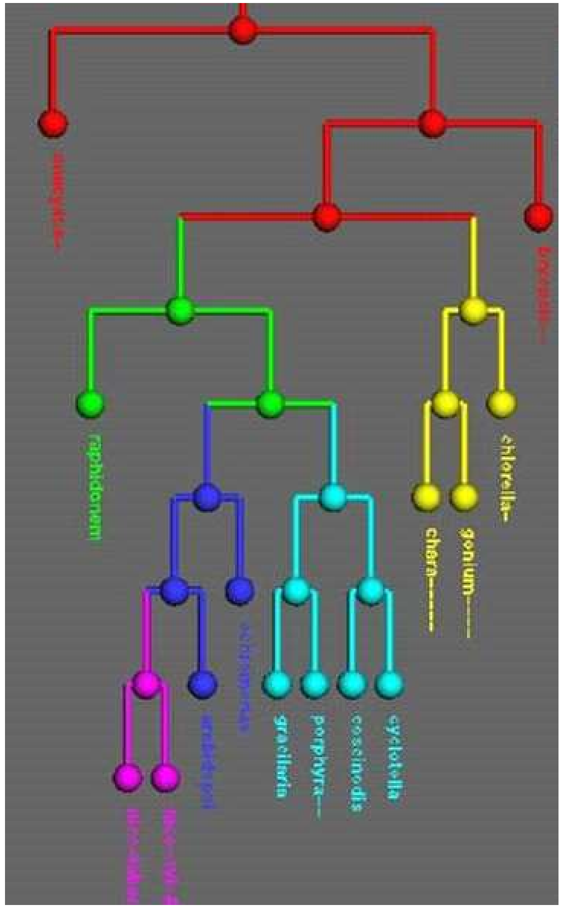
Phylogenetic Tree Viewer, Indiana University
http://www.avl.iu.edu/projects/DNAml/v2/gallery/ladder1.jpg

Application: Phylogenetic trees

- A Phylogenetic tree describes the genetic relations between species.
- Colorings represent characters.
- Normally, these colorings are convex.







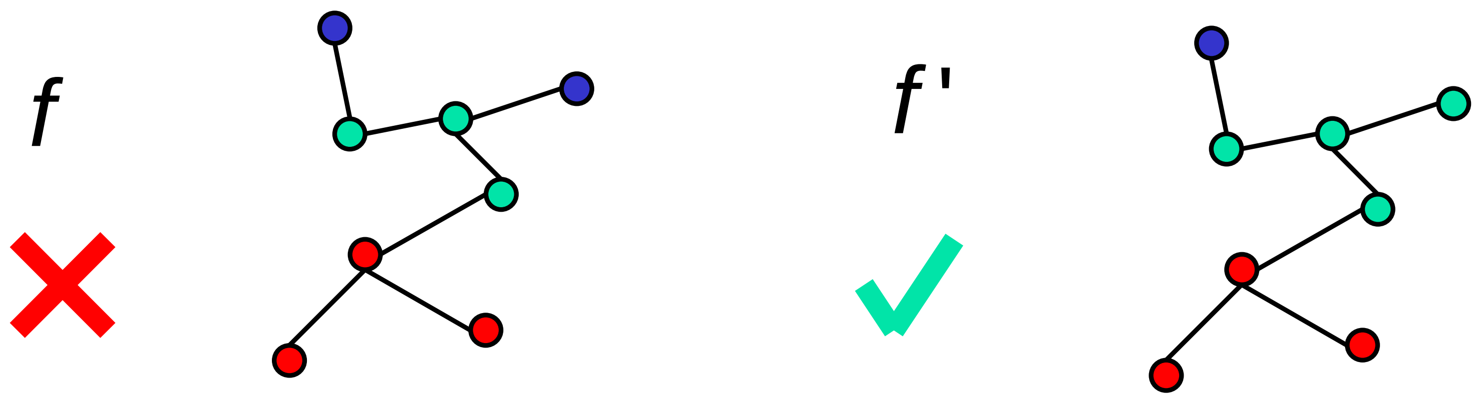
Phylogenetic Tree Viewer, Indiana University
http://www.avl.iu.edu/projects/DNAml/v2/gallery/ladder1.jpg

Property testing – General setting

- Given a domain D of inputs, a distance function $d : D^2 \to [0,1]$ and $\varepsilon > 0$, two inputs $f, f' \in D$ are called ε-close if $d(f, f') \leq \varepsilon$.

Property testing – General setting

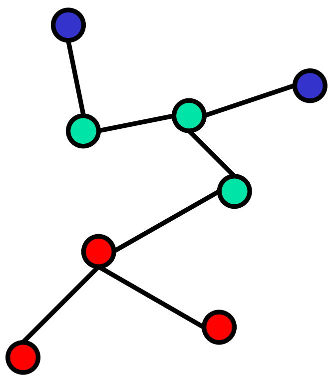
- Given a domain D of inputs, a distance function $d: D^2 \to [0,1]$ and $\varepsilon > 0$, two inputs $f, f' \in D$ are called ε-close if $d(f, f') \le \varepsilon$.



