Advanced topics in Databases

Hellenic Mediterranean University

Prof. Demos Akoumianakis (da@hmu.gr)

Agenda

- ✓ *Thus far*
  - *Relational design based on functional dependency theory*
- The next two weeks are devoted to
  - Complex data and new data types
  - Object-relational perspective on databases
    - … with *Postgres* but there are also other systems (SQL3, Oracle)
    - Focus on (this week)
      - Enum, multivalued, composite data types
      - User-defined data types
      - Examples

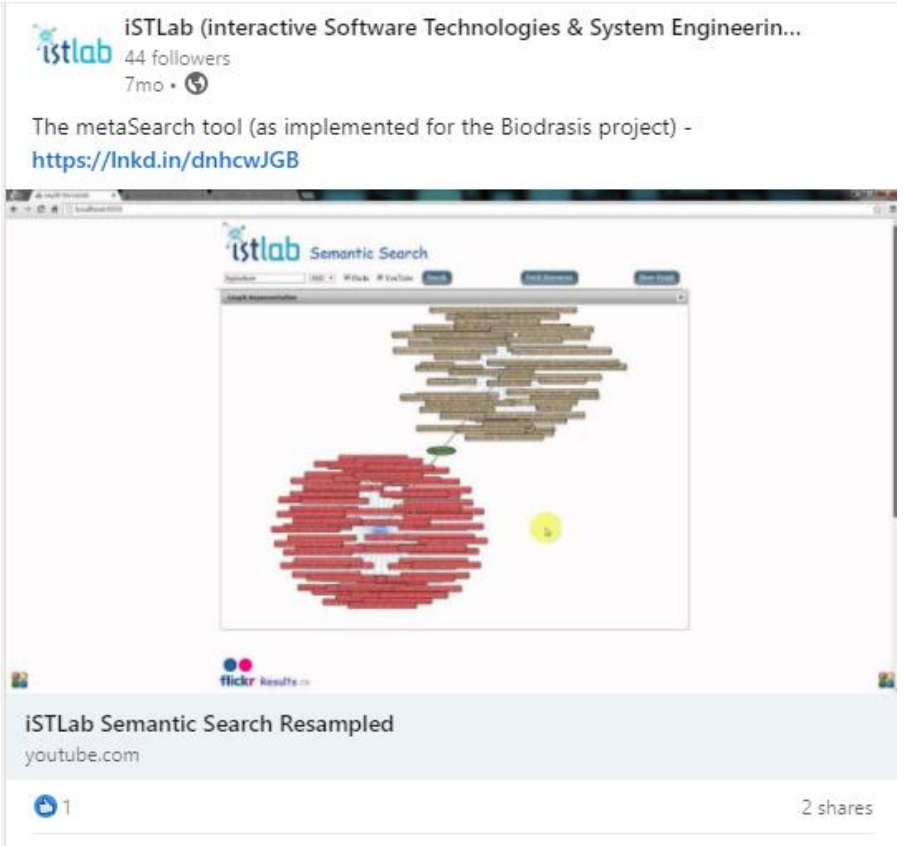
Data categories

- *Structured* data
  - Flat perspective imposed by the relational model with strict constraints

	A	B	C	D	E	F	G
1	Purchase ID	Last name	First name	Birthday	Country	Date of purchase	Amount of purchase
2	1	Davidson	Michael	04/03/1986	United States	10/12/2016	37
3	2	Vito	Jim	09/01/1994	United Kingdom	02/02/2016	85
4	3	Johnson	Tom	23/08/1972	France	02/11/2016	83
5	4	Lewis	Peter	18/10/1979	Germany	22/11/2016	27
6	5	Koenig	Edward	13/05/1983	Argentina	26/03/2015	43
7	6	Preston	Jack	16/06/1991	United States	06/11/2016	77
8	7	Smith	David	11/03/1965	Canada	15/11/2016	23
9	8	Brown	Luis	03/09/1997	Australia	03/07/2015	74
10	9	Miller	Thomas	07/01/1980	Germany	07/11/2016	13
11	10	Williams	Bill	26/07/1960	United States	20/11/2015	80
12	11	Gemini	Alexia	12/09/1995	Canada	11/03/2017	35
13	12	Bond	James	25/02/1975	United Kingdom	12/08/2017	40
14	13	Burgle	Patricia	01/12/1990	United States	18/01/2015	55
15	14	Reding	Michelle	07/04/1985	Canada	23/02/2017	28
16	15	Harvey	Billy	14/07/1971	United Kingdom	12/01/2016	41
17							

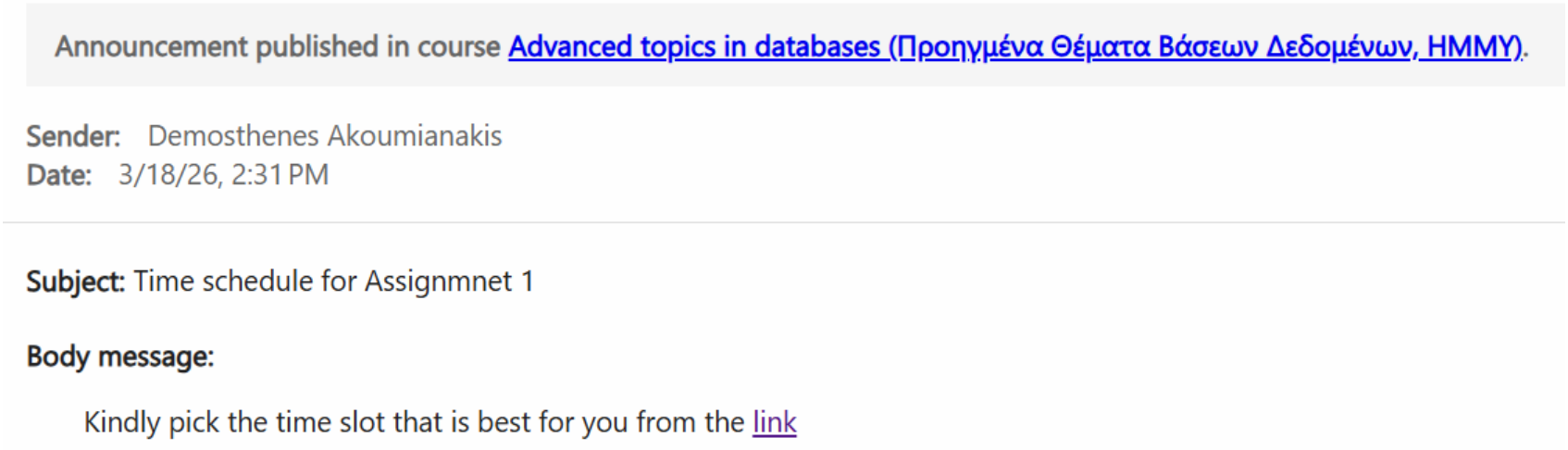
Data categories (cont.)

- *Semi-structured* data
  - No strict structure in a post
  - Various components including
    - ✓ Text
    - ✓ Links
    - ✓ Video or photos
    - ✓ Metrics (shares or hashtags), etc.



Data categories (cont.)

- Email is an example of *semi-structured* data



Announcement published in course Advanced topics in databases (Προηγμένα Θέματα Βάσεων Δεδομένων, HMMY).

