CptS 415 Big Data

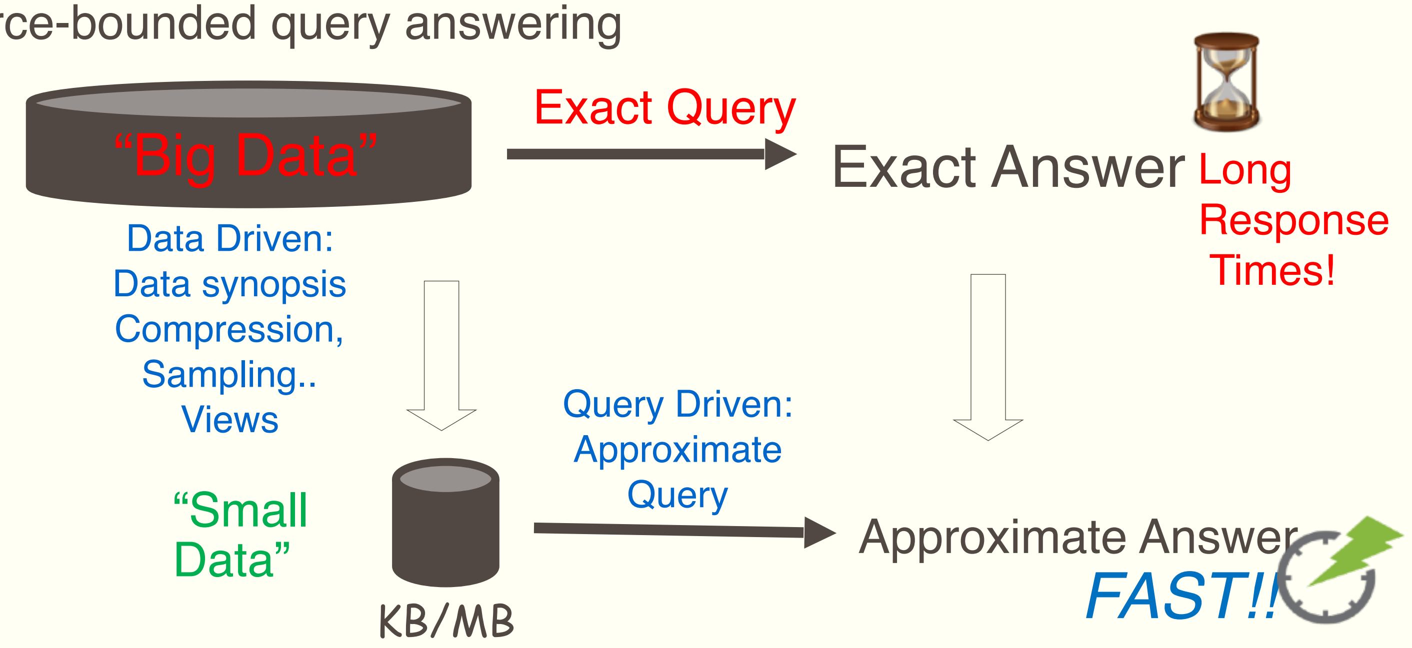
# Approximate Query Processing

Srini Badri

Acknowledgement: Tinghui Wang

# Make Case for Computationally Efficient Queries

- Approximate Query Answering
  - Query-driven approximation
    - Rewrite queries to computationally efficient query classes
  - Data-drive approximation
    - Compact data synopses, materialized views, compression, summaries, sketches, spanners
    - Resource-bounded query answering

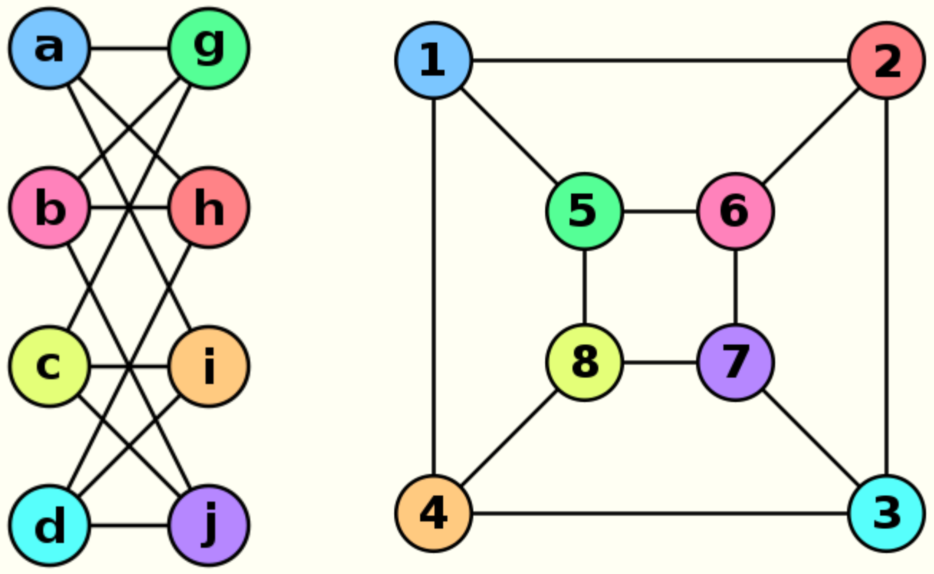


# Query Driven Approximation: Graph Pattern Matching

- Input: A pattern graph $P$, a data graph $G$, matching semantics
- Output: correspondence from $P$ to $G$
  - Matching relation/function
  - Matched nodes, edges, subgraphs
- A “special case” of general graph matching.
  - Difference: semantic of P (as a graph or a pattern)
- Variants of graph (pattern) matching
  - Small pattern vs. large graph matching
  - Single data graph vs. multiple graphs
  - Rich semantics vs. simple label equality
  - Flexible matching semantics vs. strict matching functions
  - Approximate matching vs. exact alignment

# Graph Isomorphism

- Graphs *G* and *H* are said to be Isomorphic if:
  - there exists a bijective relation between vertices of *G* and *H*:
    - *f: V(G) -> V(H)*
  - for any two vertices *u* and *v* that are adjacent in *G*, *f(u)* and *f(v)* are adjacent in *H*
- Subgraph Isomorphism:
  - Graph *G* contains a subgraph $G_o$ that is isomorphic to *H*




source: https://en.wikipedia.org/wiki/Graph_isomorphism

# Matching by Subgraph Isomorphism

- Input: A direct graph $G$ and a graph pattern $P$
- Output: All subgraphs of $G$ that are isomorphic to $P$
- Complexity: NP-Complete
  - Remains NP-Hard even when
  - P is a forest and G is a tree
  - P is a tree and G is acyclic
- P-TIME if $P$ is a tree and $G$ is a forest

# Ullmann’s algorithm

$$\begin{bmatrix}1&1&1&1\\1&1&1&1\\0&1&0&0\end{bmatrix}$$

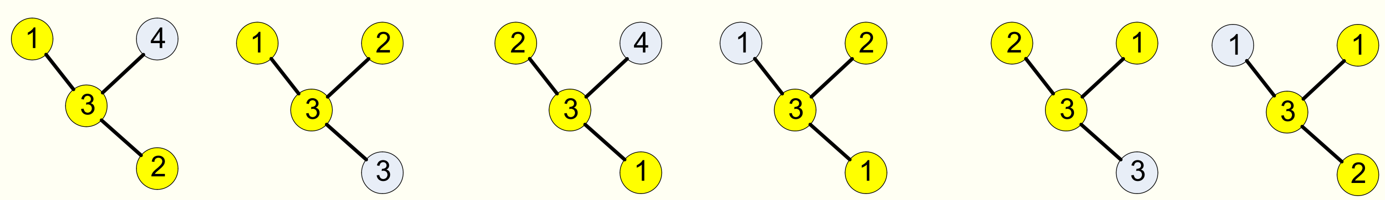
Step 1. Set up matrix $M_{n\times m}$

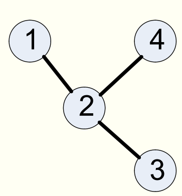
$$\begin{bmatrix}1&0&0&0\\1&1&1&1\\0&1&0&0\end{bmatrix}\quad\begin{bmatrix}0&\boxed{1}&0&0\\1&1&1&1\\0&\boxed{1}&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&1&0\\1&1&1&1\\0&1&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&1\\1&1&1&1\\0&1&0&0\end{bmatrix}$$

Step 2. iteratively change 1 to 0 subject to M’; maximum depth: |Q|

$$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\end{bmatrix}\quad\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&1&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&1&0\\1&0&0&0\\0&1&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&1&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&1\\1&0&0&0\\0&1&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix}$$

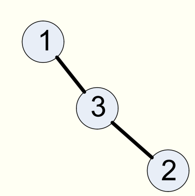
Step 3. verify.