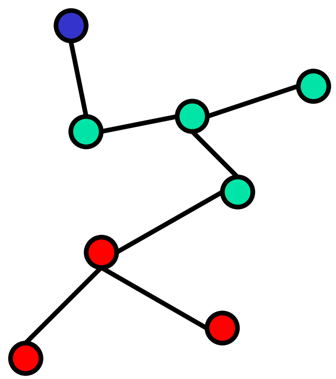
Property testing – General setting

- Given a domain D of inputs, a distance function $d : D^2 \rightarrow [0,1]$ and $\varepsilon > 0$, two inputs $f, f' \in D$ are called ε**-close** if $d(f, f') \leq \varepsilon$.
- Given a property $P \subseteq D$ and $\varepsilon > 0$, an input $f \in D$ is ε**-close** to P if there exists $f' \in P$ which is ε-close to f. Otherwise, f is ε-far from P.

f



f'



Property testers

An ε-test T for a property P and $\varepsilon > 0$ is an algorithm such that for every input $f \in D$:

- If $f \in P$ then T *accepts* f with probability at least 2/3.
- If f is ε-far from P, then T *rejects* f with probability at least 2/3.

Property testing in practice

- Property testing vs. classical decision problems:
  - Relaxed requirements
  - Reading only part of the input

Property testing in practice

- Property testing vs. classical decision problems:
  - Relaxed requirements
  - Reading only part of the input
- Property testing is useful for large data sets.
- Property testers are randomized algorithms which query the input in some locations.
- Normally, the query complexity is considered more crucial than the computational complexity.

Property testers (cont.)

- A test is called 1-sided if it accepts every input satisfying the property with probability 1.
- A test is called non-adaptive if the choice of which locations to query does not depend on the answers for previous queries.

Our settings – Colored trees

- The structure of the tree $T=(V,E)$ is fully known and unchangeable.
- The *k*-coloring $c:V\rightarrow\{1,\ldots,k\}$ is unknown.
- A query of a vertex *u* returns *c(u).*

Distance in our model

- We are given a weight function $\mu : V \to (0,1)$ such that $\Sigma_{u \in V} \mu(u) = 1$ (distribution function).
- The distance between two colorings c_1 and c_2 is $\{\mu(u) \mid c_1(u) \neq c_2(u)\}$.
- Options for μ:
  - Given in advance
  - Distribution free testing

Our main results

- A distribution free test for convexity on trees
- A lower bound for convexity on trees
- Tests for variants of convexity

Our main results

- A distribution free test for convexity on trees
- A lower bound for convexity on trees
- Tests for variants of convexity
- The query complexity of our tests depends only on k and ε - not on n.
- All our tests are 1-sided.

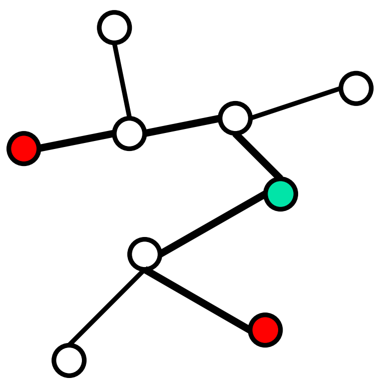
Theorem: Testing convexity on trees

For every $\varepsilon > 0$ *there exists a 1-sided, non-adaptive, distribution free* ε*-test for convexity of tree colorings with:*

- *Query complexity* $O(k/\varepsilon)$
- *Computational complexity:* $O(n)$*, or* $\tilde{O}(k/\varepsilon)$ *with a preprocessing stage of* $O(n)$.

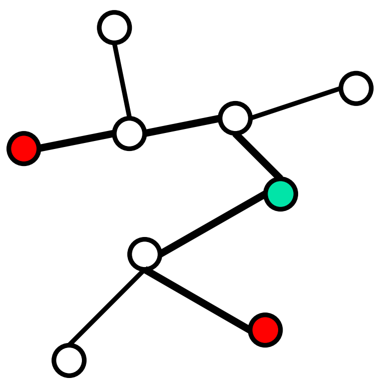
Testing convexity – main idea

- Forbidden subpath: 3 vertices of alternating colors on a path



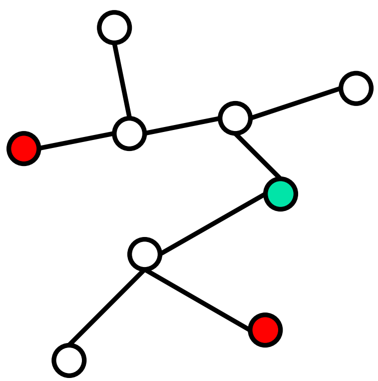
Testing convexity – main idea

- Forbidden subpath: 3 vertices of alternating colors on a path
- Convex coloring ⇔ no forbidden subpath



