CS 207: Discrete Structures

Graph theory

Graph automorphisms, Connected components, cut edges.

Lecture 27

Sept 24 2015

Topic 3: Graph theory

Recap of last four lectures:

1. Basics: graphs, paths, cycles, walks, trails; connected graphs.
2. Eulerian graphs and a characterization in terms of degrees of vertices.
3. Bipartite graphs and a characterization in terms of odd length cycles.
4. Cliques and independent sets.

Topic 3: Graph theory

Recap of last four lectures:

1. Basics: graphs, paths, cycles, walks, trails; connected graphs.
2. Eulerian graphs and a characterization in terms of degrees of vertices.
3. Bipartite graphs and a characterization in terms of odd length cycles.
4. Cliques and independent sets.
5. A proof and algo for large bipartite subgraphs of a graph.
6. Graph representation (as matrices) and isomorphism

Reference: Sections 1.1-1.3 of Chapter 1 from Douglas West.

This lecture

- How to check if two graphs are isomorphic/non-isomorphic?
- Graph automorphisms
- Relations between vertices

Graph isomorphism

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

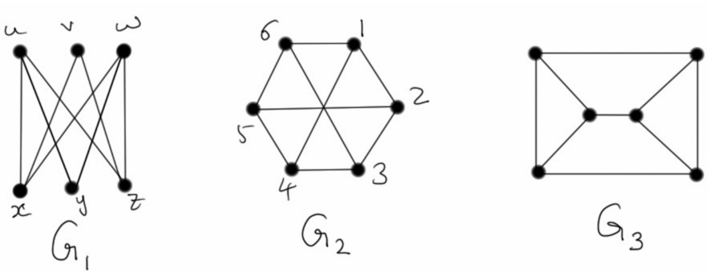
Graph isomorphism

Definition

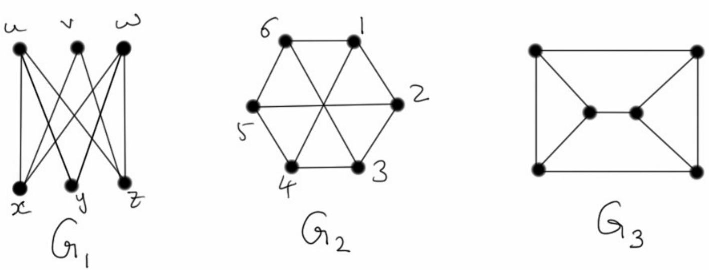
An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

- Thus, it is a bijection that “preserves” the edge relation.
- Can be checked using adjacency matrix by reordering/renaming.
- The isomorphism relation is an equivalence relation on simple graphs.
- By an “unlabeled” graph, we mean the isomorphism class of that graph.

Graph isomorphism



Graph isomorphism



- To show that two graphs are isomorphic, you have to
    1. give names to vertices
    2. specify a bijection & check that it preserves the adjacency relation
       i.e., check that adjacency matrices become identical by permuting rows and columns.
- To show that two graphs are non-isomorphic, find a structural property that is different.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

- Are C_5 and $P_5 \cup \{e\}$ isomorphic?

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

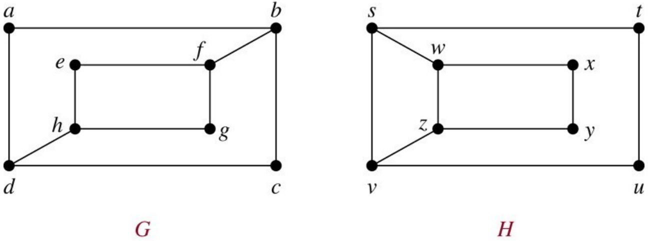
Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.
3. G, H have the same # vertices of degree k, $\forall k \in \mathbb{N}$.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.



Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.
3. G, H have the same # vertices of degree k, $\forall k \in \mathbb{N}$.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.
3. G, H have the same # vertices of degree k, $\forall k \in \mathbb{N}$.
4. G has k paths/cycles of length r iff H has k paths/cycles of length r.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.
3. G, H have the same # vertices of degree k, $\forall k \in \mathbb{N}$.
4. G has k paths/cycles of length r iff H has k paths/cycles of length r.
5. G is bipartite iff H is bipartite.
6. G contains K_n as a subgraph iff H does.
7. $\ldots$

Graph Automorphisms

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

Graph Automorphisms

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

What if $G = H$?