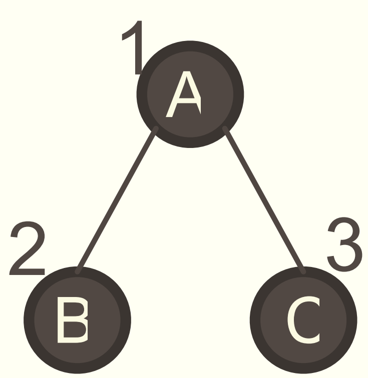
$$P = M'(M'A_H)^T = \begin{bmatrix}0&0&1\\0&0&1\\1&1&0\end{bmatrix}$$

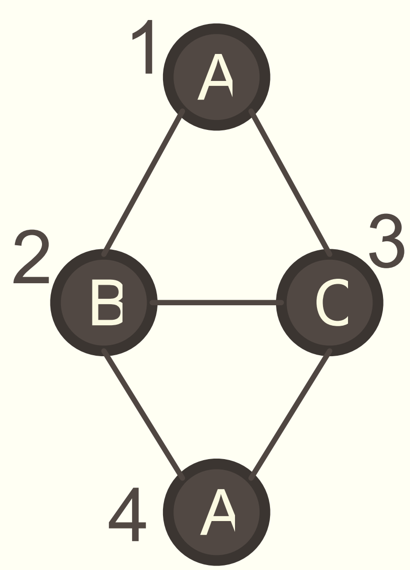
compared with $A_G = \begin{bmatrix}0&0&1\\0&0&1\\1&1&0\end{bmatrix}$



# VF2

- Considering two graphs $Q$ and $G$, the (sub)graph isomorphism from $Q$ to $G$ is expressed as the set of $pairs(n, m)$ (with $n \in Q,\ m \in G$)
- Idea: finding the (sub)graph isomorphism between Q and G is a sequence of state transition.
- an intermediate state $s$ denotes a partial mapping from Q to G



