Big Graph Processing Systems

Part I: Graph Query Paradigms and their Semantics

▶ **Chapter 1:** Graph Pattern Matching

Christopher Spinrath

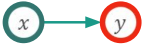
CNRS – LIRIS – Lyon 1 Université

DISS Master 2025

This presentation is an adaption of slides from Angela Bonifati

Graph Pattern Matching

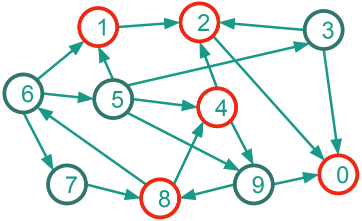
Subgraph pattern p



Graph with place holders x and y

Matching p on G

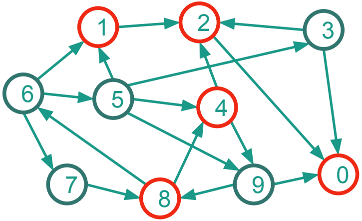
Finds all subgraphs in G that fit to p



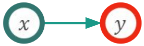
Graph Pattern Matching

Matching p on G

Finds all subgraphs in G that fit to p



Subgraph pattern p



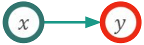
Graph with place holders x and y

Intuitively, how many matches does p have on G?

Number of Results	
A	5
B	8
C	10
D	14

Graph Pattern Matching

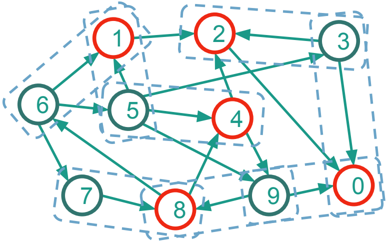
Subgraph pattern p



Graph with place holders x and y

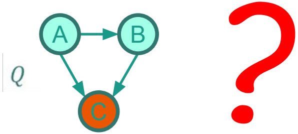
Matching p on G

Finds all subgraphs in G that fit to p



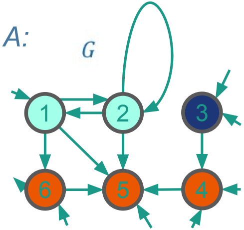
Graph Pattern Matching

Given the following query:

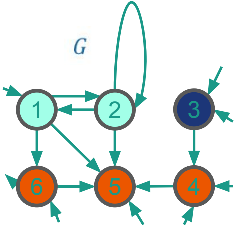


What you think is a valid match?

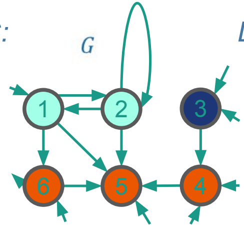
A:



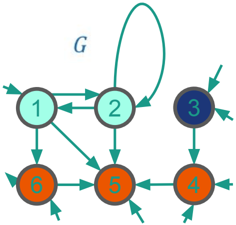
B:



C:

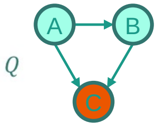


D:



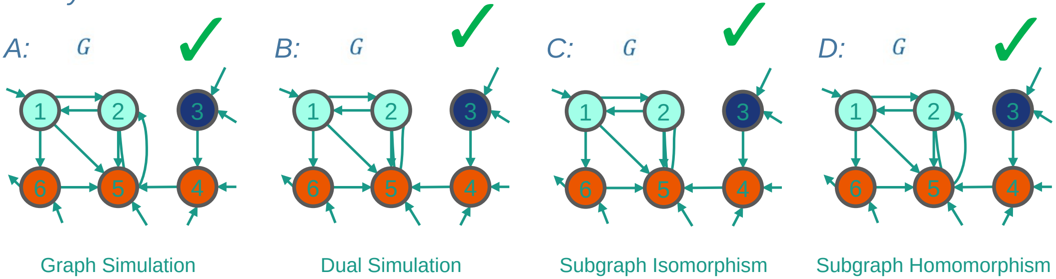
Graph Pattern Matching

Given the following query:



Well, depends on the matching semantics.

What you think is a valid match?



Matching Semantics

Plethora of Matching Semantics

[Miller et al., "Research Directions for Big Data Graph Analytics", *IEEE International Congress on Big Data, New York City, NY, USA, June 27 - July 2*, 2015]

Different Matching Semantics

Model	Complexity Class	Source	Results Contained in
Graph Simulation	Quadratic	Henzinger et al. 1995 [18]	-
Dual Simulation	Cubic	Ma et al. 2011 [19]	Graph Simulation
Strong Simulation	Cubic	Ma et al. 2011 [19]	Dual Simulation
Strict Simulation	Cubic	Fard et al., 2013 [20]	Strong Simulation
Tight Simulation	Cubic	Fard et al., 2014 [21]	Strict Simulation
CAR-Tight Simulation	Cubic	Fard et al., 2014 [22]	Tight Simulation
Graph Homeomorphism	$\mathcal{NP}$-hard	Fortune et al., 1980 [28]	-
Graph Homomorphism	$\mathcal{NP}$-hard	Hell and Nesetril, 1990 [23]	Graph Homeomorphism and Tight Simulation
Subgraph Isomorphism	$\mathcal{NP}$-hard	Garey and Johnson, 1979 [29]	Graph Homomorphism and CAR-Tight Simulation

[18] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke, "Computing simulations on finite and infinite graphs," in *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on.* IEEE, 1995, pp. 453–462.

[19] S. Ma, Y. Cao, W. Fan, J. Huai, and T. Wo, "Capturing topology in graph pattern matching," *Proceedings of the VLDB Endowment*, vol. 5, no. 4, pp. 310–321, 2011.

[20] A. Fard, M. U. Nisar, L. Ramaswamy, J. A. Miller, and M. Saltz, "A distributed vertex-centric approach for pattern matching in massive graphs," in *Big Data Conference*, Oct 2013, pp. 403–411.

[21] A. Fard, M. U. Nisar, J. A. Miller, and L. Ramaswamy, "Distributed and scalable graph pattern matching: Models and algorithms," *International Journal of Big Data (IJBD)*, vol. 1, no. 1, 2014.

[22] A. Fard, S. Manda, L. Ramaswamy, and J. A. Miller, "Effective caching techniques for accelerating pattern matching queries," in *Big Data (Big Data), 2014 IEEE International Conference on.* IEEE, 2014, pp. 491–499.