Sender: Demosthenes Akoumianakis
Date: 3/18/26, 2:31 PM

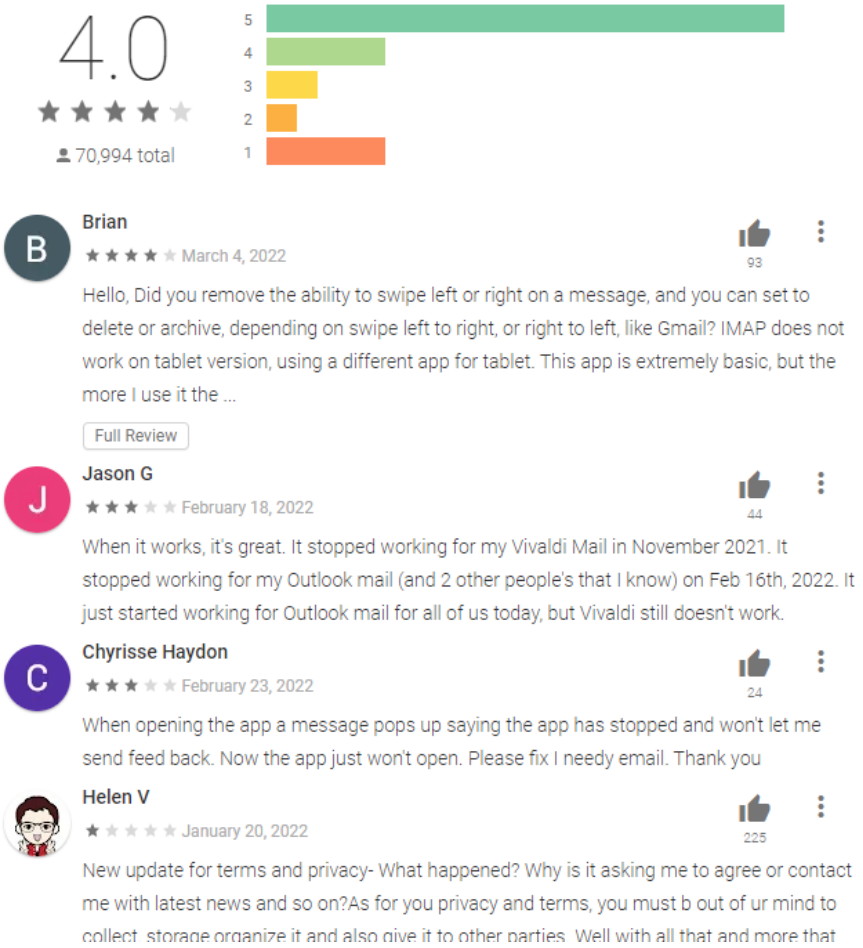
Subject: Time schedule for Assignmnet 1

Body message:

Kindly pick the time slot that is best for you from the link

Data categories (cont.)

- Ratings of an app and the reviews it receives are also examples of *semi-structured* data



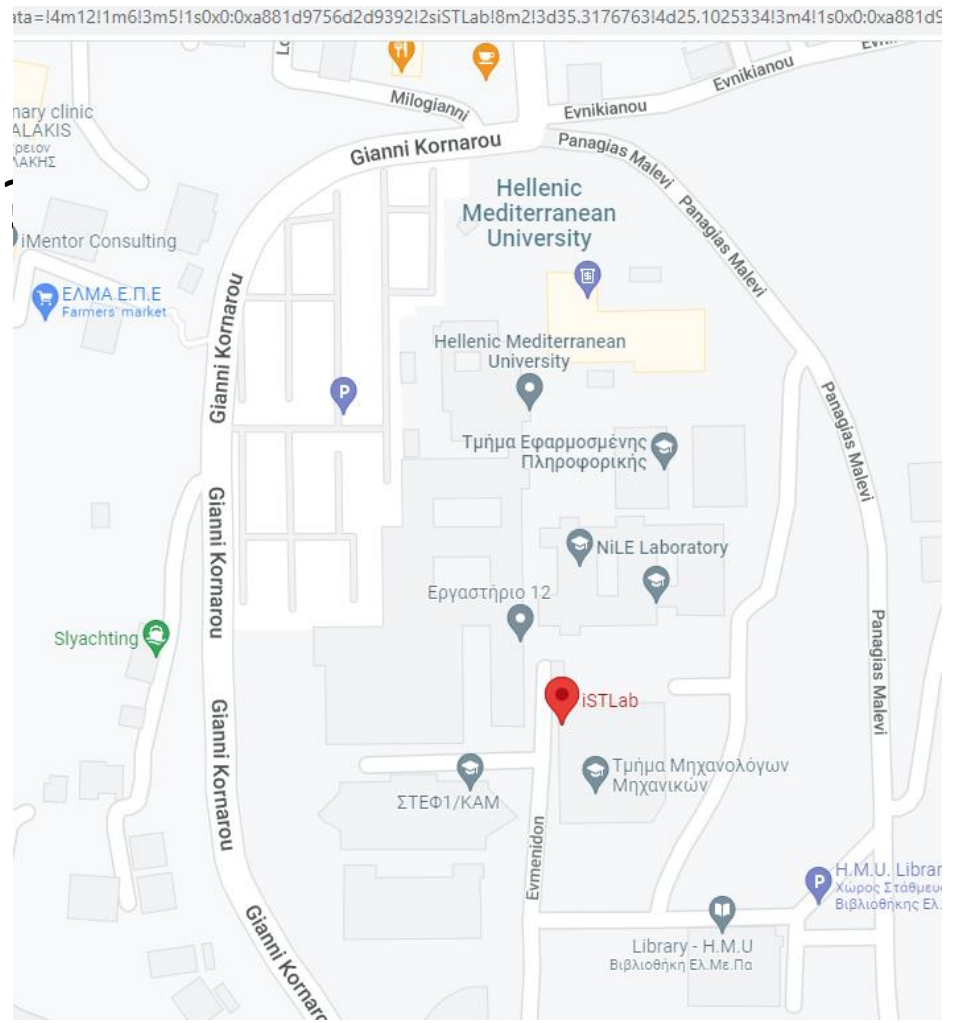
Data categories (cont.)

- Other forms of *unstructured* data such as
  - Spatial data
  - Data on pathways
  - Social data
  - Streaming data
- All the above create new requirements which are often beyond the capabilities of relational technology

Spatial data

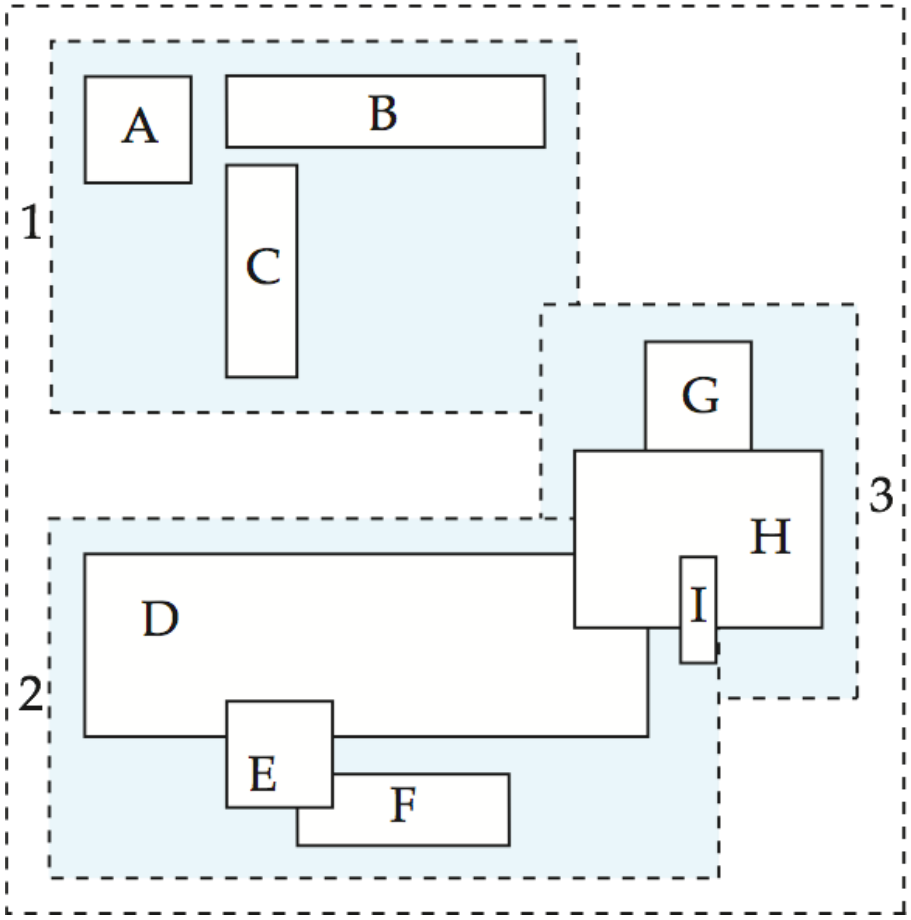
Example of spatial data

- See figure
  - What data are there?
  - Are they characterized?
  - Are they connected?
    - Is there an order?
    - Are there boundaries?
  - What queries are of interest?
  - What processing is required?
    - What is produced as answer?