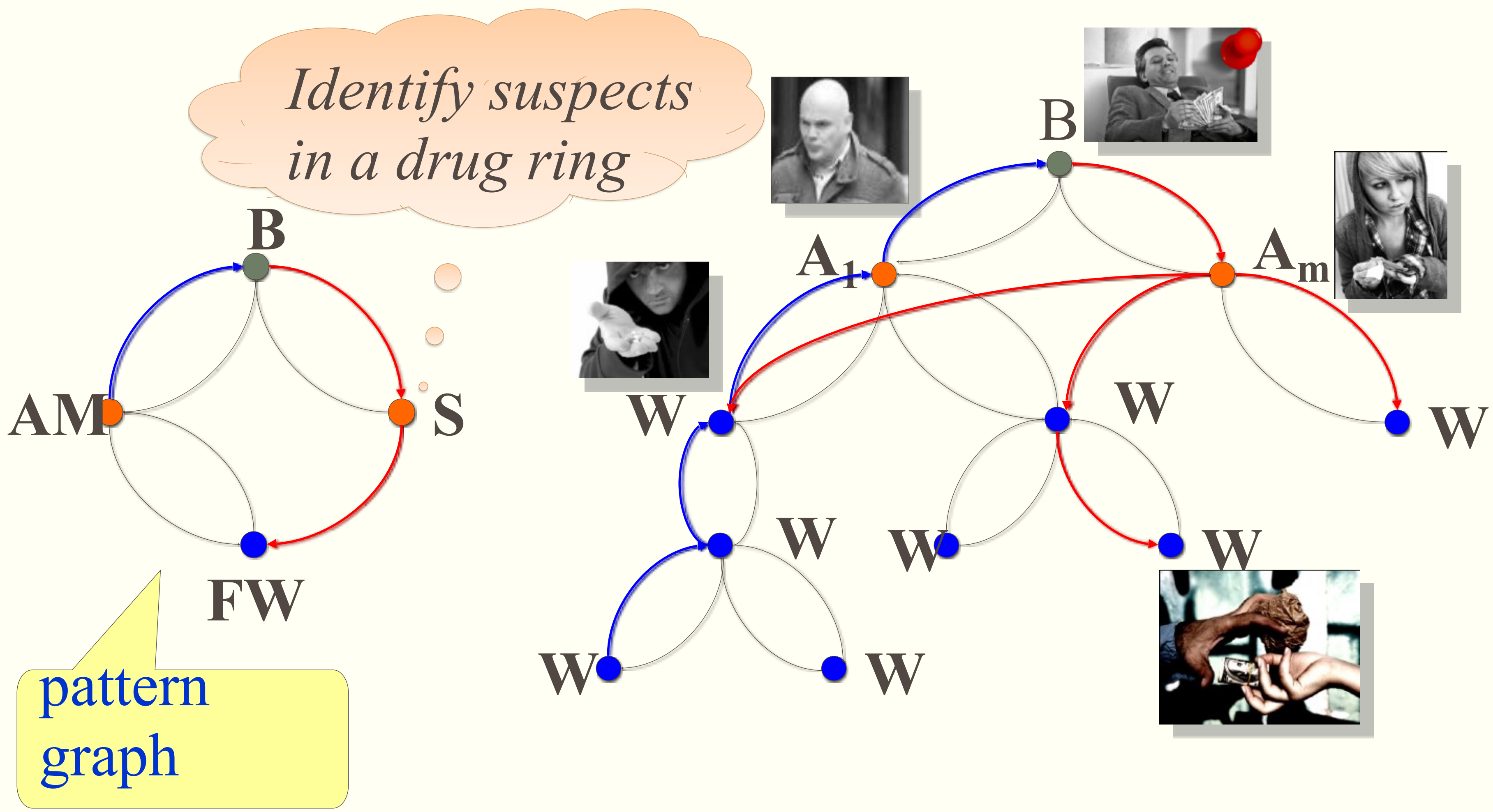




| | |
|---|---|
| (1, 1) | (1, 4) |
| (2, 2) | (2, 2) |
| (3, 3) | (3, 3) |

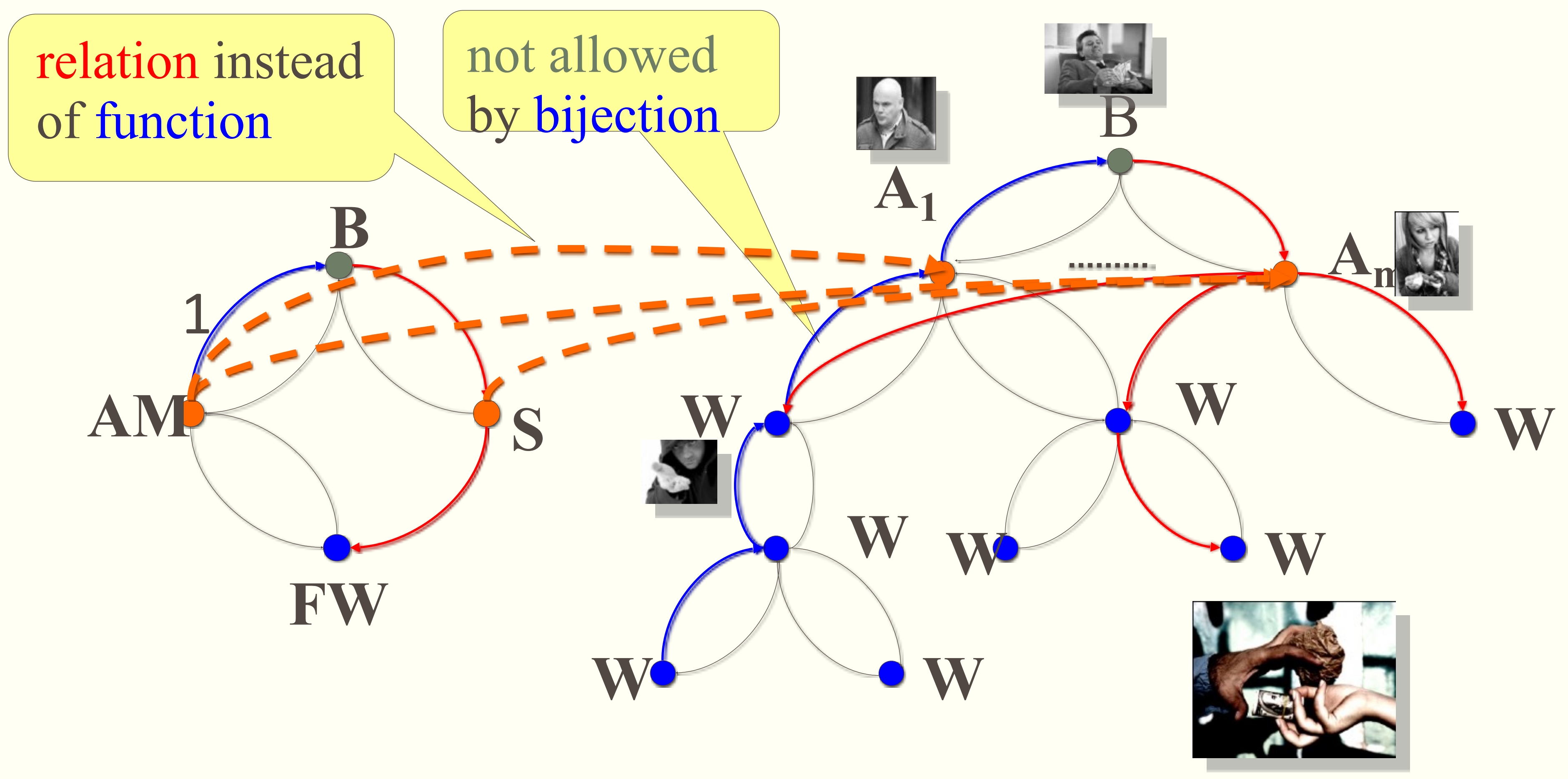
| | Intermediate States |
|---|---|
| s1 | (1,1) |
| s2 | (1,1) (2,2) |
| s3 | (1,1)(2,2)(3,3) |