[23] P. Hell and J. Nešetřil, "On the complexity of h-coloring," *Journal of Combinatorial Theory, Series B*, vol. 48, no. 1, pp. 92–110, 1990.

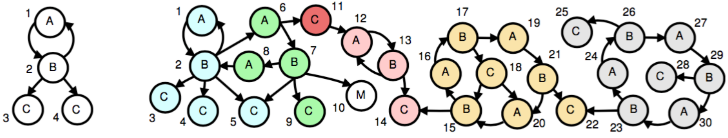
[28] S. Fortune, J. Hopcroft, and J. Wyllie, "The directed subgraph homeomorphism problem," *Theoretical Computer Science*, vol. 10, no. 2, pp. 111–121, 1980.

[29] R. G. Michael and S. J. David, "Computers and intractability: a guide to the theory of NP-completeness," *WH Freeman & Co., San Francisco*, 1979.

Plethora of Matching Semantics – Example

Example

A: Arts Book
B: Biography Book
C: Children's Book
M: Music CD



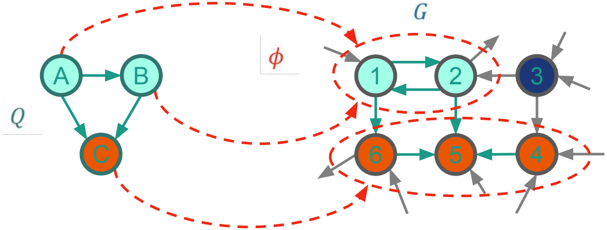
a) Q: Pattern   b) G: Data Graph

Results of Different Semantics

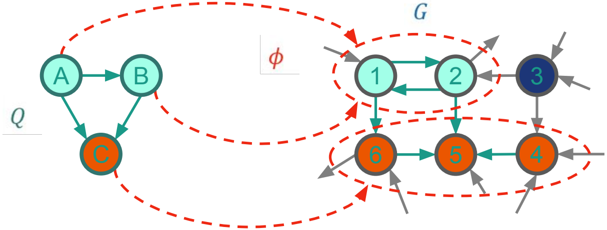
Model	Subgraph Results	Count
Graph Simulation	$\Phi(1,2,3,4) \rightarrow (\{1,6,8,12,16,19,20,24,27,30\}, \{2,7,13,15,17,21,23,26,29\}, \{3,4,5,9,11,14,18,22,25,28\}, \{3,4,5,9,11,14,18,22,25,28\})$	29
Dual Simulation	$\Phi(1,2,3,4) \rightarrow (\{1,6,8,12,16,19,20,24,27,30\}, \{2,7,13,15,17,21,23,26,29\}, \{3,4,5,9,14,18,22,25,28\}, \{3,4,5,9,14,18,22,25,28\})$	28
Strong Simulation	$\Phi(1,2,3,4) \rightarrow (\{1,6,8\}, \{2,7\}, \{3,4,5,9\}, \{3,4,5,9\}), (12,13,14,14), (\{16,19,20\}, \{15,17,21\}, \{14,18,22\}, \{14,18,22\})$	20
Strict Simulation	$\Phi(1,2,3,4) \rightarrow (\{1,6,8\}, \{2,7\}, \{3,4,5,9\}, \{3,4,5,9\}), (12,13,14,14)$	12
Tight Simulation	$\Phi(1,2,3,4) \rightarrow (1,2,\{3,4,5\},\{3,4,5\}), (12,13,14,14)$	8
CAR-Tight Simulation	$\Phi(1,2,3,4) \rightarrow (1,2,\{3,4,5\},\{3,4,5\})$	5
Graph Homomorphism	$f(1,2,3,4) \rightarrow (1,2,3,4), (1,2,3,5), (1,2,4,5), (1,2,3,3), (1,2,4,4), (1,2,5,5), (12,13,14,14)$	8
Subgraph Isomorphism	$f(1,2,3,4) \rightarrow (1,2,3,4), (1,2,3,5), (1,2,4,5)$	5

Simulation-Based Semantic

Graph Simulation



Graph Simulation



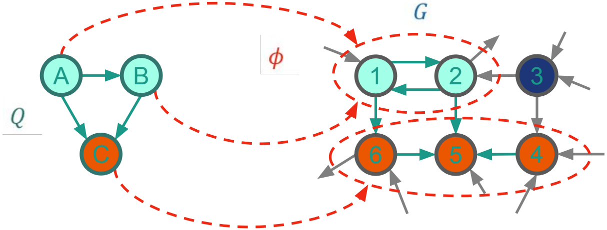
Definition

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$

Graph Simulation

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$

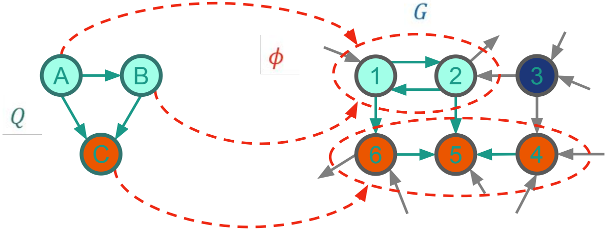


Definition

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$



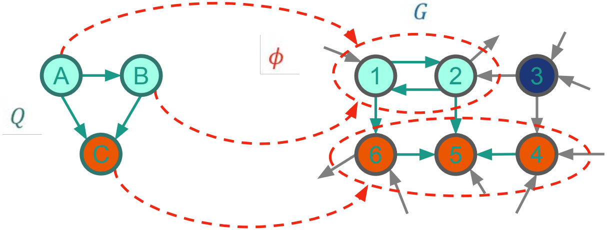
Definition

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- A graph simulation matching is a function $\phi\colon V_Q \to \mathcal{P}(V_G)$ such that for all $v_Q \in V_Q$

Graph Simulation

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$



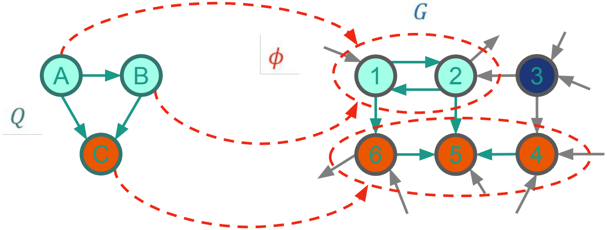
Definition

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- A graph simulation matching is a function $\phi\colon V_Q \to \mathcal{P}(V_G)$ such that for all $v_Q \in V_Q$
    1. for all $v_G \in \phi(v_Q)$ the properties of v_G and v_Q match (here: they have the same color) and