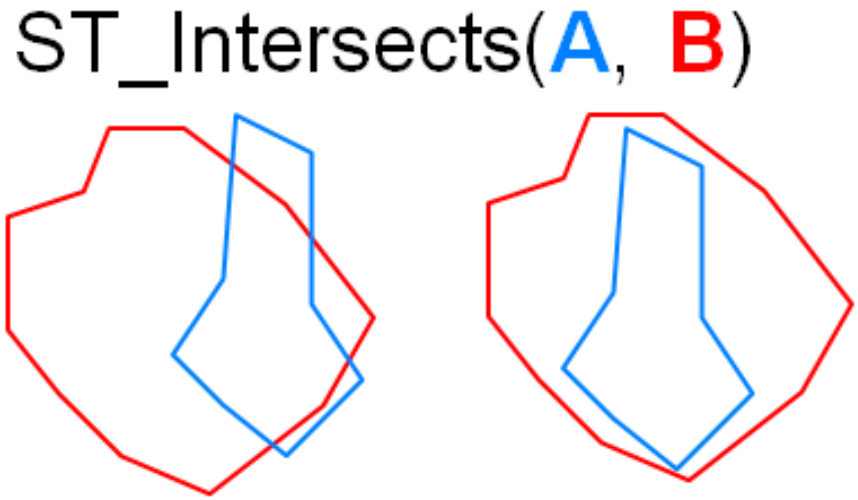
Example of spatial data (cont.)

- See figure
  - What data are there?
  - Are they characterized?
  - Are they connected?
    - Is there an order?
    - Are there boundaries?
  - What *operators*?
    - Containment?
    - Sequence?



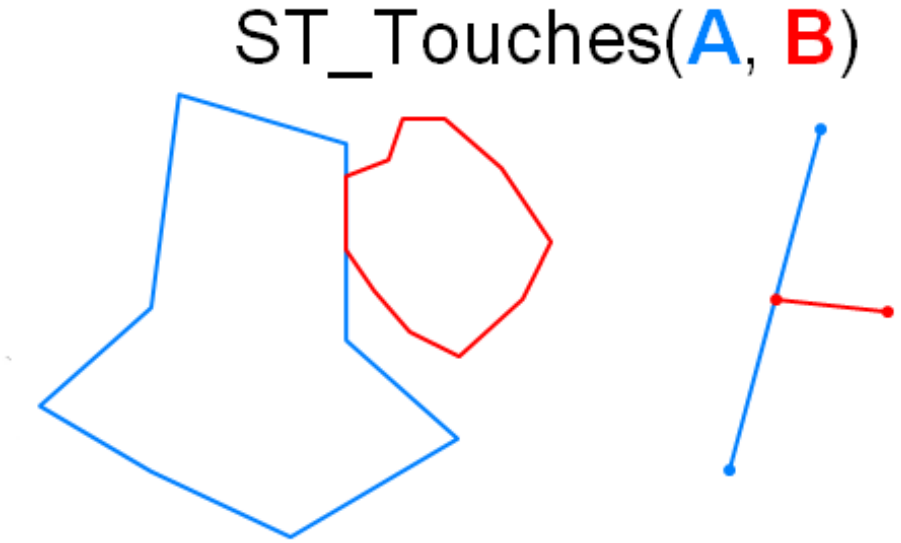
Example of spatial data (cont.)

- Spatial data operators
  - Intersection



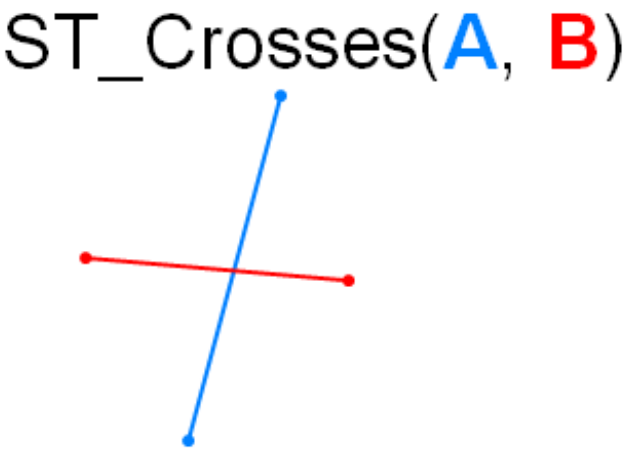
Example of spatial data (cont.)

- Spatial data operators
  - ✓ *Intersection*
  - – Touch



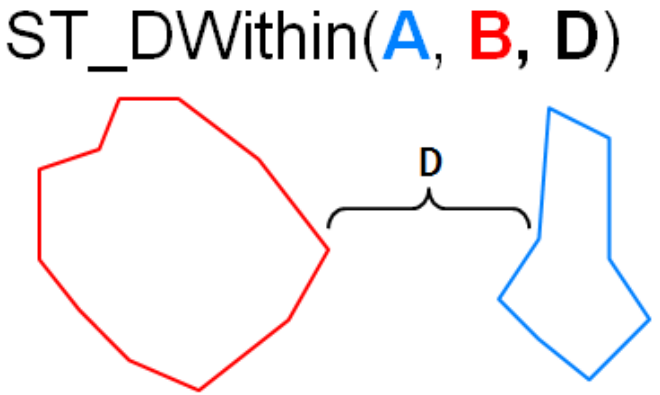
Example of spatial data (cont.)

- Spatial data operators
  - ✓ *Intersection*
  - ✓ *Touch*
  - – Cross



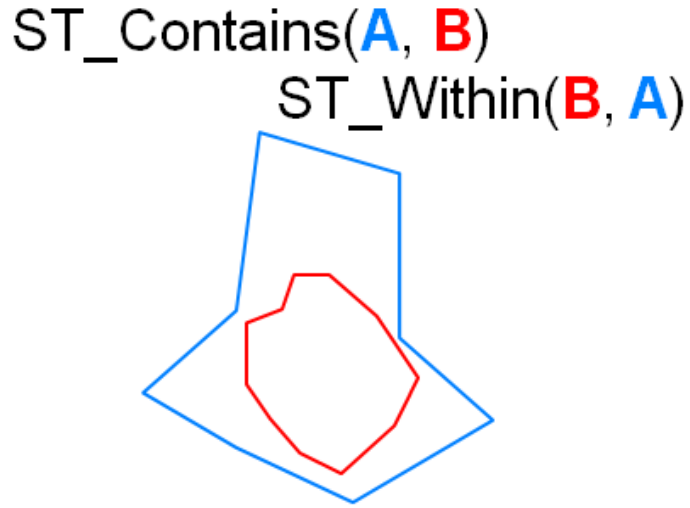
Example of spatial data (cont.)