# Pattern Matching in Social Graph

Find all matches of a pattern in a graph

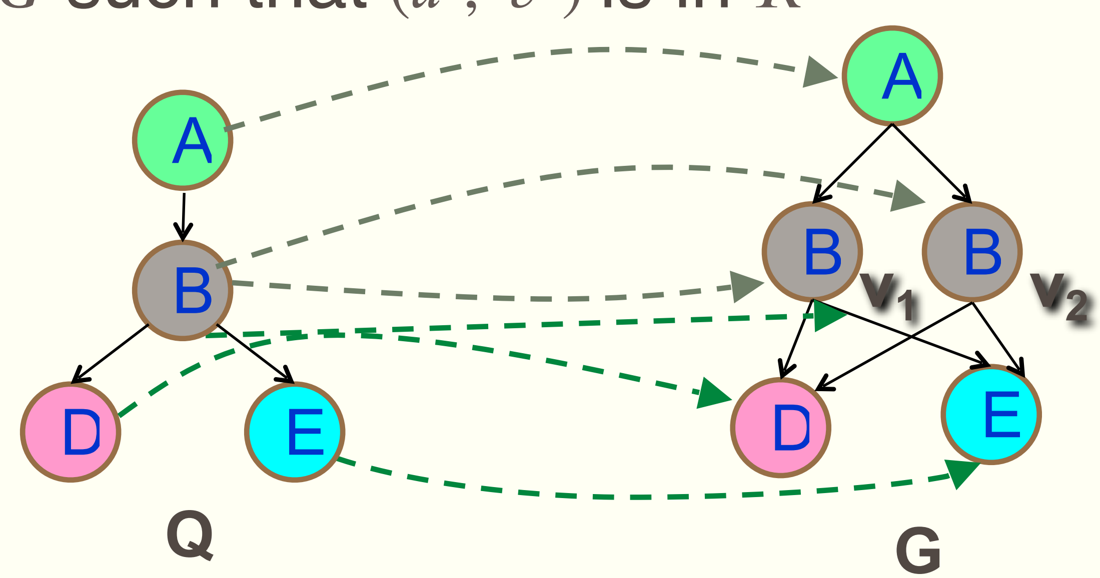


# Pattern Matching in Social Graphs



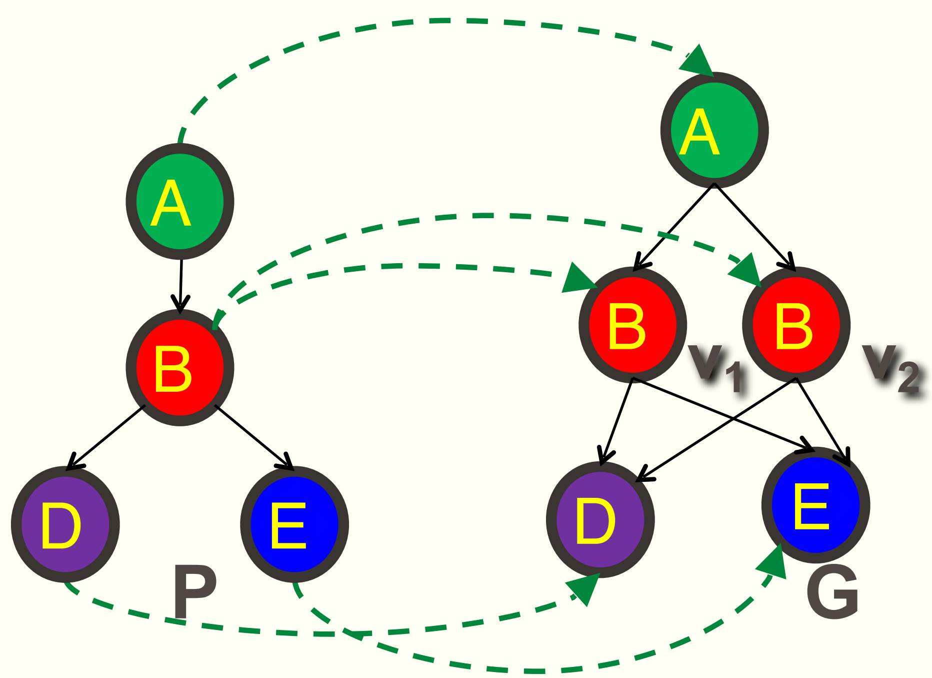
# Graph Simulation

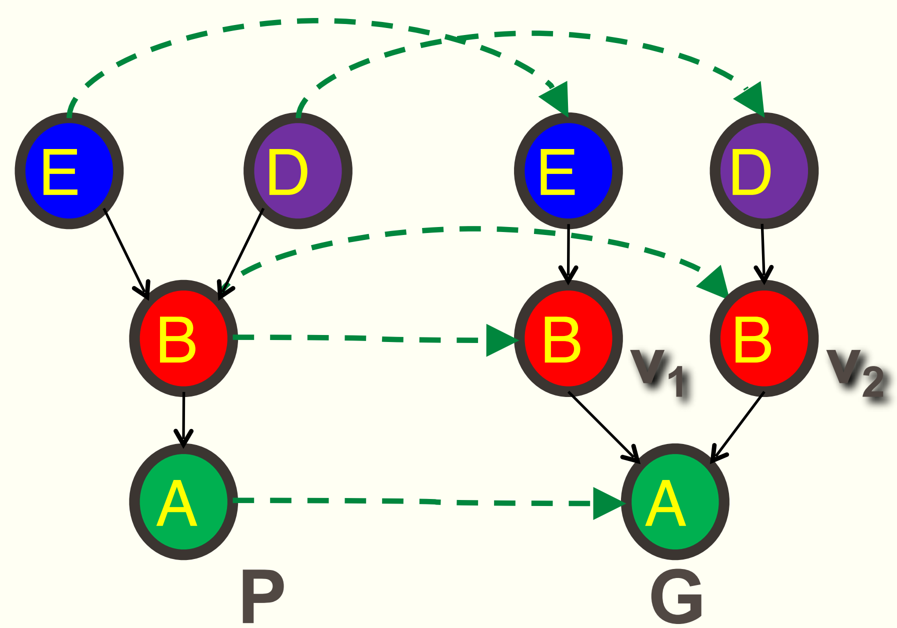
- A binary relation $R$ on the nodes of $Q$ and the nodes of $G$:
- For each node $u$ in $Q$, there exists a node $v$ in $G$ such that $(u, v)$ is in $R$, and $u$ and $v$ have the same label;
- If there exists an edge $(u, u')$ in $Q$ and each pair $(u, v)$ is in $R$, then there exists an edge $(v, v')$ in $G$ such that $(u', v')$ is in $R$



# SubGraph Isomorphism Vs. Graph Simulation

- Node label equivalence Vs. Node search constraints
- Bijective function vs. Many-to-many relation



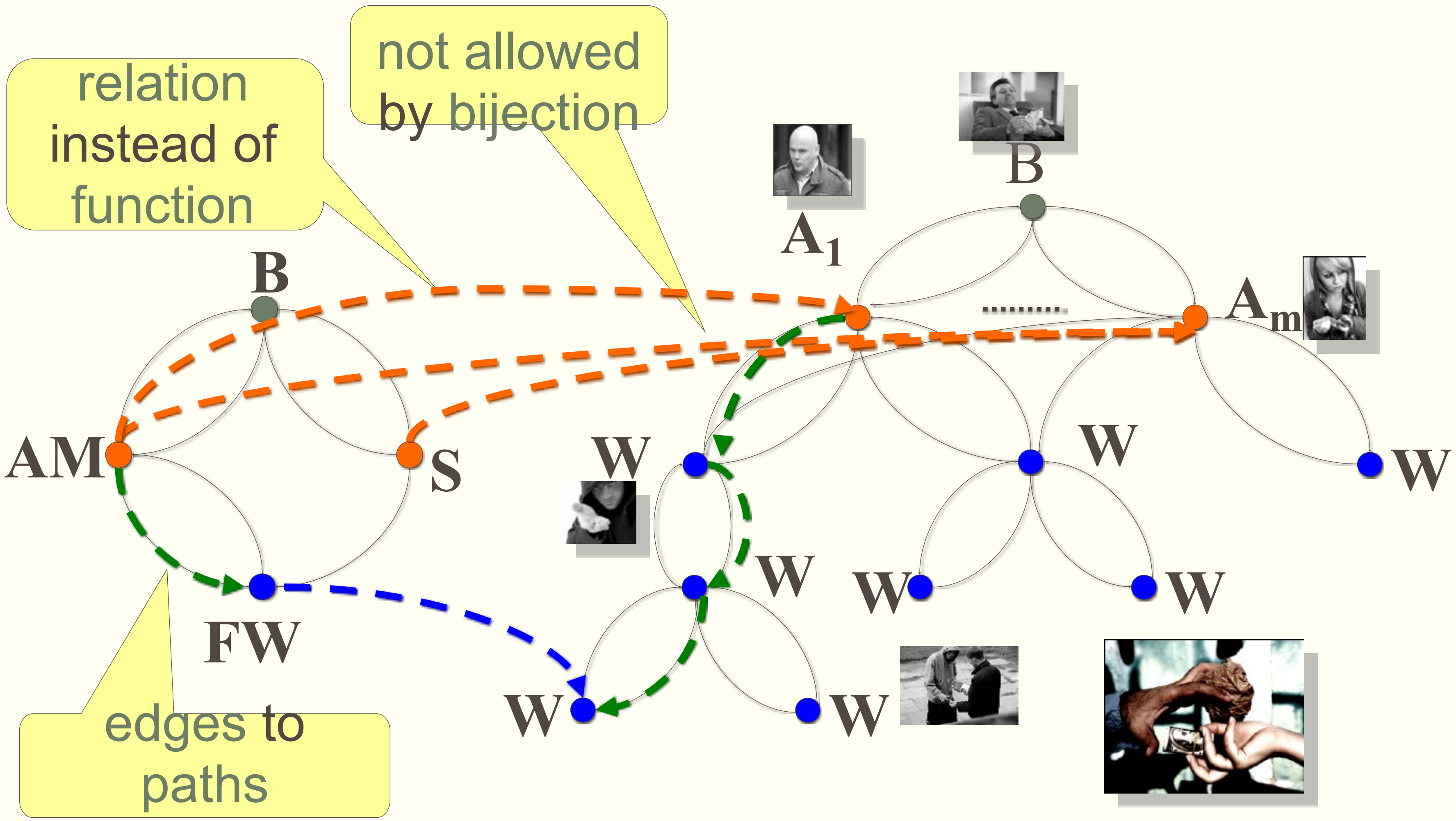
# Matching by Graph Simulation

- Input: A directed graph G, a graph pattern Q
- Output: The maximum simulation relation R
- Maximum simulation relation: always exists and is unique
  - If a match relation exists, then there exists a maximum one
  - Otherwise, it is the empty set – still maximum
- Complexity: $O\Big(\big(|V| + |V_Q|\big)\big(|E| + |E_Q|\big)\Big)$
- The output is a unique relation, possibly of size $|Q| \cdot |V|$