Graph Simulation

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$



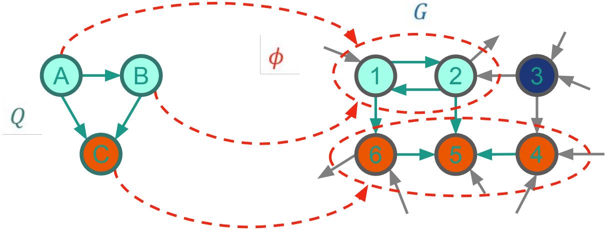
Definition

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- A graph simulation matching is a function $\phi\colon V_Q \to \mathcal{P}(V_G)$ such that for all $v_Q \in V_Q$
  1. for all $v_G \in \phi(v_Q)$ the properties of v_G and v_Q match (here: they have the same color) and
  2. for all $v_G \in \phi(v_Q)$ and $(v_Q, w_Q) \in E_Q$ there is a $w_G \in \phi(w_Q)$ with $(v_G, w_G) \in E_G$

Graph Simulation

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$

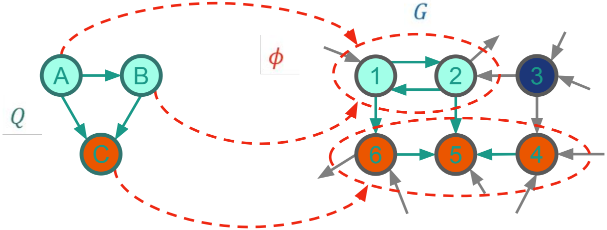


- $1 \in \phi(A)$ and $(A, B) \in E_Q$: there is $2 \in \phi(B)$ with $(1, 2) \in E_G$

Graph Simulation

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$

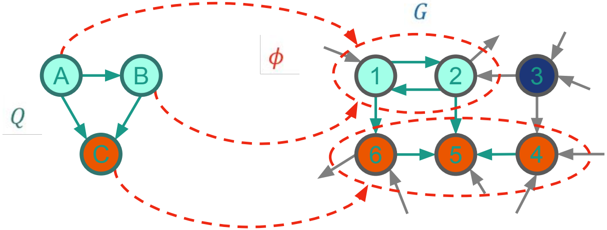


- $1 \in \phi(A)$ and $(A, B) \in E_Q$: there is $2 \in \phi(B)$ with $(1, 2) \in E_G$
- $1 \in \phi(A)$ and $(A, C) \in E_Q$: there is $6 \in \phi(C)$ with $(1, 6) \in E_G$

Graph Simulation

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$

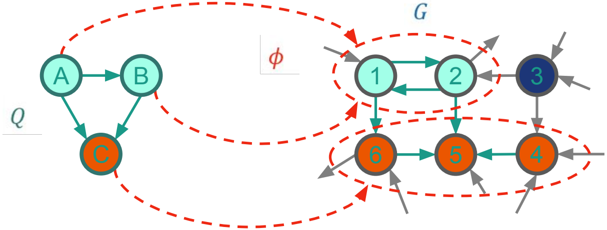


- $1 \in \phi(A)$ and $(A, B) \in E_Q$: there is $2 \in \phi(B)$ with $(1, 2) \in E_G$
- $1 \in \phi(A)$ and $(A, C) \in E_Q$: there is $6 \in \phi(C)$ with $(1, 6) \in E_G$
- $2 \in \phi(A)$ and $(A, B) \in E_Q$: there is $1 \in \phi(B)$ with $(2, 1) \in E_G$
- $2 \in \phi(A)$ and $(A, C) \in E_Q$: there is $5 \in \phi(C)$ with $(2, 5) \in E_G$

Graph Simulation

Example

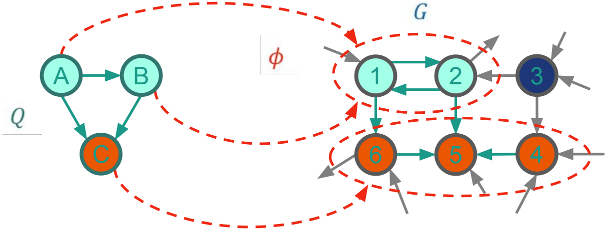
- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$



- $1 \in \phi(A)$ and $(A, B) \in E_Q$: there is $2 \in \phi(B)$ with $(1, 2) \in E_G$
- $1 \in \phi(A)$ and $(A, C) \in E_Q$: there is $6 \in \phi(C)$ with $(1, 6) \in E_G$
- $2 \in \phi(A)$ and $(A, B) \in E_Q$: there is $1 \in \phi(B)$ with $(2, 1) \in E_G$
- $2 \in \phi(A)$ and $(A, C) \in E_Q$: there is $5 \in \phi(C)$ with $(2, 5) \in E_G$
- $1 \in \phi(B)$ and $(B, C) \in E_Q$: there is $6 \in \phi(C)$ with $(1, 6) \in E_G$
- $2 \in \phi(B)$ and $(B, C) \in E_Q$: there is $5 \in \phi(C)$ with $(2, 5) \in E_G$

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5, 4\}$

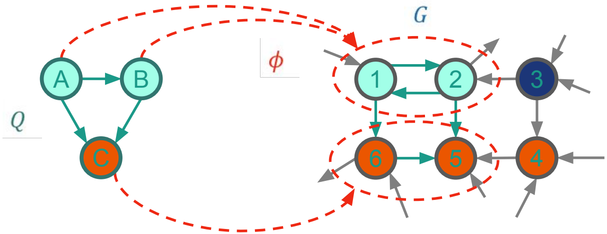


- $1 \in \phi(A)$ and $(A, B) \in E_Q$: there is $2 \in \phi(B)$ with $(1, 2) \in E_G$
- $1 \in \phi(A)$ and $(A, C) \in E_Q$: there is $6 \in \phi(C)$ with $(1, 6) \in E_G$
- $2 \in \phi(A)$ and $(A, B) \in E_Q$: there is $1 \in \phi(B)$ with $(2, 1) \in E_G$
- $2 \in \phi(A)$ and $(A, C) \in E_Q$: there is $5 \in \phi(C)$ with $(2, 5) \in E_G$
- $1 \in \phi(B)$ and $(B, C) \in E_Q$: there is $6 \in \phi(C)$ with $(1, 6) \in E_G$
- $2 \in \phi(B)$ and $(B, C) \in E_Q$: there is $5 \in \phi(C)$ with $(2, 5) \in E_G$
- C has no outgoing edges