Testing convexity – main idea

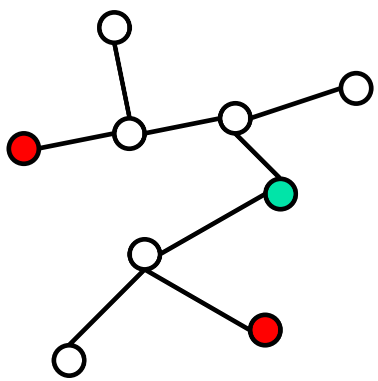
- Forbidden subpath: 3 vertices of alternating colors on a path
- Convex coloring ⇔ no forbidden subpaths
- Detecting a forbidden subpath:



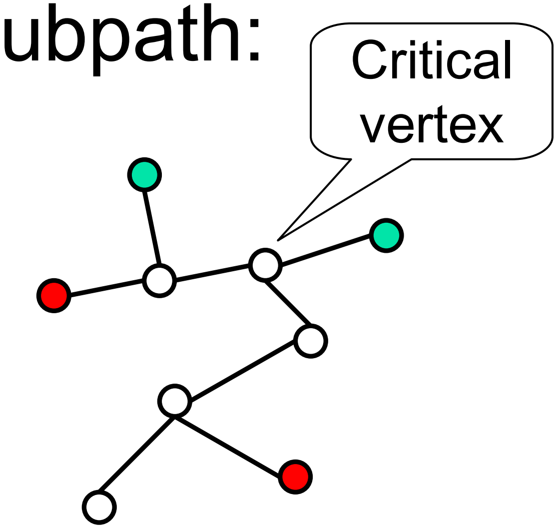
Sampling a forbidden subpath

Testing convexity – main idea

- Forbidden subpath: 3 vertices of alternating colors on a path
- Convex coloring ⇔ no forbidden subpaths
- Detecting a forbidden subpath:



Sampling a forbidden subpath



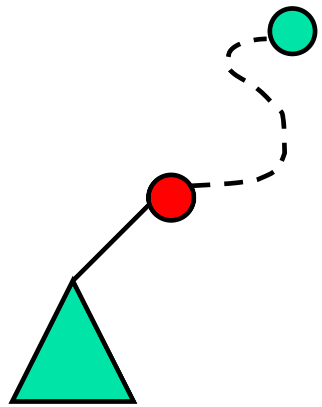
Sampling two ends of conflicting subpaths

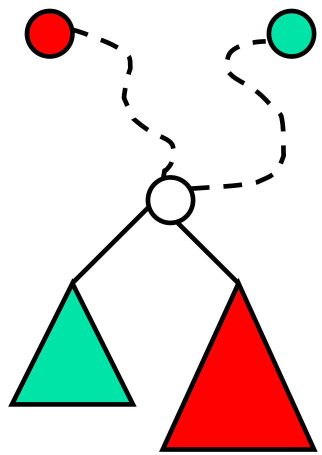
The convexity tester

- Query $O(k/\varepsilon)$ vertices uniformly and independently according to μ.
- Search the sample for forbidden subpaths, either explicit or implicit (for a critical vertex).

The convexity tester

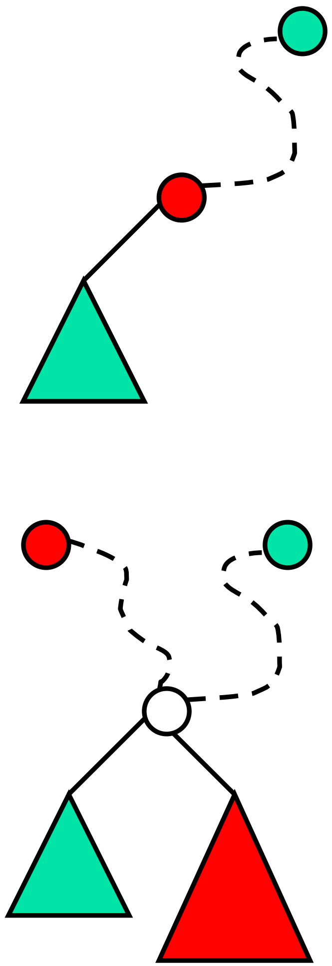
- Query $O(k/\varepsilon)$ vertices uniformly and independently according to μ.
- Search the sample for forbidden subpaths, either explicit or implicit (for a critical vertex).





The convexity tester

- Query $O(k/\varepsilon)$ vertices uniformly and independently according to μ.
- Search the sample for forbidden subpaths, either explicit or implicit (for a critical vertex).
- Reject iff a forbidden subpath was detected or inferred.

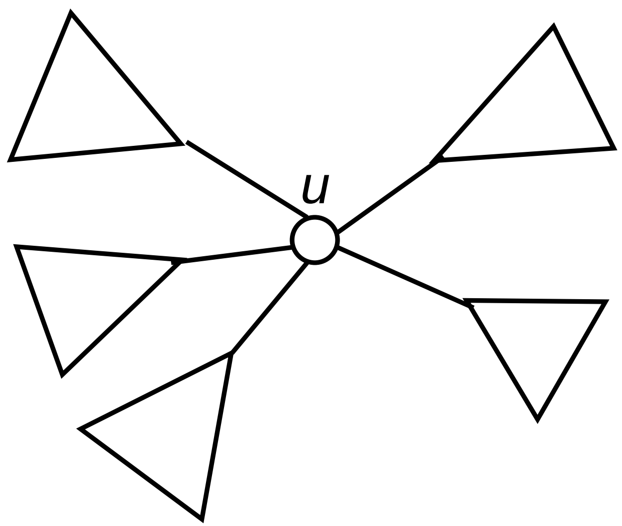


Testing convexity - correctness

- A convex coloring does not include a forbidden subpath – always accepted.
- The bulk of the proof is to show that a coloring which is ε-far from convexity is rejected with probability $\geq 2/3$.

