Updated 3/3/2023

Topic 2: Complexity of Query Evaluation
Unit 3: Provenance
Lecture 16

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp23)

https://northeastern-datalab.github.io/cs7240/sp23/

3/3/2023

Topic 2: Complexity of Query Evaluation & Reverse Data Management

- **Lecture 13 (Tue 2/21):** T2-U1 Conjunctive Queries
- **Lecture 14 (Fri 2/24):** T2-U1 Conjunctive Queries
- **Lecture 15 (Tue 2/28):** T2-U1/2 Conjunctive & Beyond Conjunctive Queries
- **Lecture 16 (Fri 3/3):** T2-U4 Reverse Data Management

Pointers to relevant concepts & supplementary material:

- **Unit 1. Conjunctive Queries**: Query evaluation of conjunctive queries (CQs), data vs. query complexity, homomorphisms, constraint satisfaction, query containment, query minimization, absorption: [Kolaitis, Vardi'00], [Vardi'00], [Kolaitis'16], [Koutris'19] L1 & L2
- **Unit 2. Beyond Conjunctive Queries**: unions of conjunctive queries, bag semantics, nested queries, tree pattern queries: [Kolaitis'16], [Tan+'14], [G.'11], [Martens'17]
- **Unit 3. Provenance**: [Buneman+02], [Green+07], [Cheney+09], [Green,Tannen'17], [Kepner+16], [Buneman, Tan'18]
- **Unit 4. Reverse Data Management**: update propagation, resilience: [Buneman+02], [Kimelfeld+12], [Freire+15]

Topic 3: Efficient Query Evaluation & Factorized Representations

- Spring break (Tue 3/7, Fri 3/10: Northeast Database day 2023 @ Northeastern)
- **Lecture 17 (Tue 3/14):** T3-U1 Acyclic Queries

Topic 4: Normalization, Information Theory & Axioms for Uncertainty

- **Lecture:** Normal Forms & Information Theory
- **Lecture:** Axioms for Uncertainty

Topic 5: Linear Algebra & Iterative Graph Algorithms

- **Lecture:** Graphs & Linear Algebra
- **Lecture:** Computation Graphs

Outline: T2-3/4: Provenance & Reverse Data Management

- T2-3: Provenance
  - Data Provenance
  - The Semiring Framework for Provenance
  - Algebra: Monoids and Semirings
  - Query-rewrite-insensitive provenance
- T2-4: Reverse Data Management
  - View Deletion Problem
  - Resilience & Causality

~ Explanations

Data provenance.

Imagine a computational process that uses a complex input consisting of multiple items. The granularity and nature of "input item" can vary significantly. It can be a single tuple, a database table, or a whole database. It can a spreadsheet describing an experiment, a laboratory notebook entry, or another form of capturing annotation by humans in software. It can also be a file, or a storage system component. It can be a parameter used by a module in a scientific workflow. It can also be a configuration rule used in software-defined routing or in a complex network protocol. Or it can be a configuration decision made by a distributed computation scheduler (think map-reduce). *Provenance analysis* allows us to understand how these different input items affect the output of the computation. When done appropriately, such

Source: Green, Tannen. "The Semiring Framework for Database Provenance", PODS 2017: https://doi.org/10.1145/3034786.3056125

Near-Term Challenges in II

II = Intelligent Infrastructure

- Error control for multiple decisions
- Systems that create markets
- Designing systems that can provide meaningful, calibrated notions of their uncertainty
- Achieving real-time performance goals
- Managing cloud-edge interactions
- Designing systems that can find abstractions quickly
- Provenance in systems that learn and predict
- Designing systems that can explain their decisions
- Finding causes and performing causal reasoning
- Systems that pursue long-term goals, and actively collect data in service of those goals
- Achieving fairness and diversity
- Robustness in the face of unexpected situations
- Robustness in the face of adversaries
- Sharing data among individuals and organizations
- Protecting privacy and issues of data ownership

Source: Michael I. Jordan: Machine Learning: Dynamical, Stochastic & Economic Perspectives, 2019: https://www.youtube.com/watch?v=-8yYFdV5SOc

Provenance: "Where Did this Data Come from?"