Dual Simulation

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5\}$



Definition

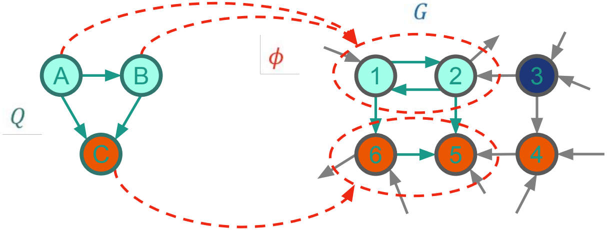
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- A graph simulation matching is a function $\phi\colon V_Q \to \mathcal{P}(V_G)$ such that for all $v_Q \in V_Q$
  1. for all $v_G \in \phi(v_Q)$ the properties of v_G and v_Q match (here: they have the same color) and
  2. for all $v_G \in \phi(v_Q)$ and $(v_Q, w_Q) \in E_Q$ there is a $w_G \in \phi(w_Q)$ with $(v_G, w_G) \in E_G$
  3. for all $v_G \in \phi(v_Q)$ and $(w_Q, v_Q) \in E_Q$ there is a $w_G \in \phi(w_Q)$ with $(w_G, v_G) \in E_G$

Dual Simulation

Example

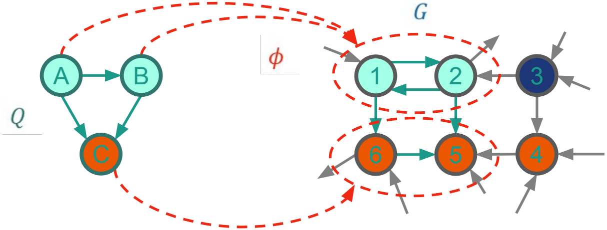
- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5\}$

- A has no incoming edges



Example

- $\phi(A) = \{1,2\}$
- $\phi(B) = \{1,2\}$
- $\phi(C) = \{6,5\}$

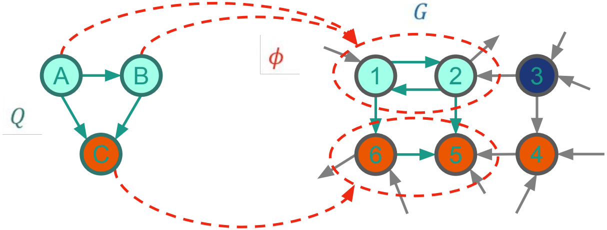


- A has no incoming edges
- $1 \in \phi(B)$ and $(A,B) \in E_Q$: there is $2 \in \phi(A)$ with $(2,1) \in E_G$

Dual Simulation

Example

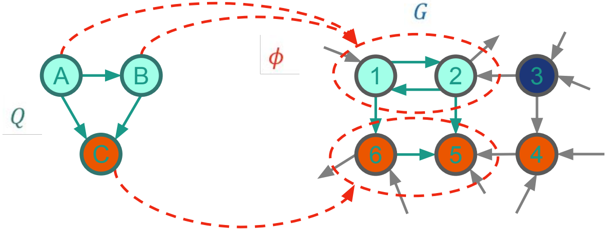
- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5\}$



- A has no incoming edges
- $1 \in \phi(B)$ and $(A, B) \in E_Q$: there is $2 \in \phi(A)$ with $(2, 1) \in E_G$
- $2 \in \phi(B)$ and $(A, B) \in E_Q$: there is $1 \in \phi(A)$ with $(1, 2) \in E_G$

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5\}$

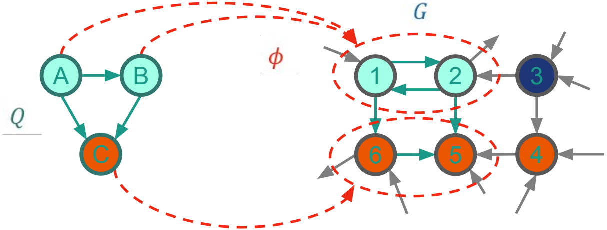


- A has no incoming edges
- $1 \in \phi(B)$ and $(A, B) \in E_Q$: there is $2 \in \phi(A)$ with $(2, 1) \in E_G$
- $2 \in \phi(B)$ and $(A, B) \in E_Q$: there is $1 \in \phi(A)$ with $(1, 2) \in E_G$
- $5 \in \phi(C)$ and $(A, C) \in E_Q$: there is $2 \in \phi(A)$ with $(2, 5) \in E_G$
- $5 \in \phi(C)$ and $(B, C) \in E_Q$: there is $2 \in \phi(B)$ with $(2, 5) \in E_G$

Dual Simulation

Example

- $\phi(A) = \{1, 2\}$
- $\phi(B) = \{1, 2\}$
- $\phi(C) = \{6, 5\}$

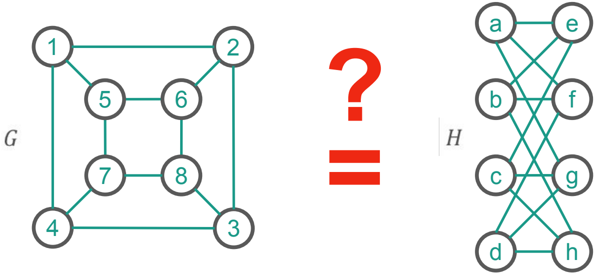


- A has no incoming edges
- $1 \in \phi(B)$ and $(A, B) \in E_Q$: there is $2 \in \phi(A)$ with $(2, 1) \in E_G$
- $2 \in \phi(B)$ and $(A, B) \in E_Q$: there is $1 \in \phi(A)$ with $(1, 2) \in E_G$
- $5 \in \phi(C)$ and $(A, C) \in E_Q$: there is $2 \in \phi(A)$ with $(2, 5) \in E_G$
- $5 \in \phi(C)$ and $(B, C) \in E_Q$: there is $2 \in \phi(B)$ with $(2, 5) \in E_G$
- $6 \in \phi(C)$ and $(A, C) \in E_Q$: there is $1 \in \phi(A)$ with $(1, 6) \in E_G$
- $6 \in \phi(C)$ and $(B, C) \in E_Q$: there is $1 \in \phi(B)$ with $(1, 6) \in E_G$

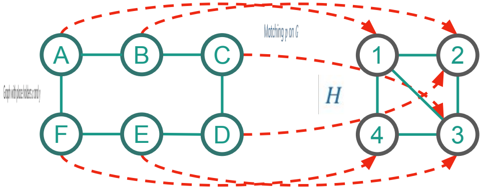
Morphism-Based Semantics

Graph Morphisms

ARE GRAPHS G AND H EQUAL/SIMILAR?