# Algorithm for Graph Simulation

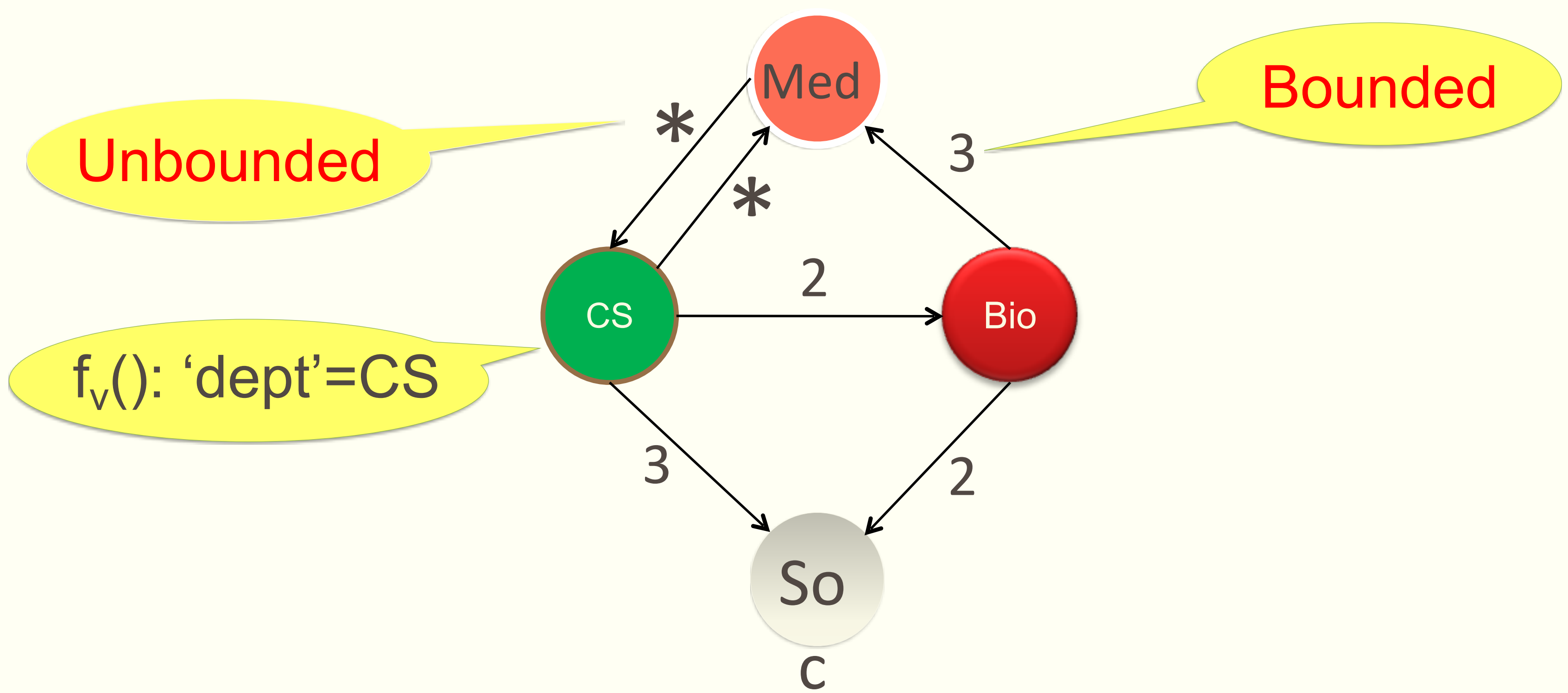
- Similarity (P)
  - For all nodes $u$ in $Q$ do:
    - $sim(u) \leftarrow$ the set of candidate matches $w$ in $G$
  - While there exists $(u, v)$ in $Q$ and $w$ in $sim(u)$ (in G) that violate the simulation condition
    - $sim(u) \leftarrow sim(u) - \{w\}$
  - Output $sim(u)$ for all $u$ in $Q$
- Initial match:
  - With the same label
  - If $u$ has an outgoing edge, so does $w$
- Simulation Condition: $successor(w) \cap sim(v) = \phi$
  - There exist an edge from $u$ to $v$ in $Q$, but the candidate $w$ (in G) of $u$ has no corresponding edge to a node $w'$ (in G) that matches $v$

# Pattern matching in social graphs



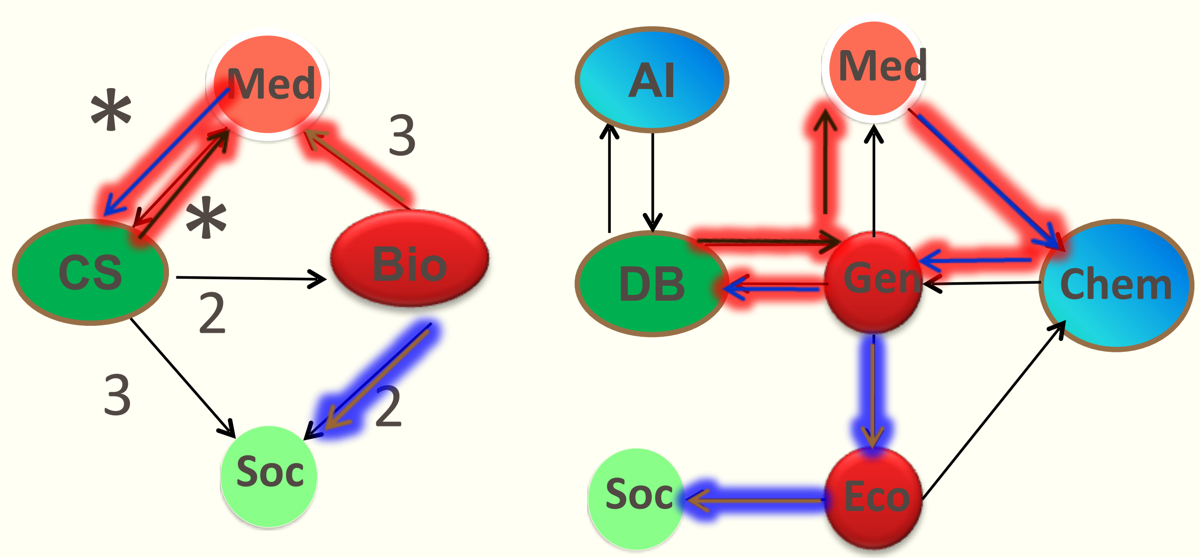
# Bounded Patterns

- Pattern Graph: $Q = (V_Q, E_Q, f_v, f_e)$
  - $f_v(u)$: a conjunction of $A\ op\ a$, $op$ in <, <=, ==, !=, >, >=
  - $f_e(u,\ u')$: a constant $k$ or a symbol $*$



# Bounded Simulation

- $G = (V, E, f_A)$ matches $Q = (V_Q,\ E_Q, f_v,\ f_e)$ via bounded simulation if there exists a binary relation $S \subseteq V_Q \times V$ such that
  - $S$ is a total mapping
  - $S$ satisfies search conditions and bounds on edge-to-path mapping

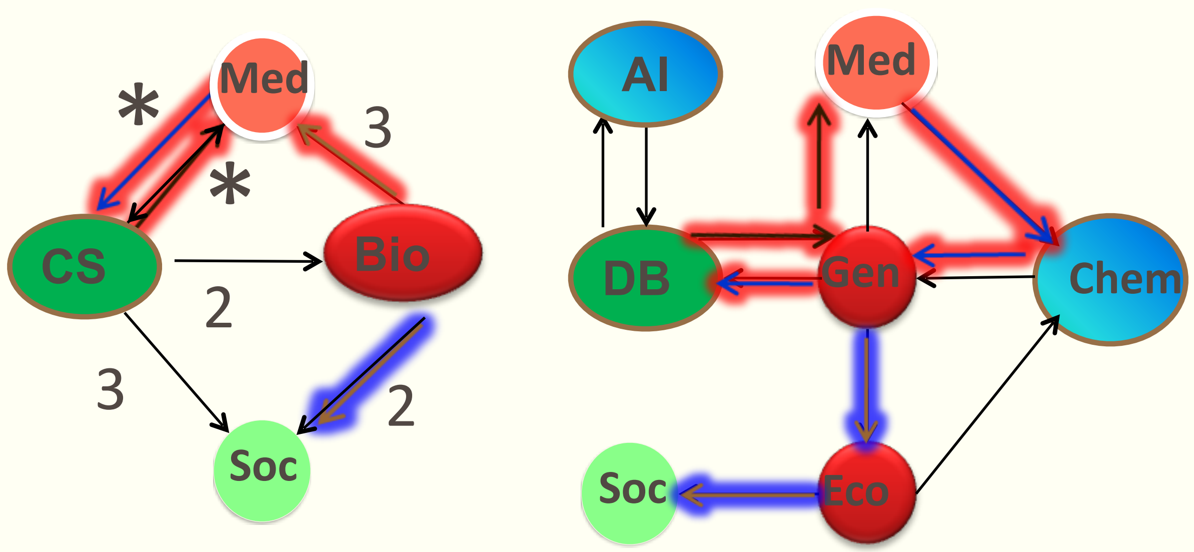



match

| |
|---|
| (CS, DB) |
| (Bio, Gen) |
| (Med, Med) |
| (Soc, Soc) |

# Bounded Simulation

- Total mapping:
  - For each $u \in V_Q$, there exists $v \in V$ such that $(u, v) \in S$
- For each $(u,\ v) \in S$
  - Attributes $f_A(v)$ satisfies predicate $f_v(u)$
  - Each $(u,\ u')$ in $E_Q$ is mapped to a path from $v$ to $v'$ of length $f_e(u,u')$ in G, where $(u',\ v') \in S$