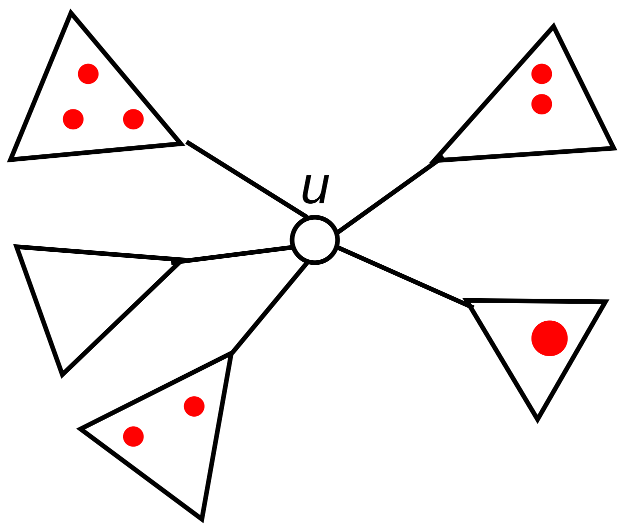
i-balanced vertices

- For a vertex *u*, ***u*-trees** are the connected components of $V \backslash \{u\}$.



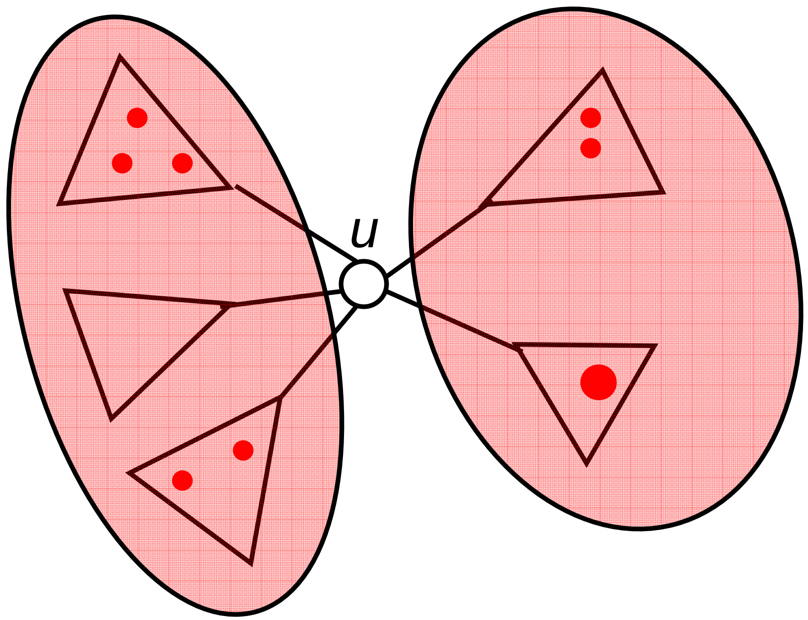
i-balanced vertices

- For a vertex *u*, *u*-trees are the connected components of $V \setminus \{u\}$.
- For a color *i*, a vertex *u* is *i*-balanced if the *u*-trees can be partitioned into two subsets, where each subset has *i*-vertices of weight ≥ $\varepsilon/8k$.



i-balanced vertices

- For a vertex *u*, ***u*-trees** are the connected components of $V \setminus \{u\}$.
- For a color *i*, a vertex *u* is ***i*-balanced** if the *u*-trees can be partitioned into two subsets, where each subset has *i*-vertices of weight ≥ $\varepsilon/8k$.



More definitions

- A color i is abundant if the total weight of i-vertices is at least $\varepsilon / 2k$.
- A vertex u is heavy if $\mu(u) \geq \varepsilon / 8k$.

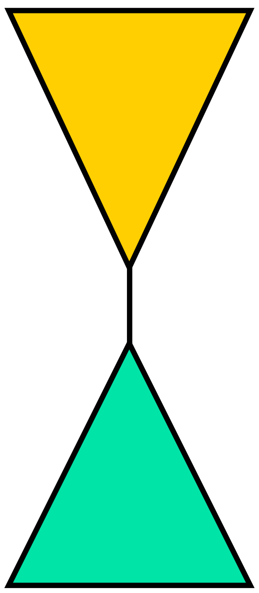
The sets B_i

For every abundant color *i*,
let $\boldsymbol{B_i}$ be the set of heavy *i*-vertices
and *i*-balanced vertices.