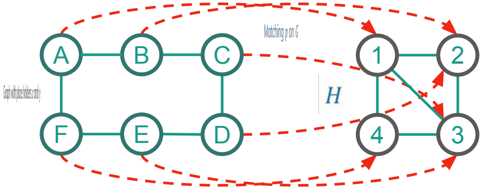


Graph Homomorphism



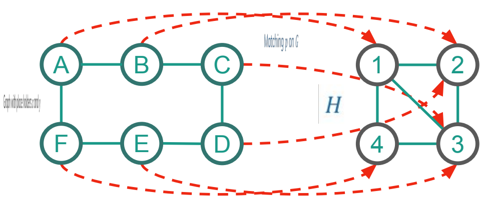
- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$

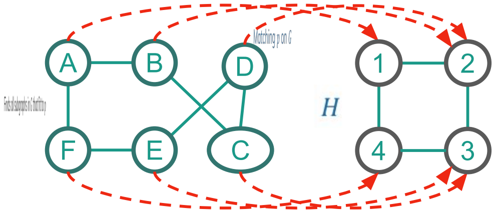
Graph Homomorphism



- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are homomorph, if there is a surjective function $\sigma\colon V_G \to V_H$ that preserves adjaceny:
  - if $(v_i, v_j) \in E_G$ then $(\sigma(v_i), \sigma(v_j)) \in E_H$

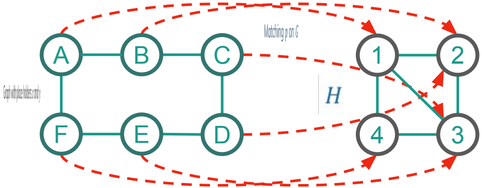
Graph Homomorphism

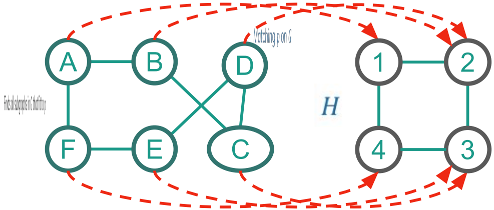




- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are homomorph, if there is a surjective function $\sigma \colon V_G \to V_H$ that preserves adjaceny:
  - if $(v_i, v_j) \in E_G$ then $(\sigma(v_i), \sigma(v_j)) \in E_H$

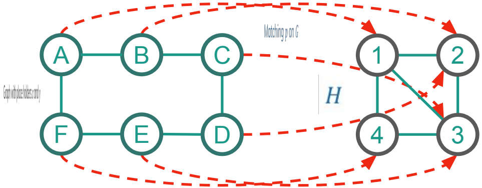
Graph Homomorphism

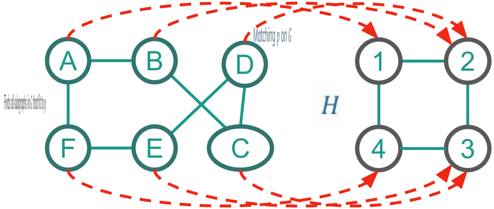




- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are homomorph, if there is a surjective function $\sigma\colon V_G \to V_H$ that preserves adjaceny:
  - if $(v_i, v_j) \in E_G$ then $(\sigma(v_i), \sigma(v_j)) \in E_H$
- σ is called a homomorphism from G to H

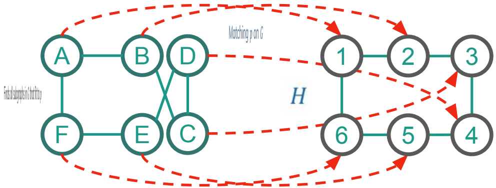
Graph Homomorphism





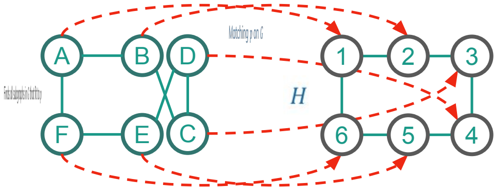
- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are homomorph, if there is a surjective function $\sigma\colon V_G \to V_H$ that preserves adjaceny:
  - if $(v_i, v_j) \in E_G$ then $(\sigma(v_i), \sigma(v_j)) \in E_H$
- σ is called a homomorphism from G to H
- Note: there may be multiple homomorphism between two graphs

Graph Isomorphism

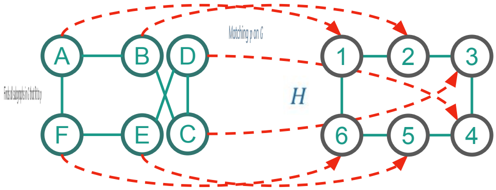


- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$

Graph Isomorphism

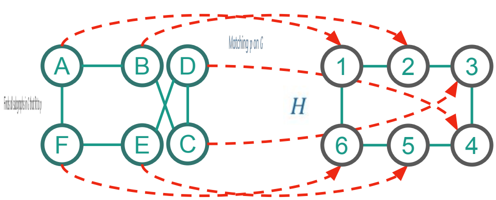


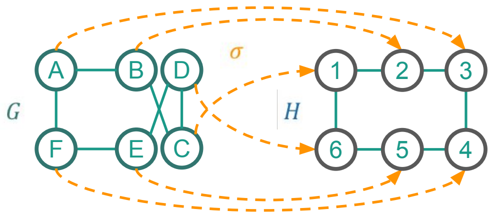
- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are isomorph, if there is a bijective function $\sigma\colon V_G \to V_H$ that preserves adjaceny and non-adjacency:
  - $(v_i, v_j) \in E_G$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_H$



- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are isomorph, if there is a bijective function $\sigma \colon V_G \to V_H$ that preserves adjaceny and non-adjacency:
  - $(v_i, v_j) \in E_G$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_H$
- σ is called a isomorphism between G and H

Graph Isomorphism





- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are isomorph, if there is a bijective function $\sigma\colon V_G \to V_H$ that preserves adjaceny and non-adjacency:
  - $(v_i, v_j) \in E_G$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_H$
- σ is called a isomorphism between G and H
- Note: there may be multiple isomorphisms between two graphs