match

| |
|---|
| (CS, DB) |
| (Bio, Gen) |
| (Med, Med) |
| (Soc, Soc) |

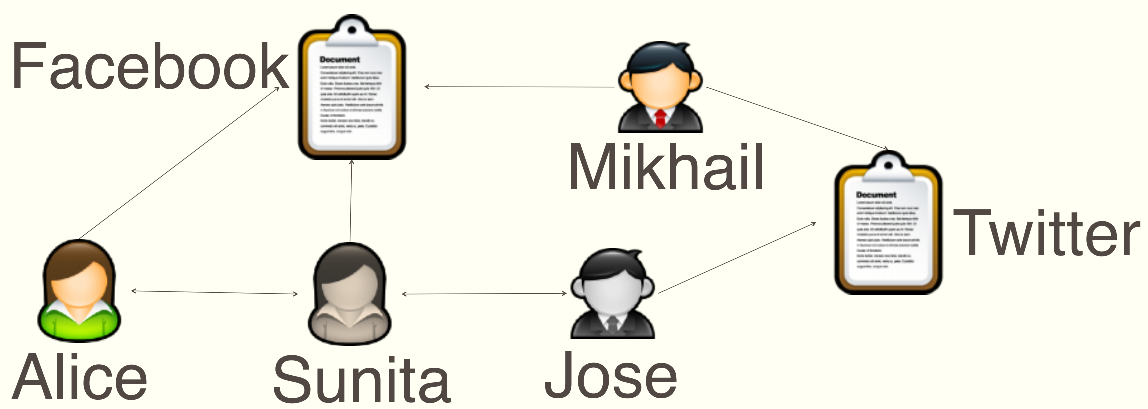
# Complexity

- Input: A directed graph G, a graph pattern Q
- Output: $Q(G)$, the unique maximum matching relation

$$Q\left(|V| \cdot |E| + \left|E_Q\right| \cdot |V|^2 + \left|V_Q\right| \cdot |V|\right)$$

- Query driven approximation:
  - Use bounded simulation instead of subgraph isomorphism
- Criteria:
  - Lower complexity
  - Effectiveness: The query answers are sensible

# Edge Relations




(Alice, Facebook)

(Alice, Sunita)

(Jose, Twitter)

(Jose, Sunita)

(Mikhail, Facebook)

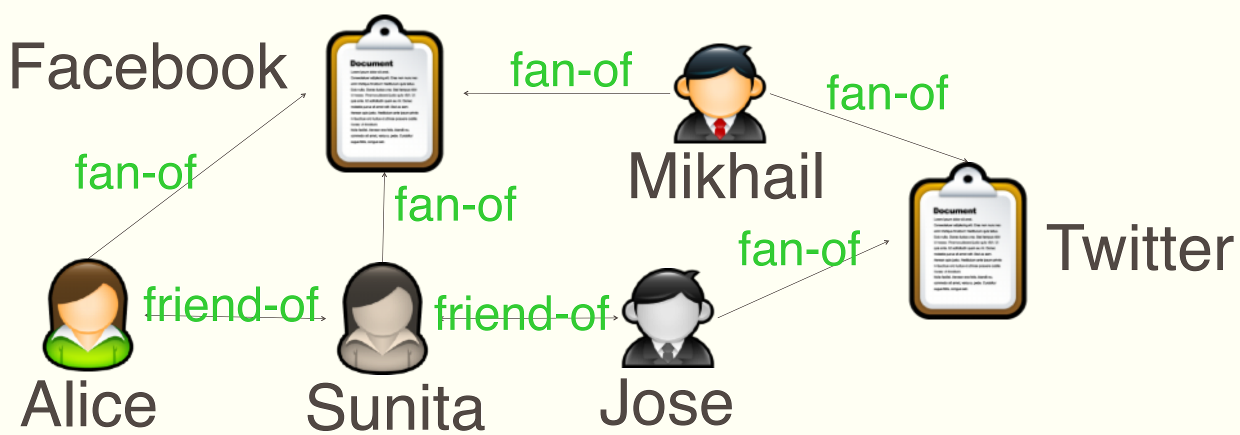
(Mikhail, Twitter)

(Sunita, Facebook)

(Sunita, Alice)

(Sunita, Jose)

# Edge Relations




(Alice, fan-of, Facebook)

(Alice, friend-of, Sunita)

(Jose, fan-of, Twitter)

(Jose, friend-of, Sunita)

(Mikhail, fan-of, Facebook)

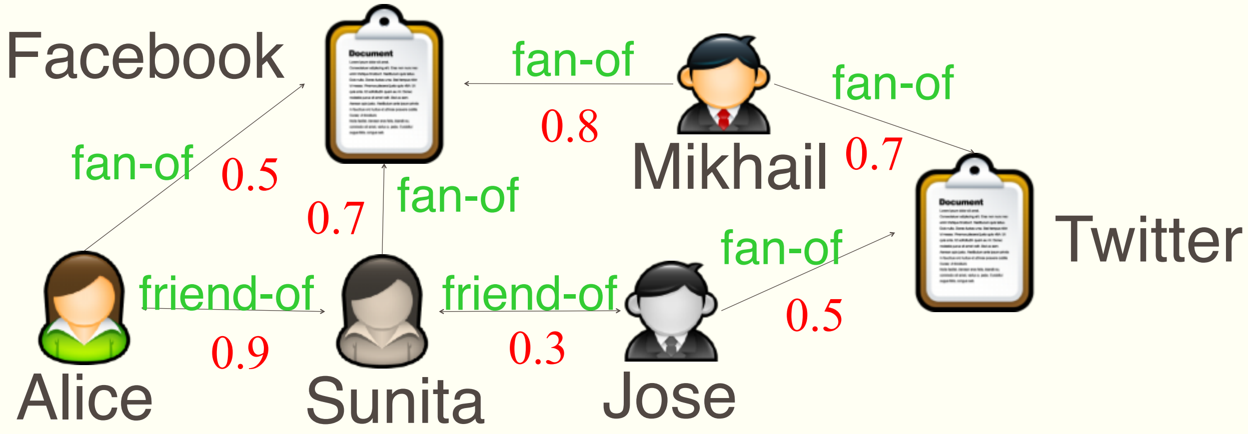
(Mikhail, fan-of, Twitter)

(Sunita, fan-of, Facebook)

(Sunita, friend-of, Alice)

(Sunita, friend-of, Jose)

# Edge Relations




(Alice, fan-of, 0.5, Facebook)

(Alice, friend-of, 0.9, Sunita)

(Jose, fan-of, 0.5, Twitter)

(Jose, friend-of, 0.3, Sunita)

(Mikhail, fan-of, 0.8, Facebook)

(Mikhail, fan-of, 0.7, Twitter)

(Sunita, fan-of, 0.7, Facebook)

(Sunita, friend-of, 0.9, Alice)

(Sunita, friend-of, 0.3, Jose)