- Whenever data is shared (e.g., science, Web) natural questions appear:
  - How did I get this data?
  - What operations were used to create the data?
  - How much should I trust (believe) it?

- Provenance: describes the origins and history of data in its life cycle
- Two types of provenance
  - Provenance inside a database: that's our focus
  - Provenance outside databases: focus of ongoing research esp. in ML (causes, influence, fairness); less well-defined; there is a standard OPM (Open Provenance Model)
- There are also questions for our focus, provenance inside DBMS:
  - What is the "right data model" of provenance?
  - How do we query it? What operations should we support?

Example of data provenance

- A typical question:
  - For a given database D, a query Q, and a tuple t in the output of $Q(D)$, which parts of D “contribute” to output tuple t?

D

R

Emp	Dept
Alice	D01
Bob	D01
Charly	D02

S

Did	Mgr
D01	Dora
D02	Ema
D03	Felix

Q →

```
SELECT  R.Emp, S.Mgr
FROM    R, S
WHERE   R.Dept=S.Did
```

Q

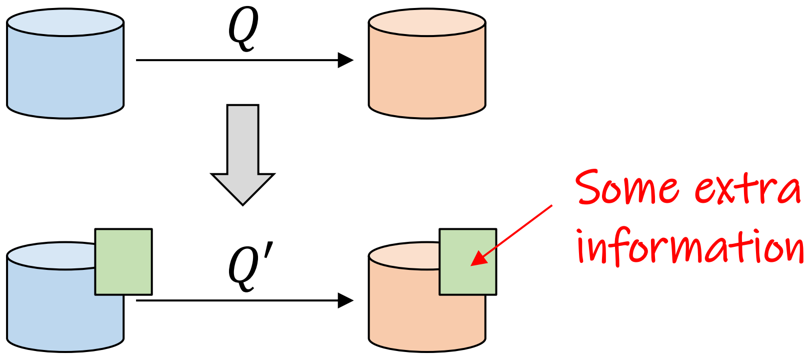
Emp	Mgr
Alice	Dora
Bob	Dora
Charly	Ema

t

  - The question can be applied to attribute values, tables, rows, etc.

Two approaches

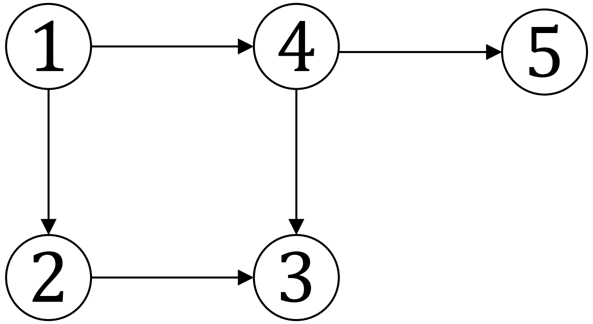
- Eager or annotation-based ("annotation propagation")
  - Changes the transformation from Q to Q' to carry extra information
  - Full source data not needed after transformation



- Lazy or non-annotation based
  - Q is unchanged
  - Recomputation and access to source required.
    - Good when extra storage is an issue.

Conceptual distinction from: Cheney, Chiticariu, Tan. "Provenance in databases -- why, how, and where", 2009. https://doi.org/10.1561/1900000006

Example graph problem, in 5 different variants



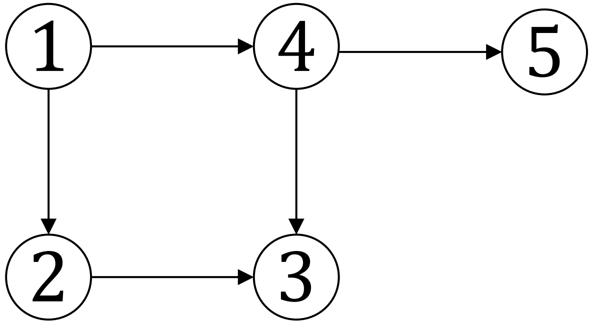
E

from	to
1	2
2	3
1	4
4	3
4	5

Q(z) :- E(1,y), E(y,z) → ?