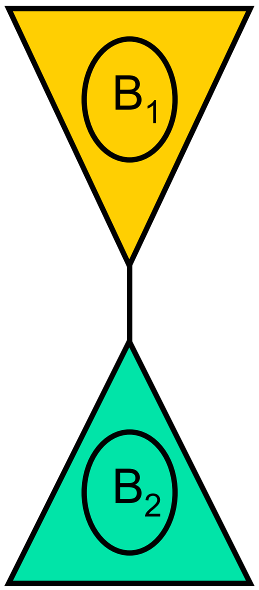
Lemma:

All the sets B_i are non-empty and
connected.

Examples

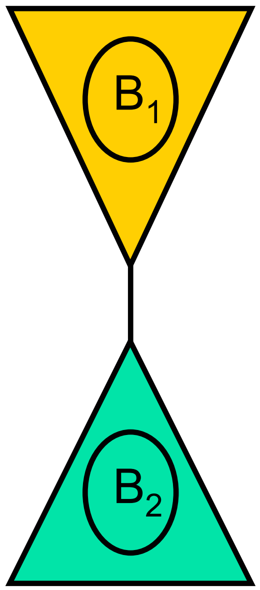


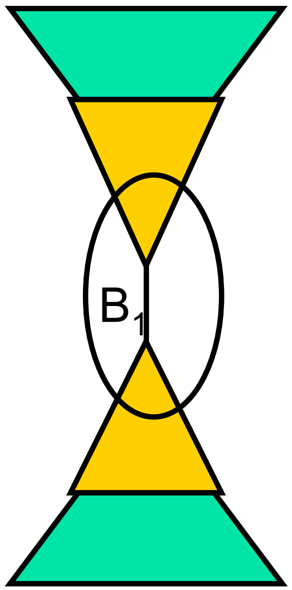
Examples



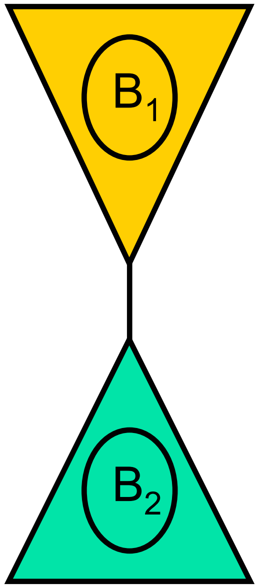
Examples

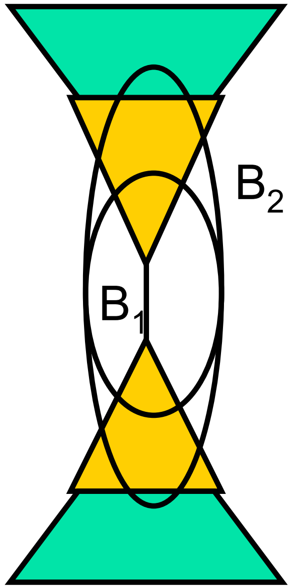
Examples





Examples



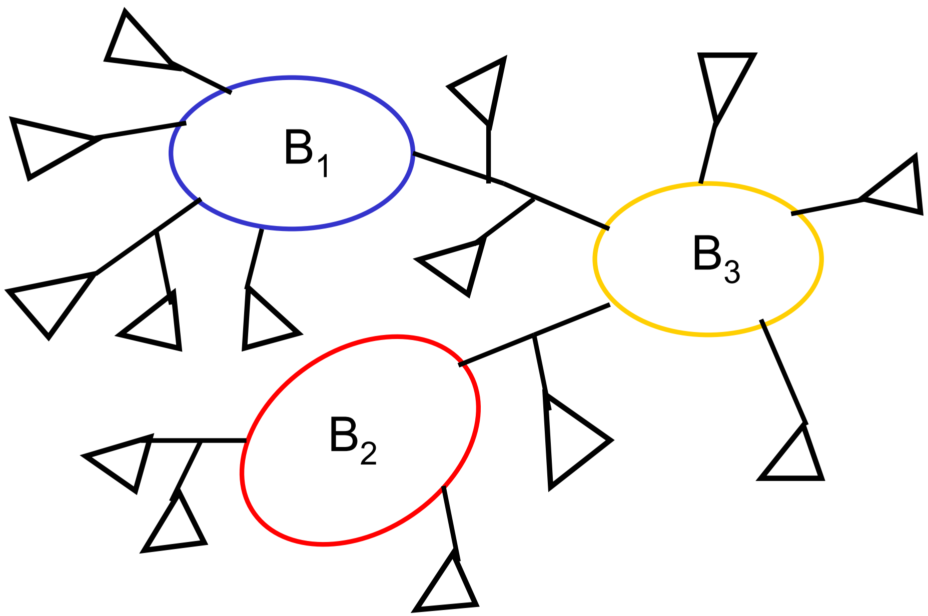


Proposition:

If a coloring c is ε-far from being convex then the B_i's are not disjoint.