- Spatial data operators
  - ✓ *Intersection*
  - ✓ *Touch*
  - ✓ *Cross*
  - – Within or distance (between)



Example of spatial data (cont.)

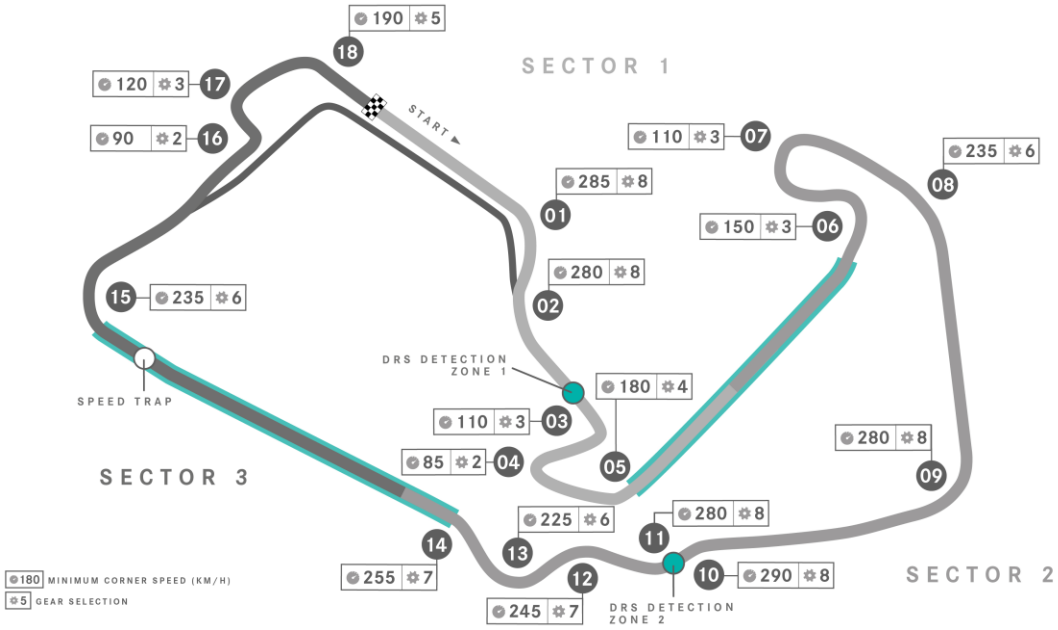
- Spatial data operators
  - ✓ *Intersection*
  - ✓ *Touch*
  - ✓ *Cross*
  - ✓ *Within or distance (between)*
  - – Contain (full/partial)
  - – Overlaps
  - – Relate



Path or trajectory data

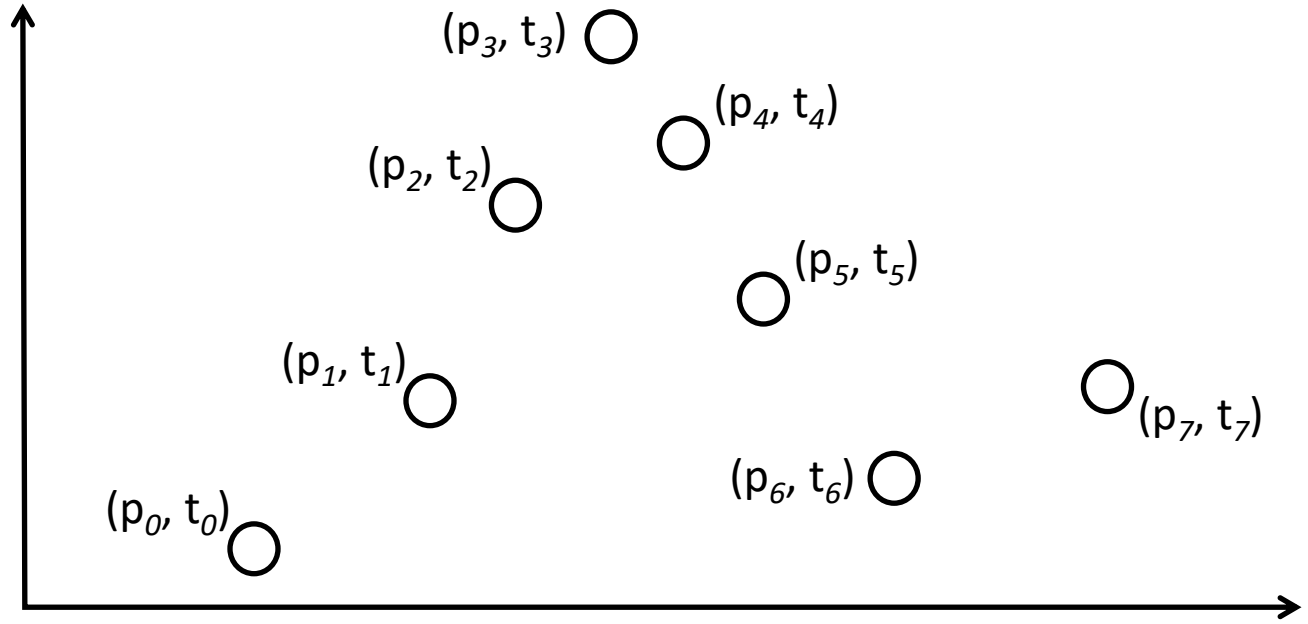
The concept of path or trajectory

- The trajectory of a moving object is the set of *points* or *positions* through which it passes during its movement, or more simply, the set of its successive positions in space as a function of time



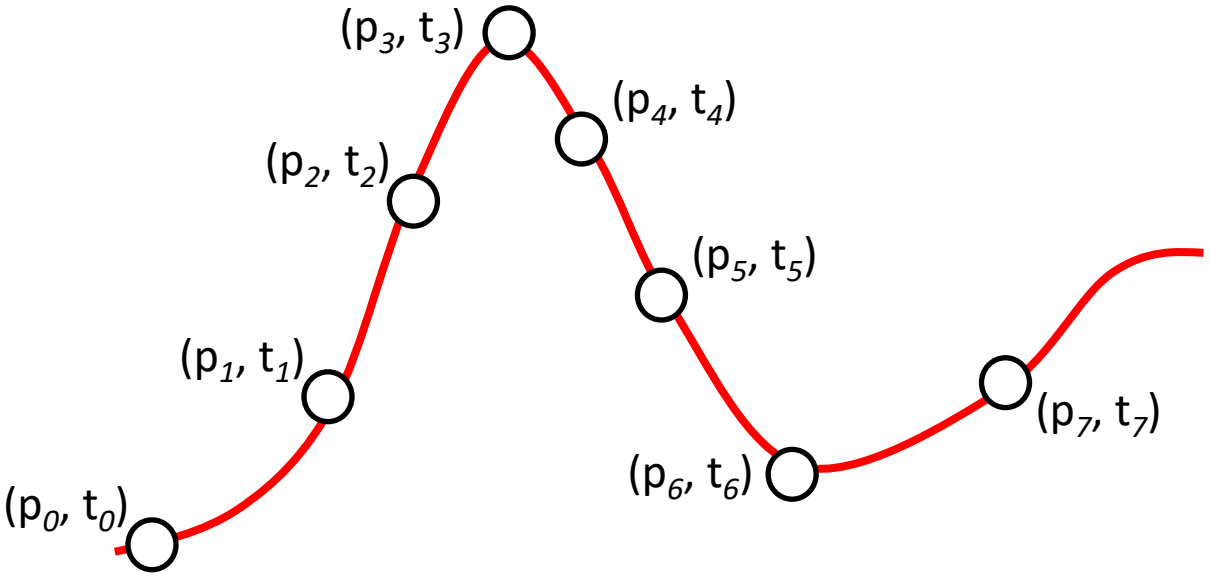
Computationally (interesting) issues

- Given the following points in a cartesian plane (as a function of time)
- What is the path/trajectory?