# Regular Patterns

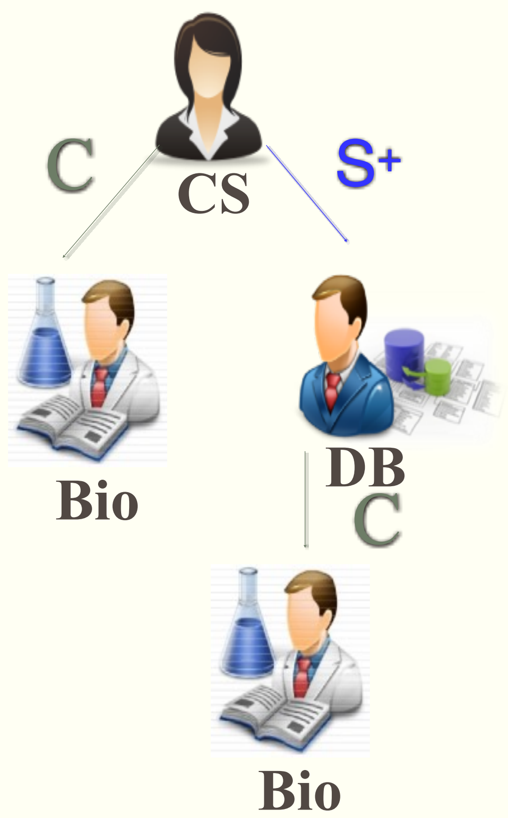
- Pattern Graph: $Q = (V_Q, E_Q, f_v, f_e)$
  - $f_v(u)$: a conjunction of $A \ op \ a$, $op$ in <, <=, ==, !=, >, >=
  - $f_e(u, u')$: a regular expression of the form:

$$f_e ::= c \mid c^{\leq k} \mid c^{+} \mid FF$$

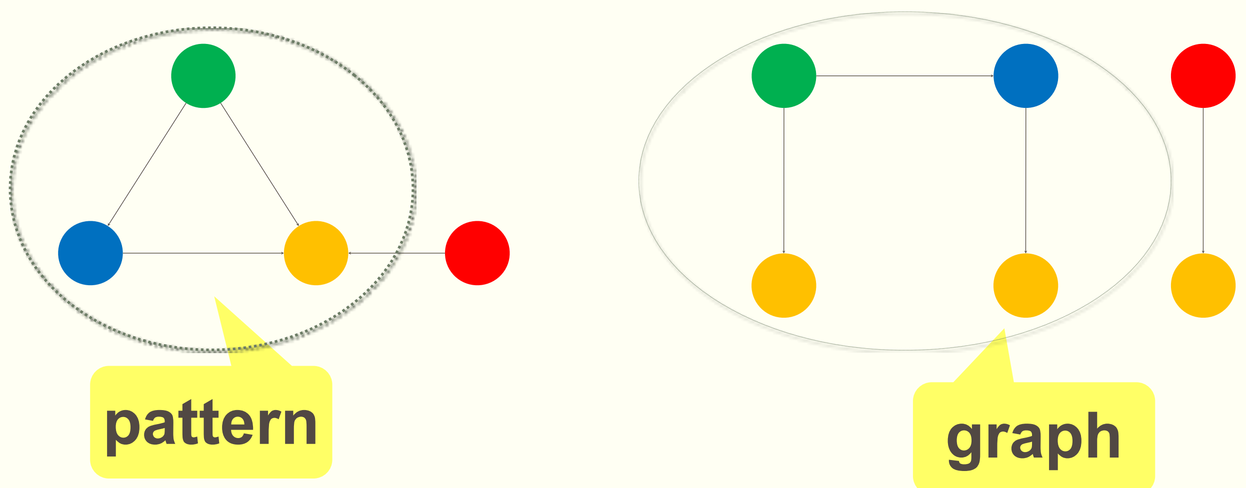
- Complexity:

$$Q\left(|V| \cdot |E| + m \cdot |E_Q| \cdot |V|^2 + |V_Q| \cdot |V|\right)$$

- Bounded simulation is a special case:
  - Single color $c$, hence $m = 1$
  - $f_e(u, u') = c$

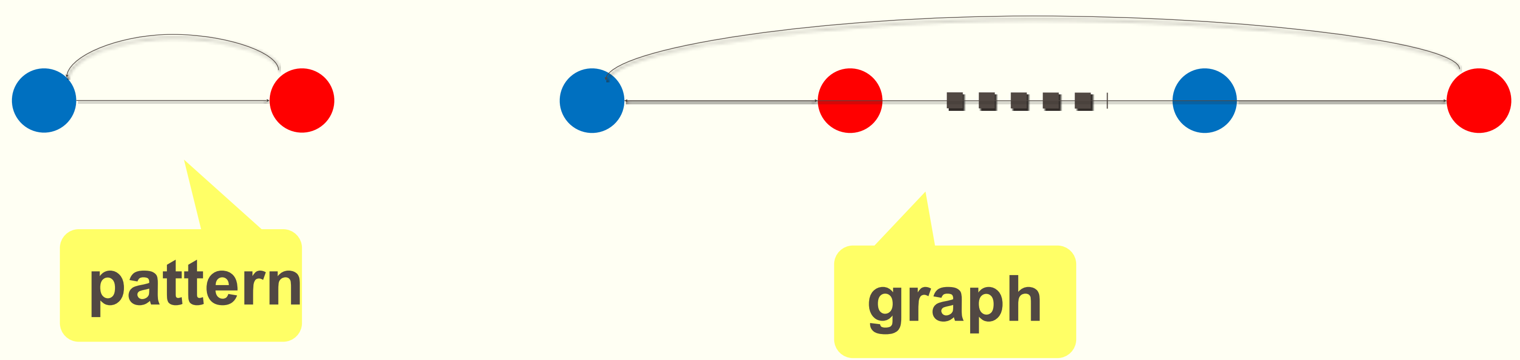


# Limitation of Graph Simulation




- A disconnected graph matches a connected pattern
- The yellow node in the pattern has 3 parents, in contrast to 1 in the graph
- An undirected cycle matches a tree
- Issue Identified: Simulation does not preserve the topology well in matching

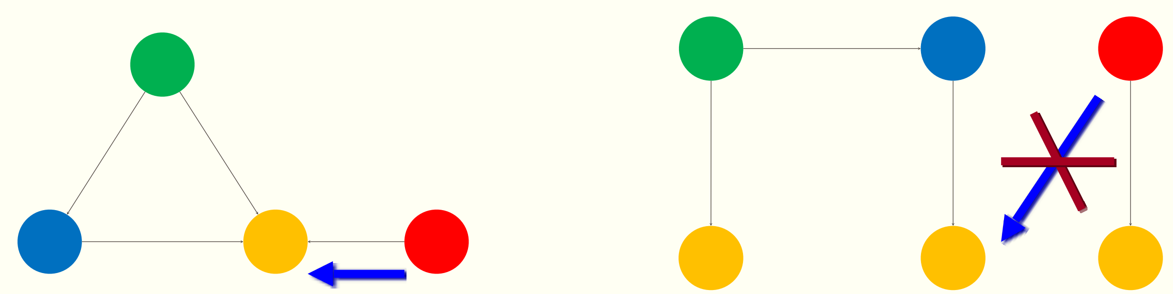
# Limitation of Graph Simulation




- A cycle with two nodes matches a cycle of unbounded length
- The match relation may be excessively large
- When social distances increase, the closeness of relationship decrease
- Issues identified: The need for revising simulation to enforce locality

# Dual Simulation

- $G = (V, E, f_A)$ matches $Q = (V_Q,\ E_Q, f_v,\ f_e)$ via bounded simulation if there exists a binary relation $S \subseteq V_Q \times V$ such that
  - $S$ is a total mapping
  - $S$ satisfies search conditions
  - $S$ preserves both "child" and "parent" relationships