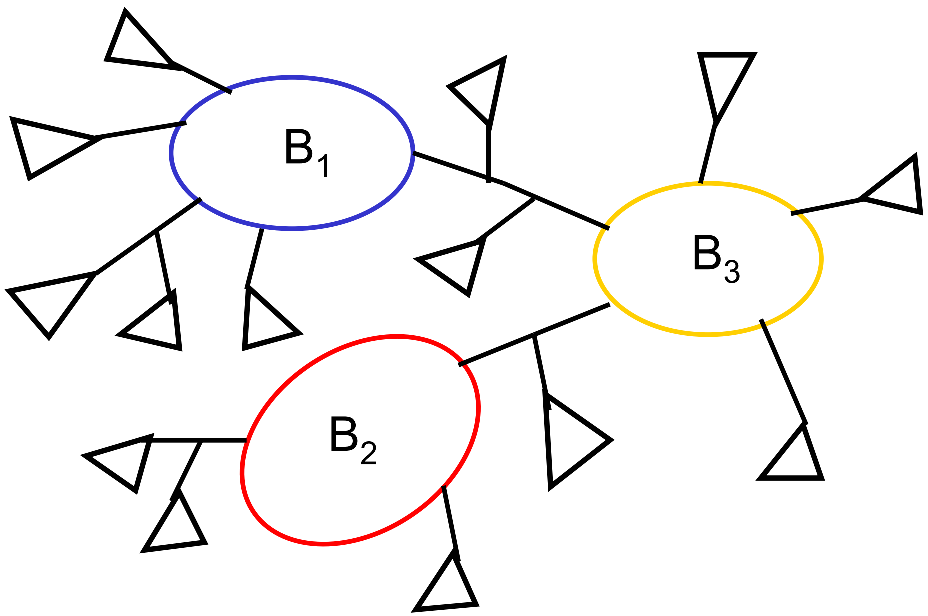
Proof of the Proposition - sketch

- Assume that the B_i's (at least 2) are disjoint.



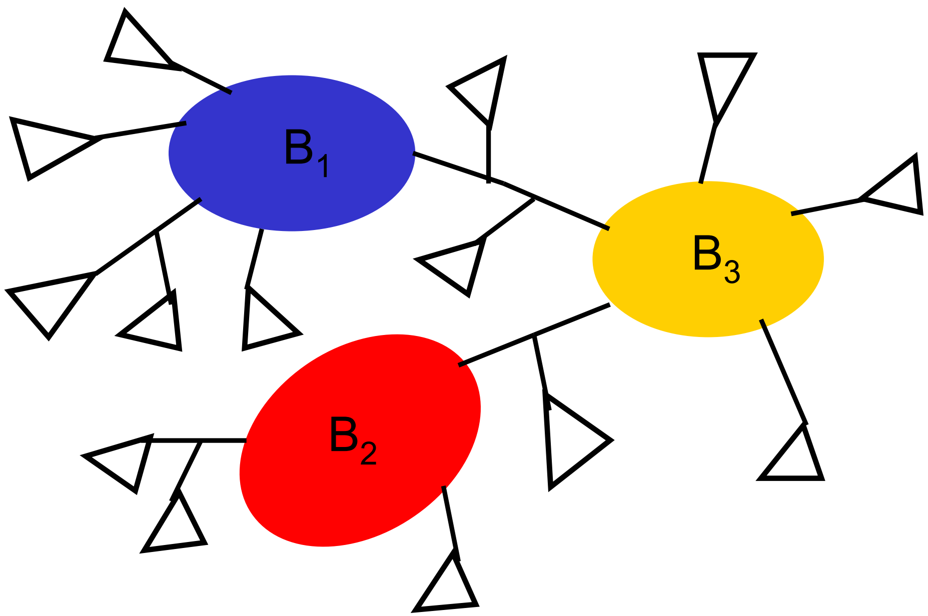
Proof of the Proposition - sketch

- Assume that the B_i's (at least 2) are disjoint.
- Define a coloring c':
  $c'(u) = i$ where B_i is the closest to u
  (choosing the minimal i in case of a tie).