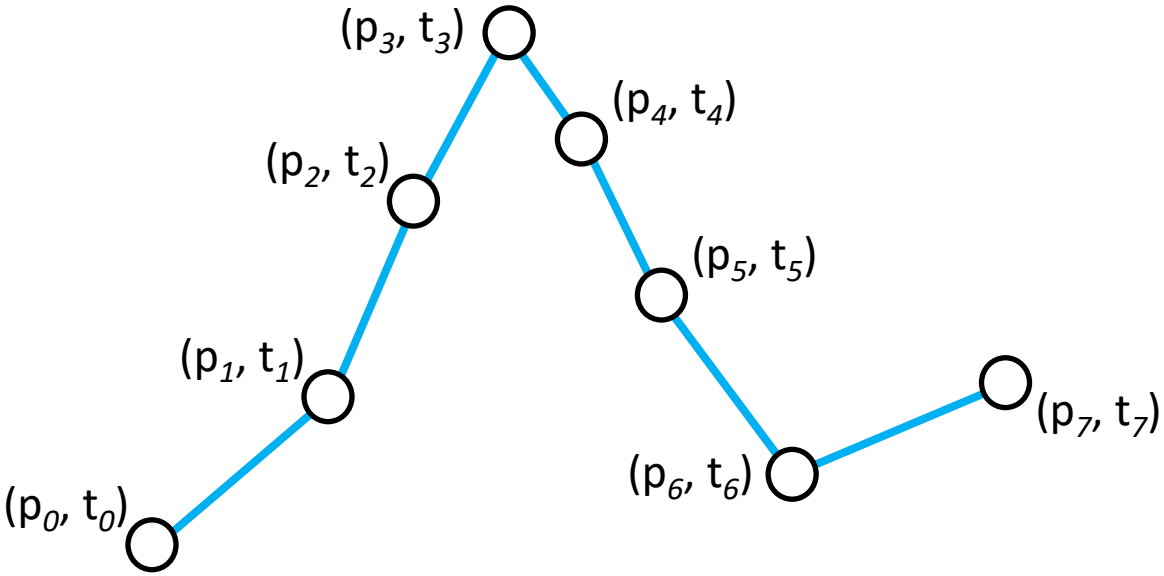
Computationally (interesting) issues (cont.)

- It could be the following



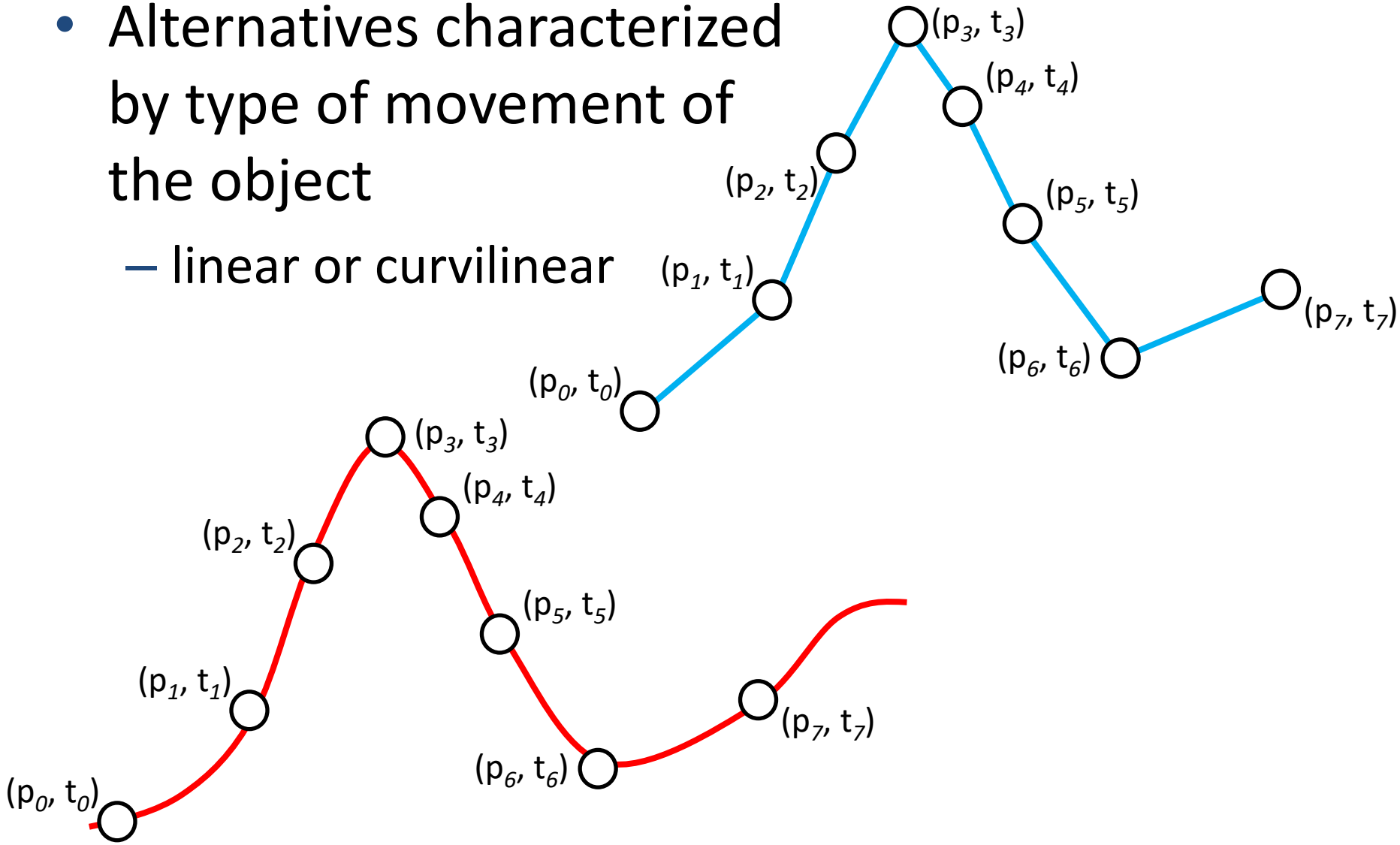
Computationally (interesting) issues (cont.)

- … or alternatively something like this



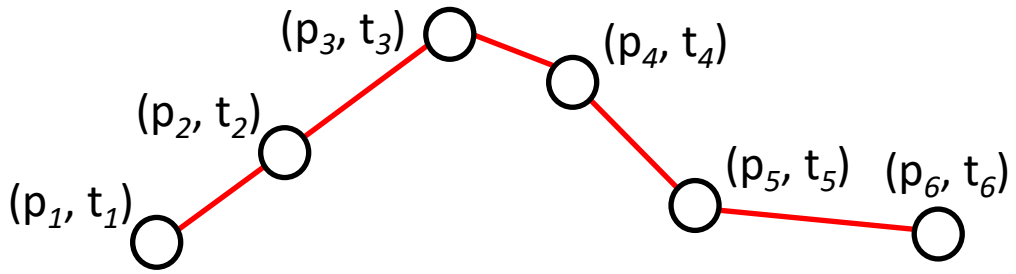
Computationally (interesting) issues (cont.)