Q: Points reachable in 2 hops, starting at node "1"

Example graph problem, in 5 different variants



E

from	to
1	2
2	3
1	4
4	3
4	5

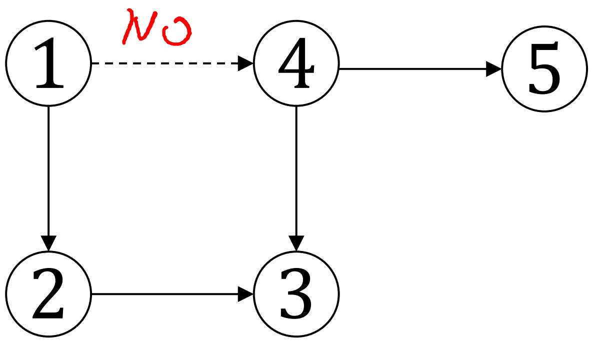
Q(z) :- E(1,y), E(y,z) →

Q

3
5

Q: Points reachable in 2 hops, starting at node "1"

Example variant 1



Now assume only certain edges are available (available yes/no or true/false). Which of the points remain reachable?

E

from	to	
1	2	yes
2	3	yes
1	4	no
4	3	yes
4	5	yes

Q(z) :- E(1,y), E(y,z)

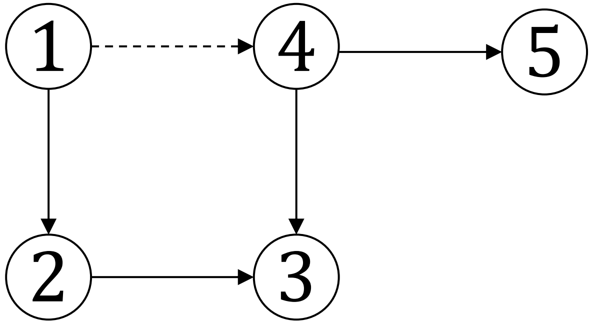
Q: Points reachable in 2 hops, starting at node "1"

Q

3
5

?

Example variant 1



Now assume only certain edges are available (available yes/no or true/false). Which of the points remain reachable?

E

from	to	
1	2	yes
2	3	yes
1	4	no
4	3	yes
4	5	yes

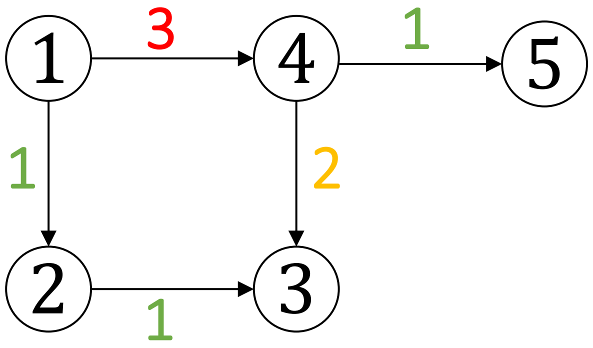
Q(z) :- E(1,y), E(y,z) →

Q

3	yes
5	no

Q: Points reachable in 2 hops, starting at node "1"

Example variant 2



Now assume passing along an edge needs a certain security clearance (1<2<3). What clearance do you need for reaching each point?