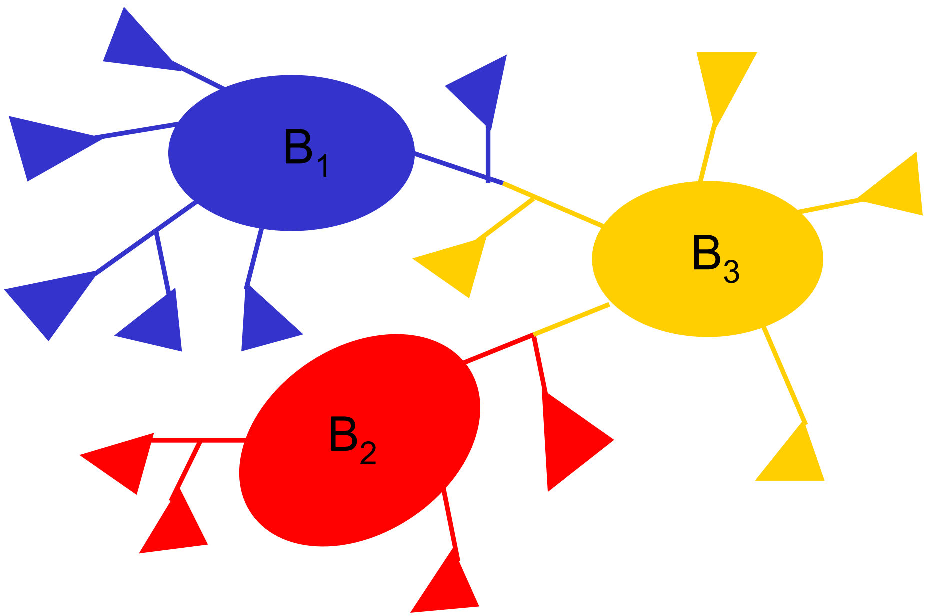
Proof of the Proposition - sketch

- Assume that the B_i's (at least 2) are disjoint.
- Define a coloring c':
  $c'(u) = i$ where B_i is the closest to u
  (choosing the minimal i in case of a tie).



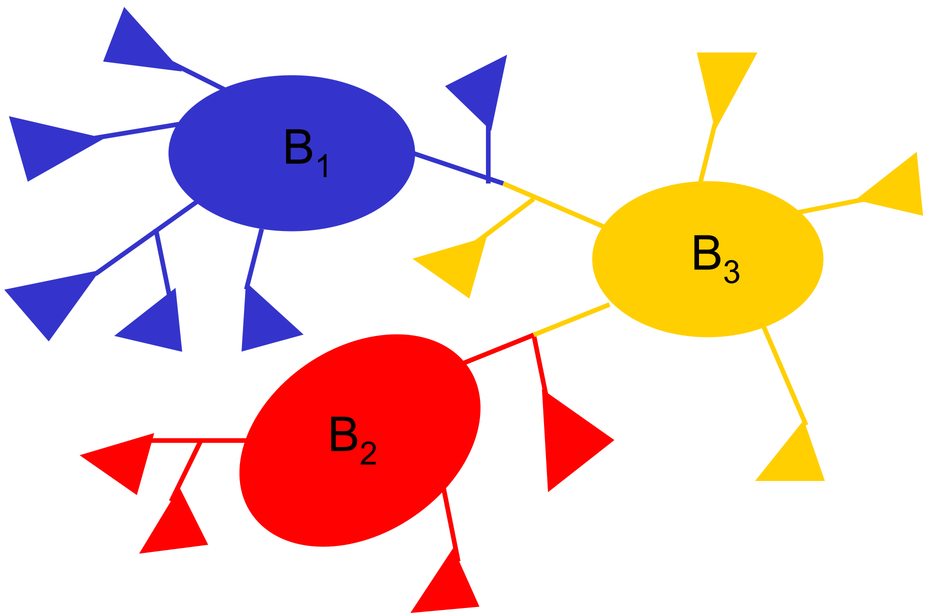
Proof of the Proposition - sketch

- Assume that the B_i's (at least 2) are disjoint.
- Define a coloring *c'*:
  $c'(u) = i$ where B_i is the closest to *u*
  (choosing the minimal *i* in case of a tie).



Proof of the Proposition - sketch

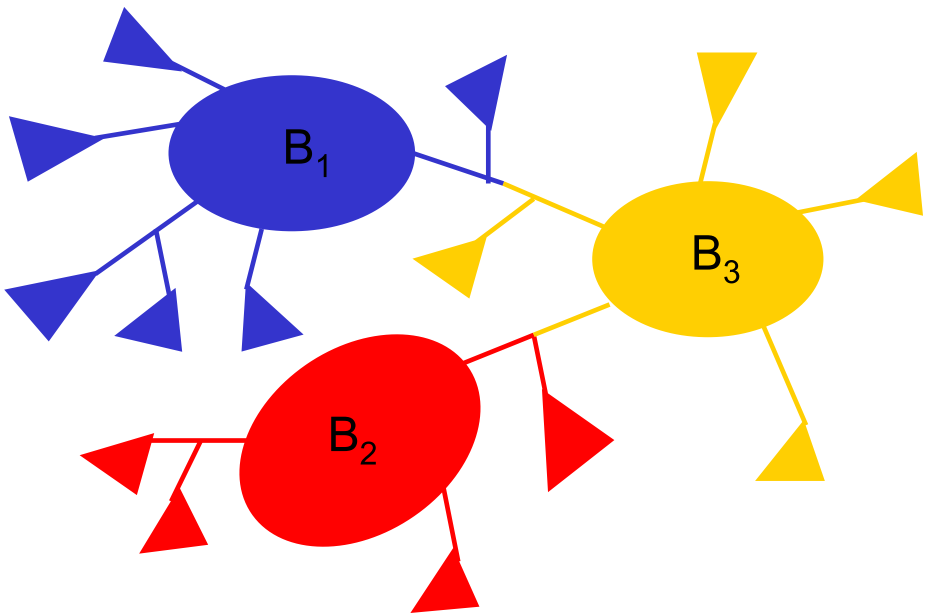
- Assume that the B_i's (at least 2) are disjoint.
- Define a coloring c':
  $c'(u) = i$ where B_i is the closest to u
  (choosing the minimal i in case of a tie).