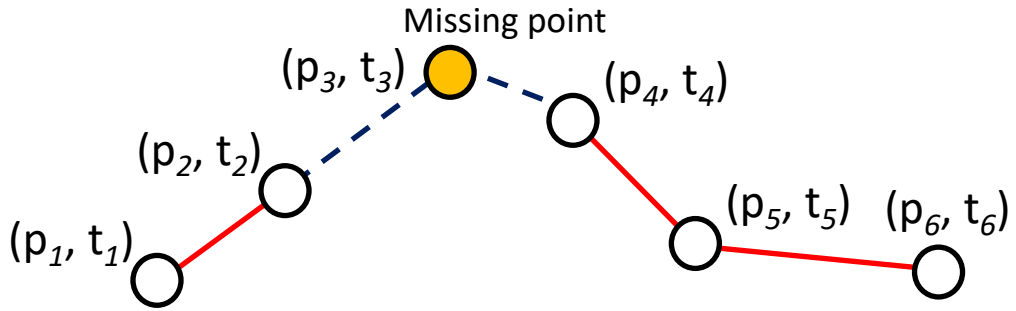
- Alternatives characterized by type of movement of the object
  - linear or curvilinear



Computationally (interesting) issues (cont.)

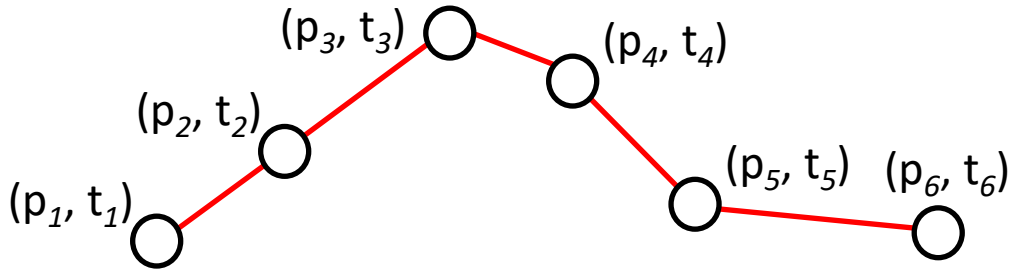
- In the example it would be of interest to compute
  - *Missing points* in a set of trajectory data
    - E.g. How many and which points are 'missing';
      - The answer is one, with coordinates (p_3, t_3)

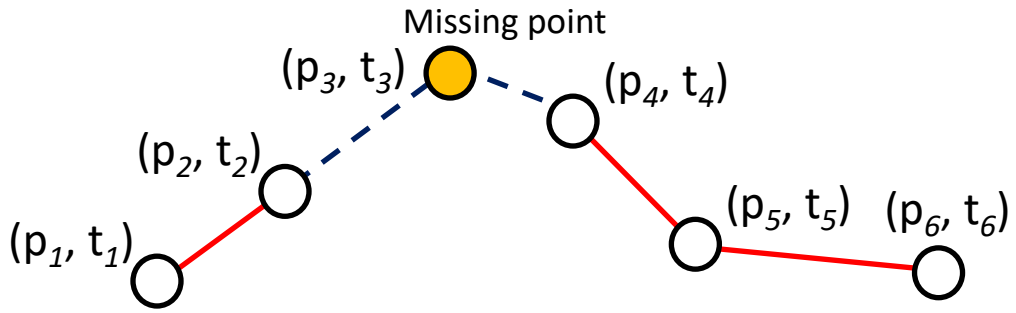




Computationally (interesting) issues (cont.)

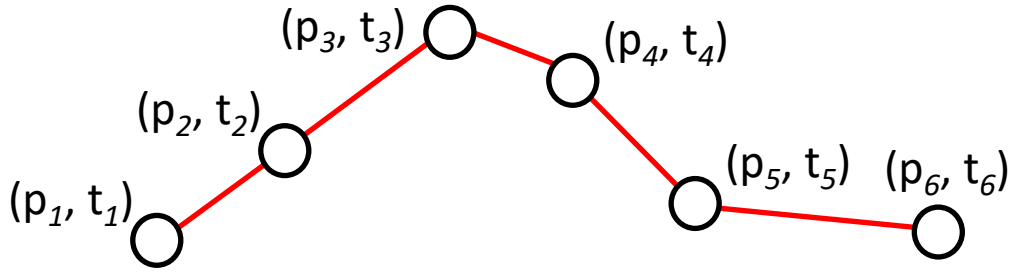
- In the example it would be of interest to compute
  - *Missing points* in a set of trajectory data
    - E.g. How many and which points are 'missing';
      - The answer is one, with coordinates (p_3, t_3)





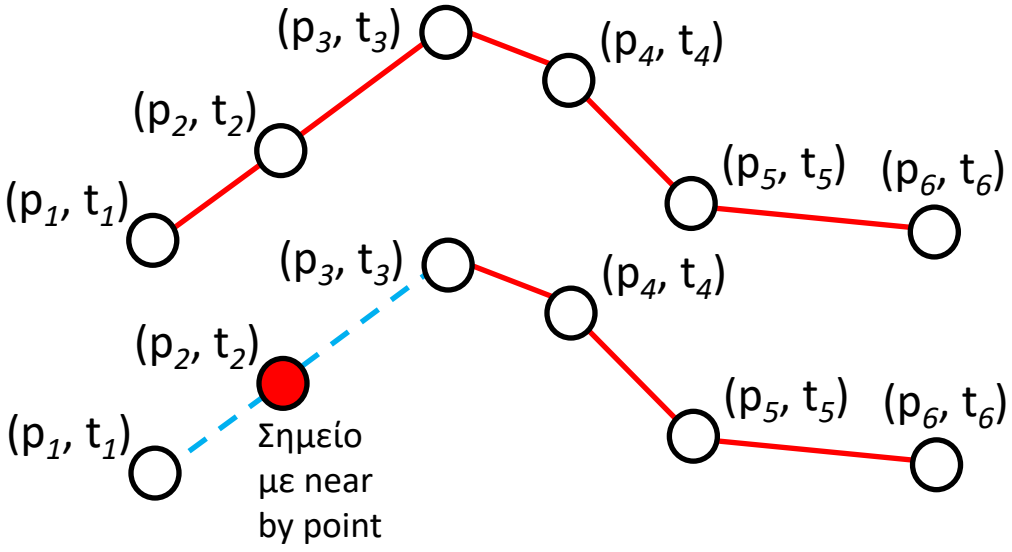
Computationally (interesting) issues (cont.)

- In the example it would be of interest to compute
  - *Near by points* (to a point) in a trajectory
    - e.g. If for a certain point in a trajectory there are two different points (left and right) in the same trajectory



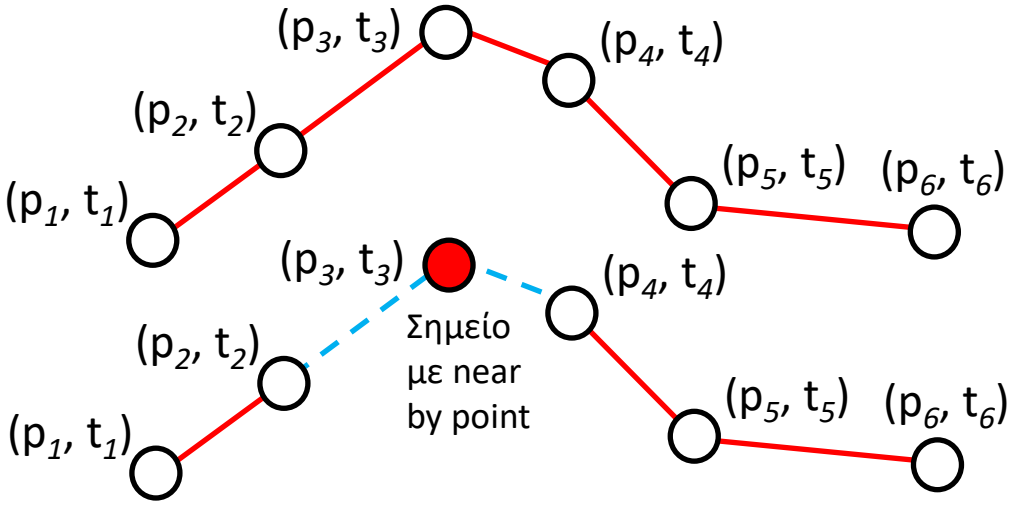
Computationally (interesting) issues (cont.)