E

from	to	
1	2	1
2	3	1
1	4	3
4	3	2
4	5	1

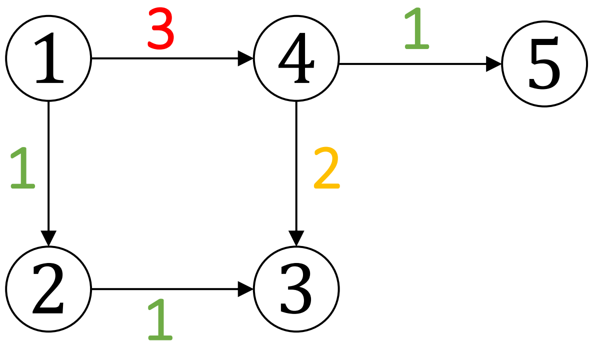
Q(z) :- E(1,y), E(y,z) →

Q
3
5

?

Q: Points reachable in 2 hops, starting at node "1"

Example variant 2



Now assume passing along an edge needs a certain security clearance (1<2<3). What clearance do you need for reaching each point?

E

from	to	
1	2	1
2	3	1
1	4	3
4	3	2
4	5	1

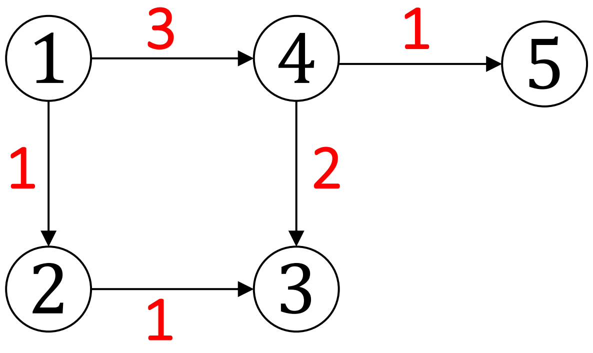
Q(z) :- E(1,y), E(y,z) →

Q

3	1
5	3

Q: Points reachable in 2 hops, starting at node "1"

Example variant 3



Now assume each edge has a weight.
What is the shortest path to reach each point?

E

from	to	
1	2	1
2	3	1
1	4	3
4	3	2
4	5	1

Q(z) :- E(1,y), E(y,z) →

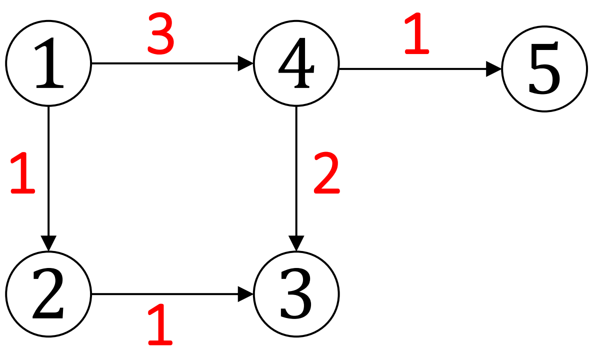
Q

3
5

?

Q: Points reachable in 2 hops, starting at node "1"

Example variant 3



Now assume each edge has a weight.
What is the shortest path to reach each point?

E

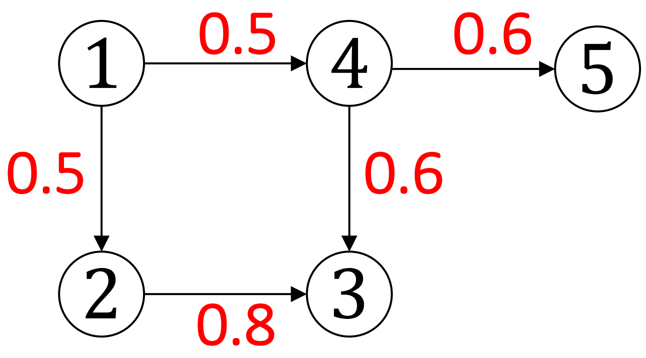
from	to	
1	2	1
2	3	1
1	4	3
4	3	2
4	5	1

Q(z) :- E(1,y), E(y,z) → Q

Q	
3	2
5	4

Q: Points reachable in 2 hops, starting at node "1"

Example variant 4



Now assume each edge has a confidence (probability of being available).
What is the probability of the most likely path?