- c' is convex.
- c' is ε-close to c.

Proof of the Proposition - sketch

- Assume that the B_i's (at least 2) are disjoint.
- Define a coloring *c'*:
  $c'(u) = i$ where B_i is the closest to *u*
  (choosing the minimal *i* in case of a tie).



- *c'* is convex.
- *c'* is ε-close to *c*.
- => c is ε-close to being convex.

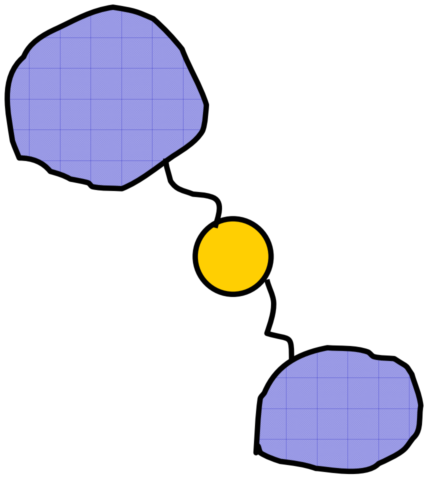
Proof of the theorem

- Assuming that a coloring *c* is ε-far from being convex, we need to show that *c* is rejected with probability ≥ 2/3.
- By the proposition, there exist two colors $i \neq j$ and a vertex u such that $u \in B_i \cap B_j$.

Proof of the theorem

- Assuming that a coloring c is ε-far from being convex, we need to show that c is rejected with probability $\geq 2/3$.
- By the proposition, there exist two colors $i \neq j$ and a vertex u such that $u \in B_i \cap B_j$.
- Case 1: u is a heavy i-vertex, and j-balanced
- Case 2: u is both i-balanced and j-balanced

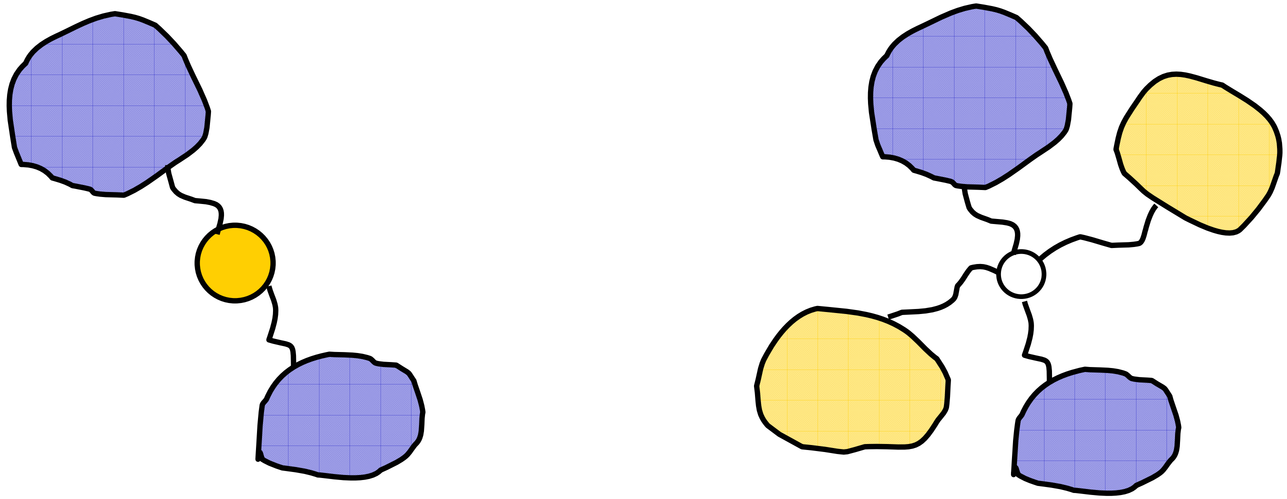
Proof of the theorem (cont.)



Case 1

- To reject *c*, it suffices to sample one vertex of each balance class (or a heavy vertex).
- This happens with probability $\geq 2/3$.

Proof of the theorem (cont.)



Case 1 Case 2

- To reject *c*, it suffices to sample one vertex of each balance class (or a heavy vertex).
- This happens with probability ≥ 2/3.

Summary

Shown here:

- A distribution free test for convexity on trees with query complexity of $O(k/\varepsilon)$

Summary

Shown here:

- A distribution free test for convexity on trees with query complexity of $O(k/\varepsilon)$

Not shown here:

- A lower bound of $\Omega\left(\sqrt{k/\varepsilon}\right)$ for the query complexity (for unweighted paths)

Summary

Shown here:

- A distribution free test for convexity on trees with query complexity of $O(k/\varepsilon)$

Not shown here:

- A lower bound of $\Omega\left(\sqrt{k/\varepsilon}\right)$ for the query complexity (for unweighted paths)
- Tests for variants of convexity:
  - Quasi-convexity
  - Relaxed convexity properties