- In the example it would be of interest to compute
  - *Near by points* (to a point) in a trajectory
    - e.g. If for a certain point in a trajectory there are two different points (left and right) in the same trajectory



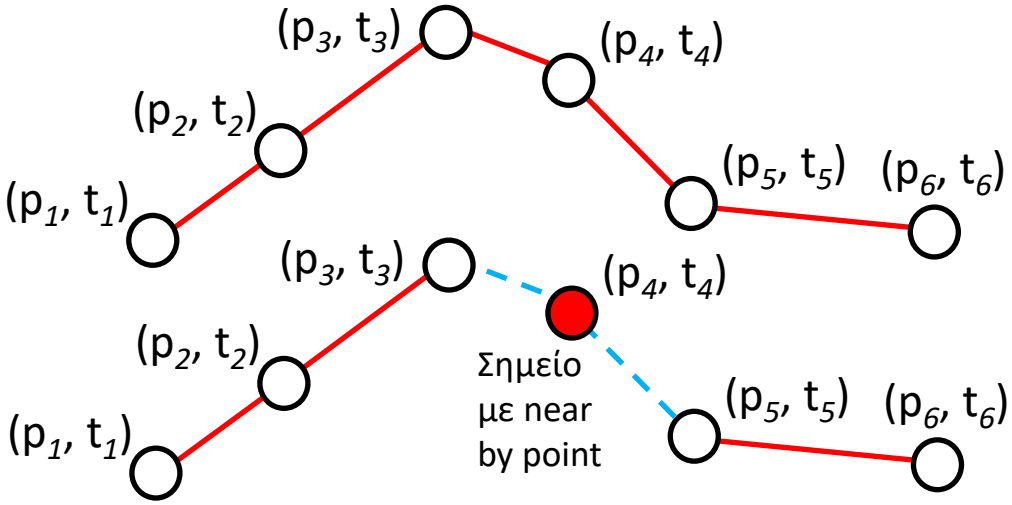
Computationally (interesting) issues (cont.)

- In the example it would be of interest to compute
  - *Near by points* (to a point) in a trajectory
    - e.g. If for a certain point in a trajectory there are two different points (left and right) in the same trajectory



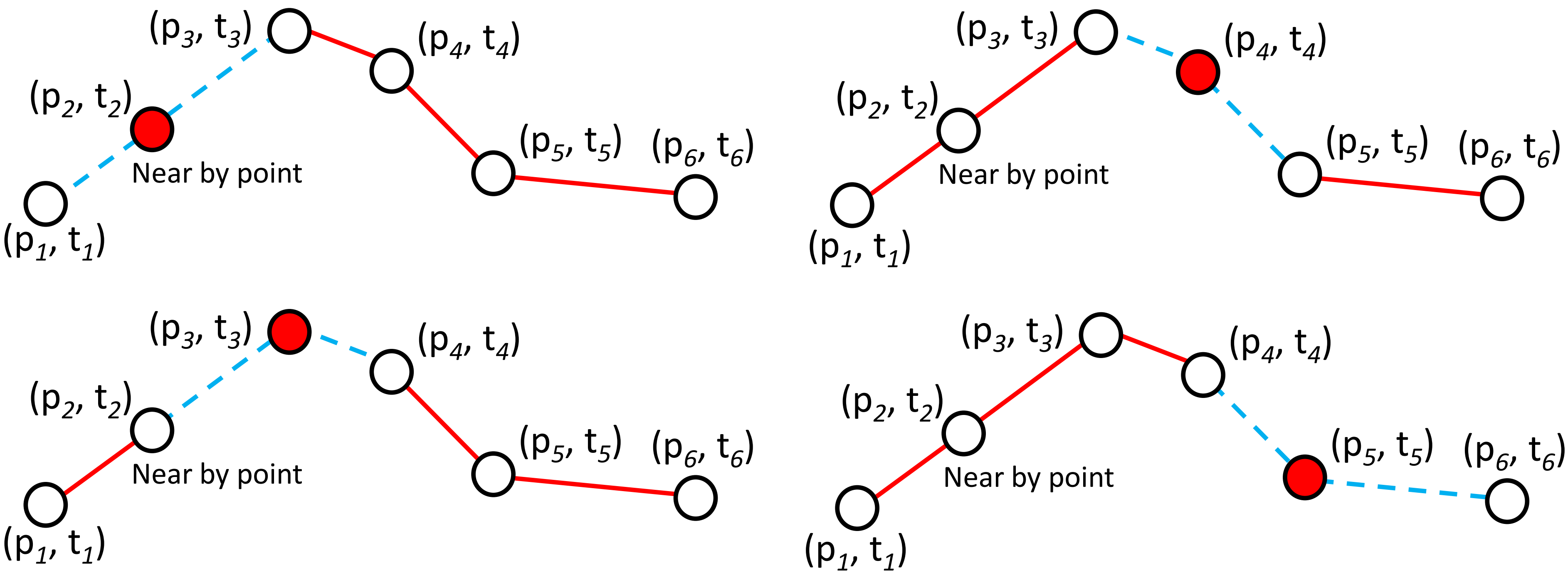
Computationally (interesting) issues (cont.)

- In the example it would be of interest to compute
  - *Near by points* (to a point) in a trajectory
    - e.g. If for a certain point in a trajectory there are two different points (left and right) in the same trajectory



Computationally (interesting) issues (cont.)

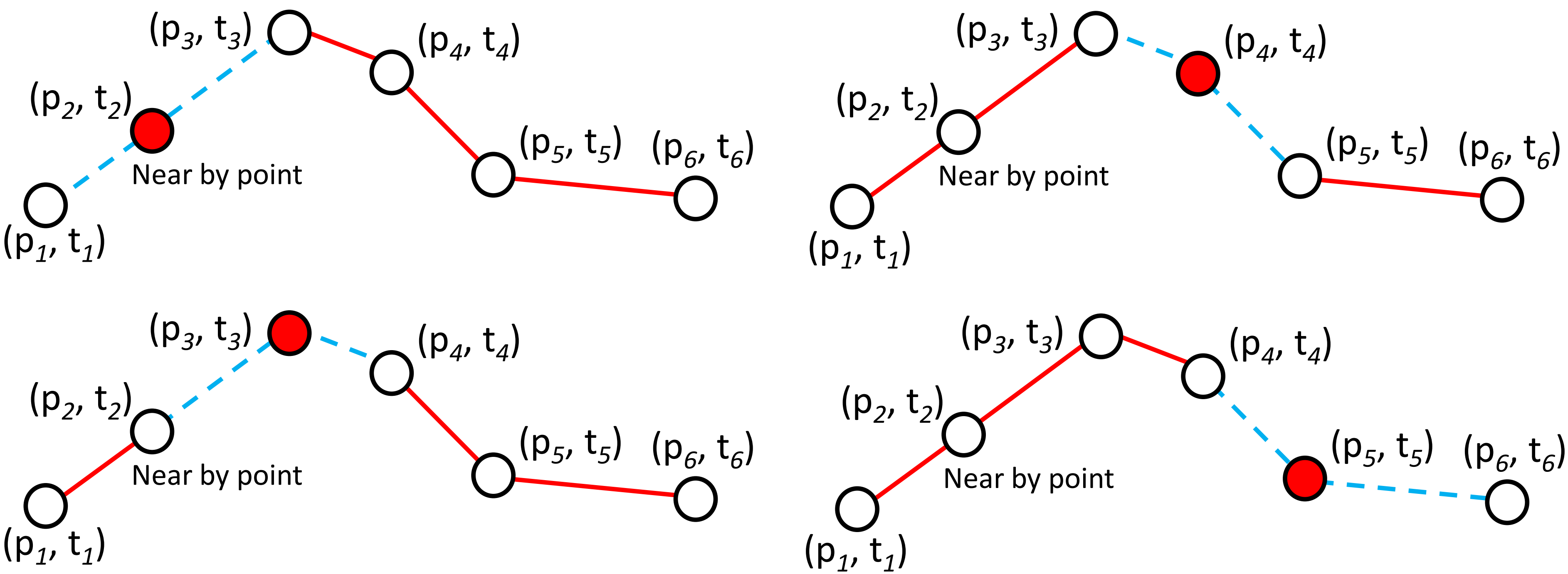
- Consequently, for a query 'How many points possess 'near by points'