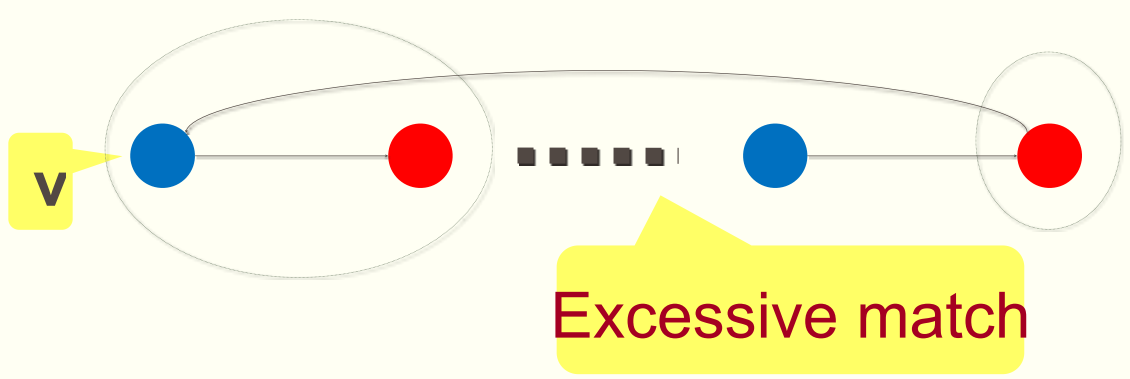


- Preserve "parent" relationships and connectivity

# Locality

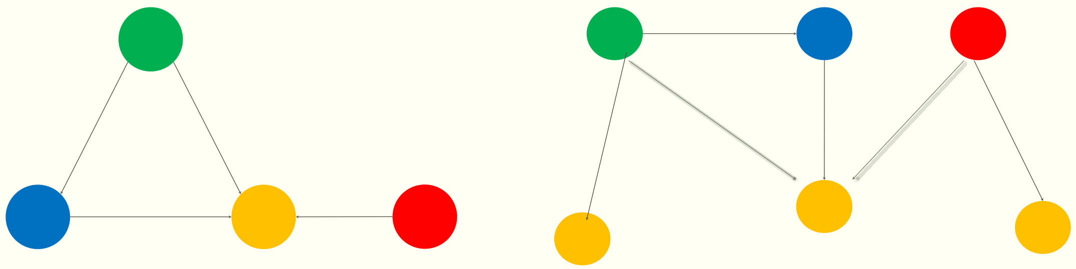
- Diameter $d_Q$
  - The maximum shortest distance (undirected path)


- $d_Q$-radius subgraph $G[v,\ d_Q]$, centered at $v$, with $d_Q$ hops



# Strong Simulation

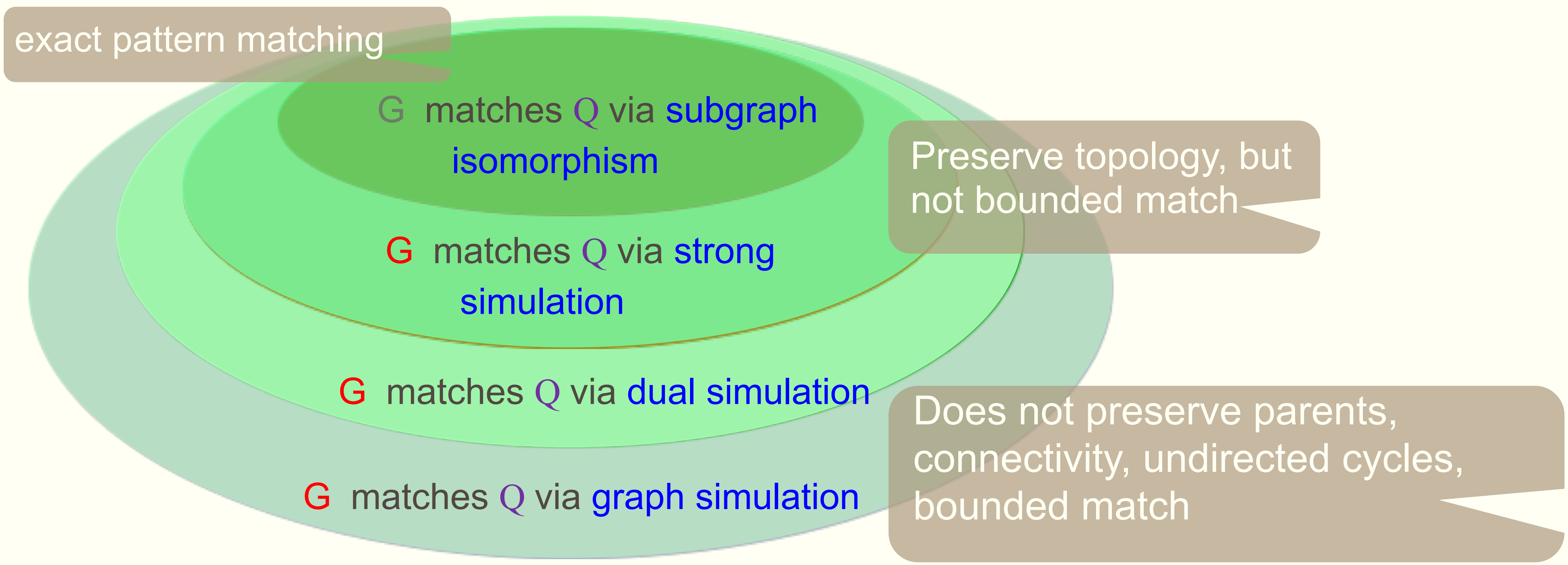
- $G$ matches $Q$ via strong simulation, if there exists a node $v$ in $G$ such that $G[v,\ d_Q]$ matches $Q$ via dual simulation
  - Duality
  - Local



- Complexity: cubic time

$$O\left(|V|\left(|V| + \left(|V_Q| + |E_Q|\right)\left(|V| + |E|\right)\right)\right)$$

# Summary


exact pattern matching
G matches Q via subgraph isomorphism
G matches Q via strong simulation
G matches Q via dual simulation
G matches Q via graph simulation
Preserve topology, but not bounded match
Does not preserve parents, connectivity, undirected cycles, bounded match

# Summary

| matching | complexity | match size |
|---|---|---|
| subgraph isomorphism | NP-complete | |
| graph simulation | quadratic time | |
| bounded simulation | cubic time | |
| regular matching | cubic time | |
| strong simulation | cubic time | |

# Paper to Review

- J. Lee, W. Han, R. Kasperovics, J. Lee. An In-depth Comparison of Subgraph Isomorphism Algorithms in Graph Databases, VLDB, 2012. http://www.vldb.org/pvldb/vol6/p133-han.pdf
- L. P. Cordella, P. Foggia, C. Sansone, M. Vento. A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs, IEEE Trans. Pattern Anal. Mach. Intell. 26, 2004 (search Google scholar)
- W. Fan. Graph Pattern Matching Revised for Social Network Analysis. ICDT 2012, March 26–30, 2012, Berlin, Germany. ACM 2012. https://homepages.inf.ed.ac.uk/wenfei/papers/icdt12.pdf
- S. Ma, Y. Cao, W. Fan, J. Huai, T. Wo: Strong simulation: Capturing topology in graph pattern matching. TODS 39(1): 4, 2014.

# Summary and Review

- Query-driven approximation
- What is subgraph isomorphism? Complexity? Algorithm? Name a few applications
- What is graph simulation? Complexity? Understand its algorithm. Name a few applications
- Why do we need to revise conventional graph pattern matching for social network analysis? How should we do it? Why?
- Understand bounded simulation. Read its algorithm. Complexity?
- What is strong simulation? Complexity? Name a few applications in which strong simulation is useful.
- Find other revisions of conventional graph pattern matching that are not covered in the lecture.