Graph Similarities

Graph Homomorphism

- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are homomorph if
  - there is a surjective function $\sigma\colon V_G \rightarrow V_H$
  - such that, if $(v_i, v_j) \in E_G$ then $(\sigma(v_i), \sigma(v_j)) \in E_H$
    (σ preserves adjaceny)

Graph Isomorphism

- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are isomorph if
  - there is a bijective function $\sigma\colon V_G \rightarrow V_H$
  - such that, $(v_i, v_j) \in E_G$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_H$
    (σ preserves adjaceny and non-adjaceny)

Graph Similarities

Graph Homomorphism

- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are homomorph if
  - there is a surjective function $\sigma : V_G \rightarrow V_H$
  - such that, if $(v_i, v_j) \in E_G$ then $(\sigma(v_i), \sigma(v_j)) \in E_H$
    (σ preserves adjaceny)

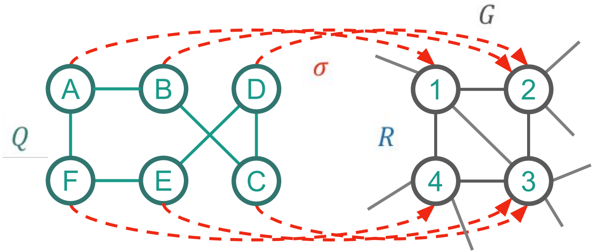
Graph Isomorphism

- Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$
- G and H are isomorph if
  - there is a bijective function $\sigma : V_G \rightarrow V_H$
  - such that, $(v_i, v_j) \in E_G$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_H$
    (σ preserves adjaceny and non-adjaceny)

Cool!

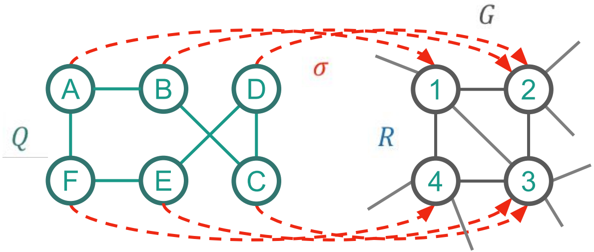
Can we apply this to find subgraphs?

Subgraph Homomorphism Query



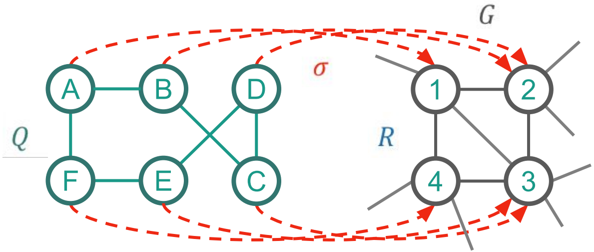
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$

Subgraph Homomorphism Query



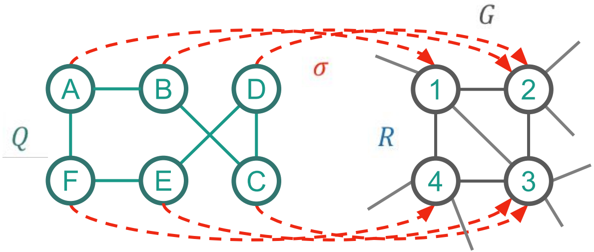
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- The subgraph $R = (V_R, E_R)$ is a result for Q on G if

Subgraph Homomorphism Query



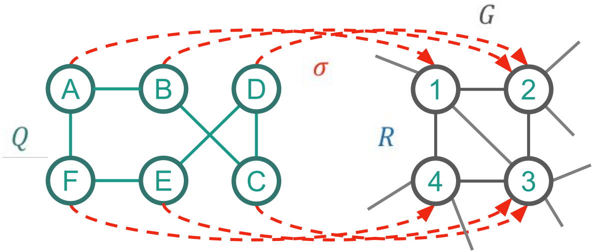
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- The subgraph $R = (V_R, E_R)$ is a result for Q on G if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G$ and
     (R is a subgraph of G)

Subgraph Homomorphism Query



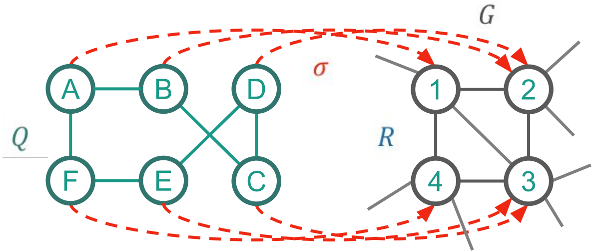
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- The subgraph $R = (V_R, E_R)$ is a result for Q on G if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G$ and
     (R is a subgraph of G)
  2. there is a surjective function $\sigma : V_Q \rightarrow V_R$ such that $(v_i, v_j) \in E_Q$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_R$
     (Q and R are homomorph and R contains no more edges than necessary)

Subgraph Homomorphism Query



- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- The subgraph $R = (V_R, E_R)$ is a result for Q on G if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G$ and
     (R is a subgraph of G)
  2. there is a surjective function $\sigma : V_Q \to V_R$ such that $(v_i, v_j) \in E_Q$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_R$
     (Q and R are homomorph and R contains no more edges than necessary)
  3. for all $v_Q \in V_Q$ the properties of v_Q and $\sigma(v_Q)$ match
  4. for all $(v_Q, w_Q) \in E_Q$ the properties of (v_Q, w_Q) and $(\sigma(v_Q), \sigma(w_Q))$ match

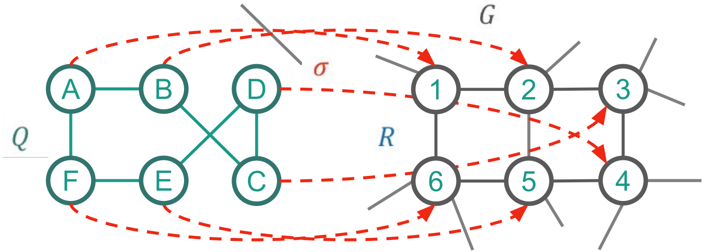
Subgraph Homomorphism Query



Notes

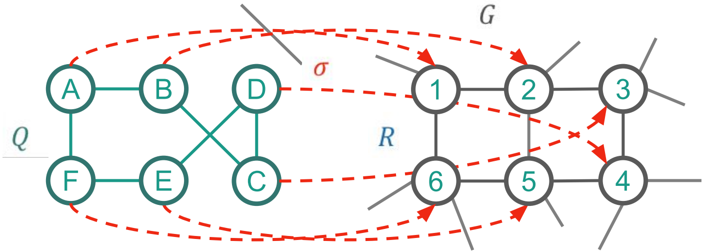
- Q can have more vertices or edges than R, i.e. in general $|V_Q| \geq |V_R|$ and $|E_Q| \geq |E_R|$
- R is not given, R has to be determined by the query mechanism (search problem)
- Vertices and edges can be part of multiple results