Graph Automorphisms

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

What if $G = H$?

Definition

An automorphism of G is an isomorphism from G to itself, i.e., a bijection $f : V(G) \to V(G)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

Graph Automorphisms

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

What if $G = H$?

Definition

An automorphism of G is an isomorphism from G to itself, i.e., a bijection $f : V(G) \to V(G)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

- What are the automorphisms of P_4?

Graph Automorphisms

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

What if $G = H$?

Definition

An automorphism of G is an isomorphism from G to itself, i.e., a bijection $f : V(G) \to V(G)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

- What are the automorphisms of P_4?
- How many automorphisms does K_n have?

Graph Automorphisms

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

What if $G = H$?

Definition

An automorphism of G is an isomorphism from G to itself, i.e., a bijection $f : V(G) \to V(G)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

- What are the automorphisms of P_4?
- How many automorphisms does K_n have?
- How many automorphisms does $K_{r,s}$ have?

Graph Automorphisms

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

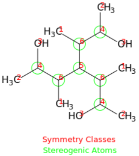
What if $G = H$?

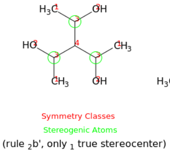
Definition

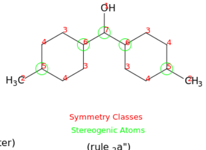
An automorphism of G is an isomorphism from G to itself, i.e., a bijection $f : V(G) \to V(G)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

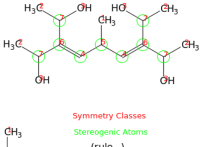
Automorphisms are a measure of symmetry.

Practical applications in graph drawing, visualization, molecular symmetry, structured boolean satisfiability, formal verification









Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

1. Adjacency: uRv iff there is an edge between u and v. Any nice properties?

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

1. Adjacency: uRv iff there is an edge between u and v. Any nice properties?
2. Connectedness: uPv iff there is a path between u and v.

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

1. Adjacency: uRv iff there is an edge between u and v. Any nice properties?
2. Connectedness: uPv iff there is a path between u and v.

   P is an equivalence relation.

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

1. Adjacency: uRv iff there is an edge between u and v. Any nice properties?
2. Connectedness: uPv iff there is a path between u and v.

   P is an equivalence relation.

Definition

A maximal connected subgraph of G is a subgraph that is connected and is not contained in any other connected subgraph of G.
The components of G are its maximal connected subgraphs.

Thus, equivalence classes of P are the vertex sets of the components of G.

Difference between maximal and maximum

- Is every maximal path maximum, i.e., have maximum length?
- A maximal structure is a structure that is not contained in a larger structure, i.e., increasing the structure will violate some property.
- Maximum just means that size is the greatest among all possible.

Exercises

1. Give a path which is maximal but not maximum.
2. Give a subgraph of a graph which is maximally connected, but not maximum (i.e., does not have maximum # edges).
3. How many maximal/maximum independent sets does $K_{r,s}$ have?

Components and cut-edges

Properties of components

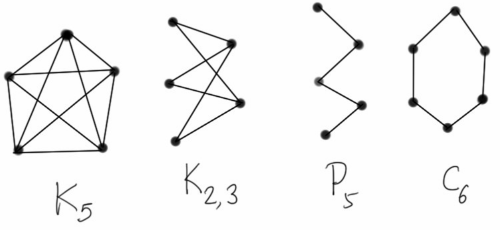
- A component with no edges is called trivial. Thus isolated verices form trivial components.
- Components are pairwise disjoint.

Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated verices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?

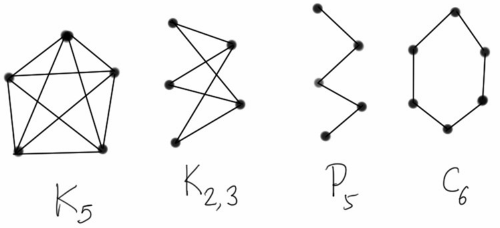


Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated verices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.



Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated verices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.

Theorem: Characterize cut-edges using cycles

Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated verices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.

Theorem: Characterize cut-edges using cycles

Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated verices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.

Theorem: Characterize cut-edges using cycles

Exercise!

Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated verices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.

Theorem: Characterize cut-edges using cycles

Exercise! An edge is a cut-edge iff it belongs to no cycle.