E

from	to	
1	2	0.5
2	3	0.8
1	4	0.5
4	3	0.6
4	5	0.6

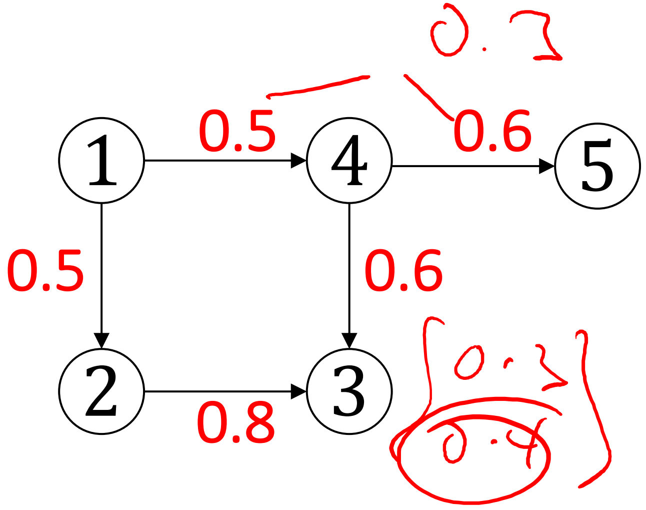
Q(z) :- E(1,y), E(y,z) → Q

Q
3
5

?

Q: Points reachable in 2 hops, starting at node "1"

Example variant 4



Now assume each edge has a confidence (probability of being available).
What is the probability of the most likely path?

E

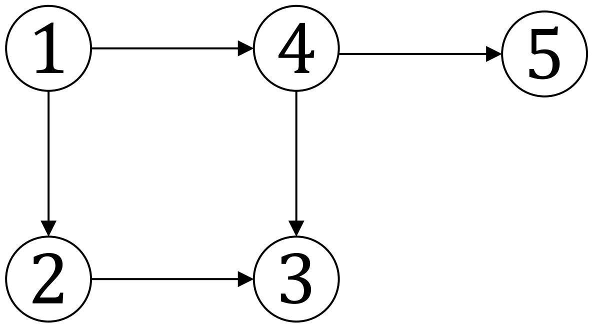
from	to	
1	2	0.5
2	3	0.8
1	4	0.5
4	3	0.6
4	5	0.6

Q(z) :- E(1,y), E(y,z) → Q

Q	
3	0.4
5	0.3

Q: Points reachable in 2 hops, starting at node "1"

Example variant 5



Finally assume we want to calculate the number of paths to a node. How many are there? What is even a reasonable way to calculate that in general?

E

from	to
1	2
2	3
1	4
4	3
4	5

Q(z) :- E(1,y), E(y,z) →

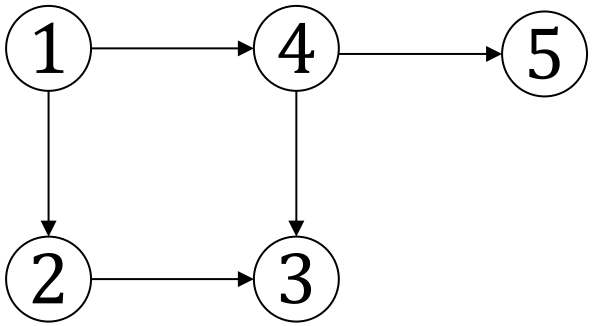
Q

3
5

?

Q: Points reachable in 2 hops, starting at node "1"

Example variant 5



Finally assume we want to calculate the number of paths to a node. How many are there? What is even a reasonable way to calculate that in general?

E

from	to
1	2
2	3
1	4
4	3
4	5

Q(z) :- E(1,y), E(y,z) →

Q

3	2
5	1

Q: Points reachable in 2 hops, starting at node "1"