Subgraph Isomorphism Query



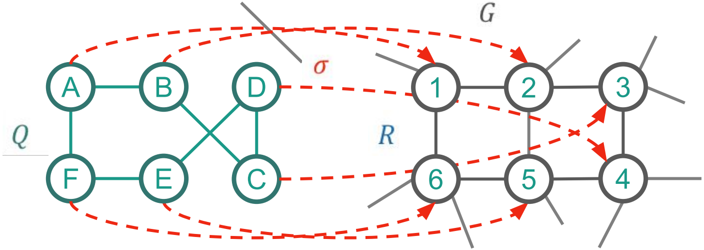
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$

Subgraph Isomorphism Query



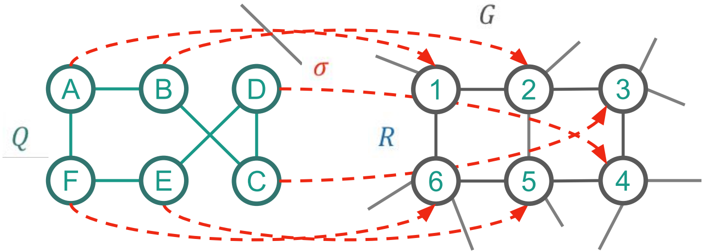
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- The subgraph $R = (V_R, E_R)$ is a result for Q on G if

Subgraph Isomorphism Query



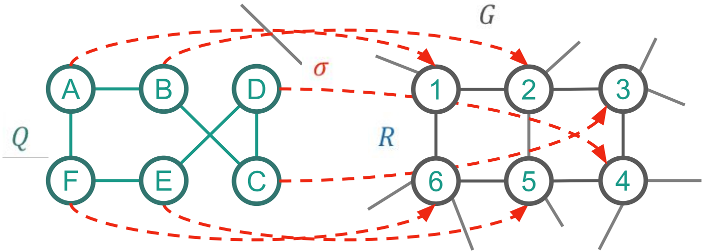
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- The subgraph $R = (V_R, E_R)$ is a result for Q on G if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G$ and
     (R is a subgraph of G)

Subgraph Isomorphism Query



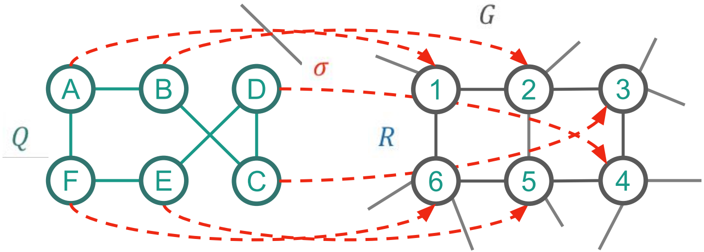
- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- The subgraph $R = (V_R, E_R)$ is a result for Q on G if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G$ and
     (R is a subgraph of G)
  2. there is a bijective function $\sigma\colon V_Q \to V_R$ such that $(v_i, v_j) \in E_Q$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_R$
     (Q and R are isomorph and R contains no more edges than necessary)

Subgraph Isomorphism Query



- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- The subgraph $R = (V_R, E_R)$ is a result for Q on G if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G$ and
     (R is a subgraph of G)
  2. there is a bijective function $\sigma\colon V_Q \to V_R$ such that $(v_i, v_j) \in E_Q$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_R$
     (Q and R are isomorph and R contains no more edges than necessary)
  3. for all $v_Q \in V_Q$ the properties of v_Q and $\sigma(v_Q)$ match
  4. for all $(v_Q, w_Q) \in E_Q$ the properties of (v_Q, w_Q) and $(\sigma(v_Q), \sigma(w_Q))$ match

Subgraph Isomorphism Query



Notes

- Q and R have always the same number of vertices and edges, i.e. $|V_Q| = |V_R|$ and $|E_Q| = |E_R|$
- R is not given, R has to be determined by the query mechanism (search problem)
- Vertices and edges can be part of multiple results

Subgraph Matching Semantics

Subraph Homomorphism

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- $R = (V_R, E_R)$ is a result for Q if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G G$;
  2. there is a surjective function $\sigma \colon V_Q \to V_R$
  3. such that $(v_i, v_j) \in E_Q$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_R$
  4. properties of vertices $v_Q \in V_Q$ and their images $\sigma(v_Q)$ match
  5. properties of edges $(v_Q, w_Q) \in E_Q$ and their images $(\sigma(v_Q), \sigma(w_Q)) \in E_R$ match
- Q can have more vertices and edges than R

Subraph Isomorphism

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- $R = (V_R, E_R)$ is a result for Q if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G G$;
  2. there is a bijective function $\sigma \colon V_Q \to V_R$
  3. such that $(v_i, v_j) \in E_Q$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_R$
  4. properties of vertices $v_Q \in V_Q$ and their images $\sigma(v_Q)$ match
  5. properties of edges $(v_Q, w_Q) \in E_Q$ and their images $(\sigma(v_Q), \sigma(w_Q)) \in E_R$ match
- Q and R always have the same number of vertices and edges

Subgraph Matching Semantics

Subraph Homomorphism

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- $R = (V_R, E_R)$ is a result for Q if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G G$;
  2. there is a surjective function $\sigma \colon V_Q \to V_R$
  3. such that $(v_i, v_j) \in E_Q$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_R$
  4. properties of vertices $v_Q \in V_Q$ and their images $\sigma(v_Q)$ match
  5. properties of edges $(v_Q, w_Q) \in E_Q$ and their images $(\sigma(v_Q), \sigma(w_Q)) \in E_R$ match
- Q can have more vertices and edges than R

Subraph Isomorphism

- Given a query graph $Q = (V_Q, E_Q)$ and a data graph $G = (V_G, E_G)$
- $R = (V_R, E_R)$ is a result for Q if
  1. $V_R \subseteq V_G$ and $E_R \subseteq E_G G$;
  2. there is a bijective function $\sigma \colon V_Q \to V_R$
  3. such that $(v_i, v_j) \in E_Q$ if and only if $(\sigma(v_i), \sigma(v_j)) \in E_R$
  4. properties of vertices $v_Q \in V_Q$ and their images $\sigma(v_Q)$ match
  5. properties of edges $(v_Q, w_Q) \in E_Q$ and their images $(\sigma(v_Q), \sigma(w_Q)) \in E_R$ match
- Q and R always have the same number of vertices and edges

A single vertex in V_G can be matched multiple times in a homomorphic subgraph but only once in an isomorphic subgraph

Why Homomorphisms are useful...

Example

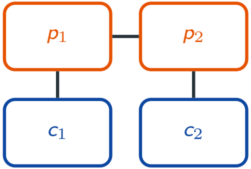
Look for all pairs of friends and the city each friend lives in

Why Homomorphisms are useful...

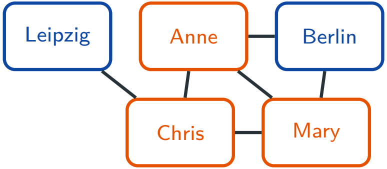
Example

Look for all pairs of friends and the city each friend lives in

Query graph Q



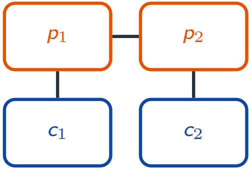
Data graph G



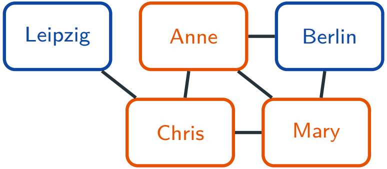
Example

Look for all pairs of friends and the city each friend lives in

Query graph Q



Data graph G



Assuming subgraph isomorphism, which tuples (c_1, p_1, p_2, c_2) represent results?

A: (Leipzig, Chris, Anne, Berlin)

B: (Leipzig, Chris, Mary, Berlin)

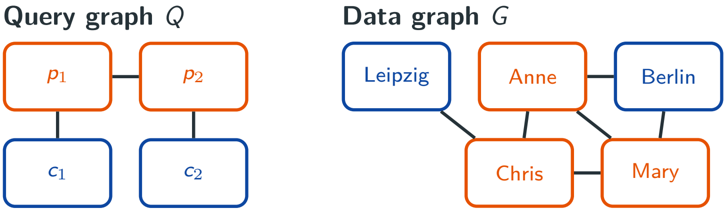
C: (Berlin, Anne, Mary, Berlin)

D: (Berlin, Anne, Chris, Leipzig)

Why Homomorphisms are useful...

Example

Look for all pairs of friends and the city each friend lives in



Assuming subgraph isomorphism, which tuples (c_1, p_1, p_2, c_2) represent results?

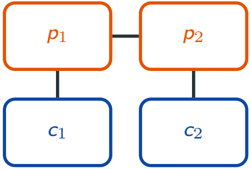
- A: (Leipzig, Chris, Anne, Berlin)
- B: (Leipzig, Chris, Mary, Berlin)
- C: (Berlin, Anne, Mary, Berlin) no result: c_1 and c_2 cannot be matched to the same vertex "Berlin"
- D: (Berlin, Anne, Chris, Leipzig)

Why Homomorphisms are useful...

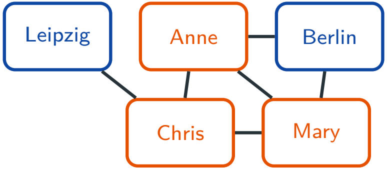
Example

Look for all pairs of friends and the city each friend lives in

Query graph Q



Data graph G



Isomorphism finds only friends living in different cities

(Leipzig, Chris, Anne, Berlin), (Leipzig, Chris, Mary, Berlin), ...and permutations of these

Homomorphism additionally finds friends living in the same city

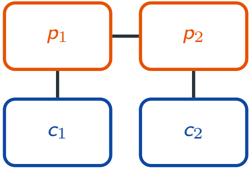
(Berlin, Mary, Anne, Berlin), (Berlin, Anne, Mary, Berlin)

Why Homomorphisms are useful...

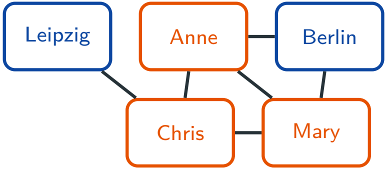
Example

Look for all pairs of friends and the city each friend lives in

Query graph Q



Data graph G



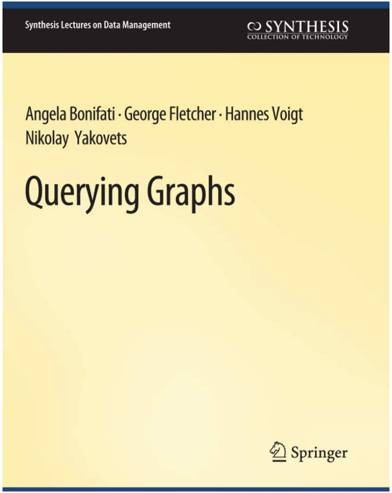
Isomorphism finds only friends living in different cities

(Leipzig, Chris, Anne, Berlin), (Leipzig, Chris, Mary, Berlin), ...and permutations of these

Homomorphism additionally finds friends living in the same city

(Berlin, Mary, Anne, Berlin), (Berlin, Anne, Mary, Berlin), (Berlin, Anne, Anne, Berlin), ...

Literature



Bonifati et al., *Querying Graphs*, 2018

References

Bonifati, Angela, George H. L. Fletcher, Hannes Voigt, and Nikolay Yakovets (2018). *Querying Graphs*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers. DOI: 10.2200/S00873ED1V01Y201808DTM051. URL: https://doi.org/10.2200/S00873ED1V01Y201808DTM051.

Miller, John A., Lakshmish Ramaswamy, Krys J. Kochut, and Arash Fard (2015). "Research Directions for Big Data Graph Analytics". In: *IEEE International Congress on Big Data, New York City, NY, USA, June 27 - July 2*. Ed. by Barbara Carminati and Latifur Khan. IEEE Computer Society, pp. 785–794. DOI: 10.1109/BIGDATACONGRESS.2015.132. URL: https://doi.org/10.1109/BigDataCongress.2015.132.