- The answer is 4

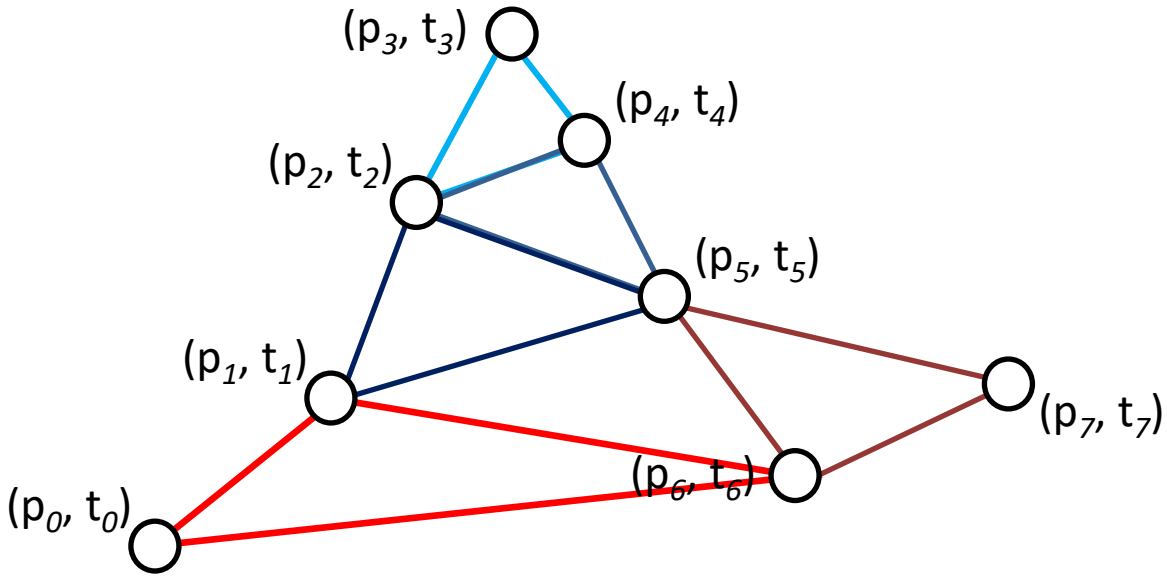
Computationally (interesting) issues (cont.)

- Notice
  - Computation of 'near by points' requires *recursive* processing of each point



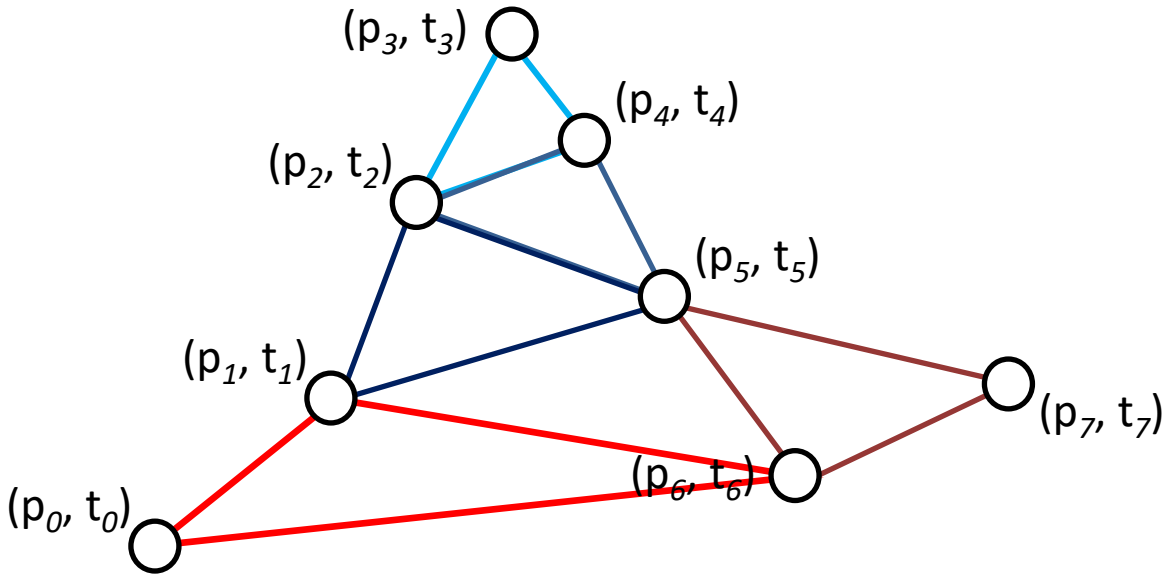
Question

- However, the points may be connected as illustrated below



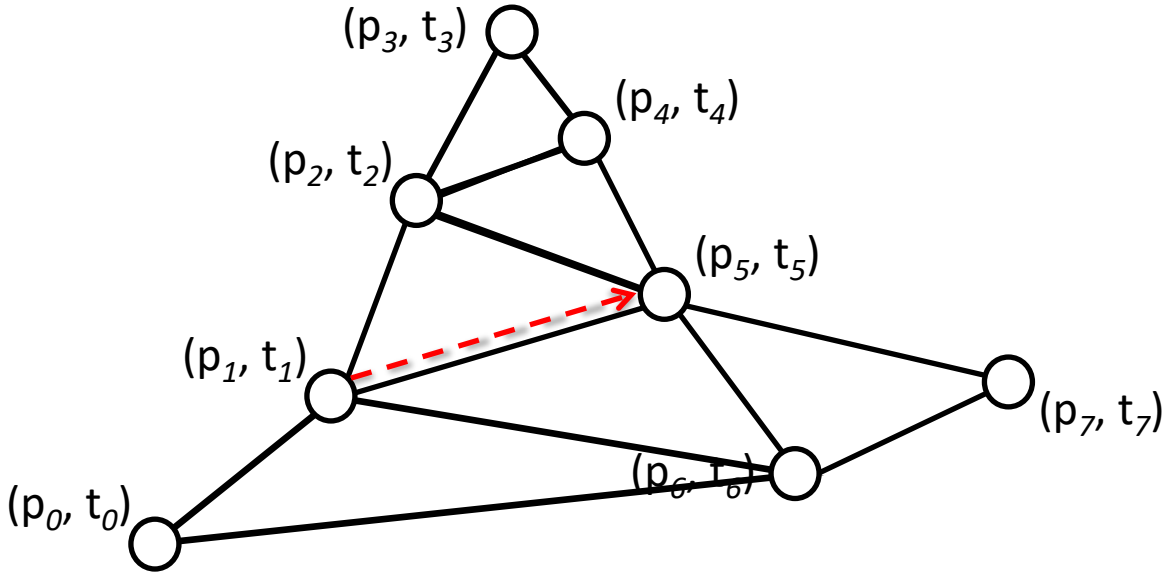
Question (cont.)

- … thus, making possible the computation of all pathways from a certain point to any other
  - e.g. Starting from (p_1, t_1) and arriving to (p_5, t_5)



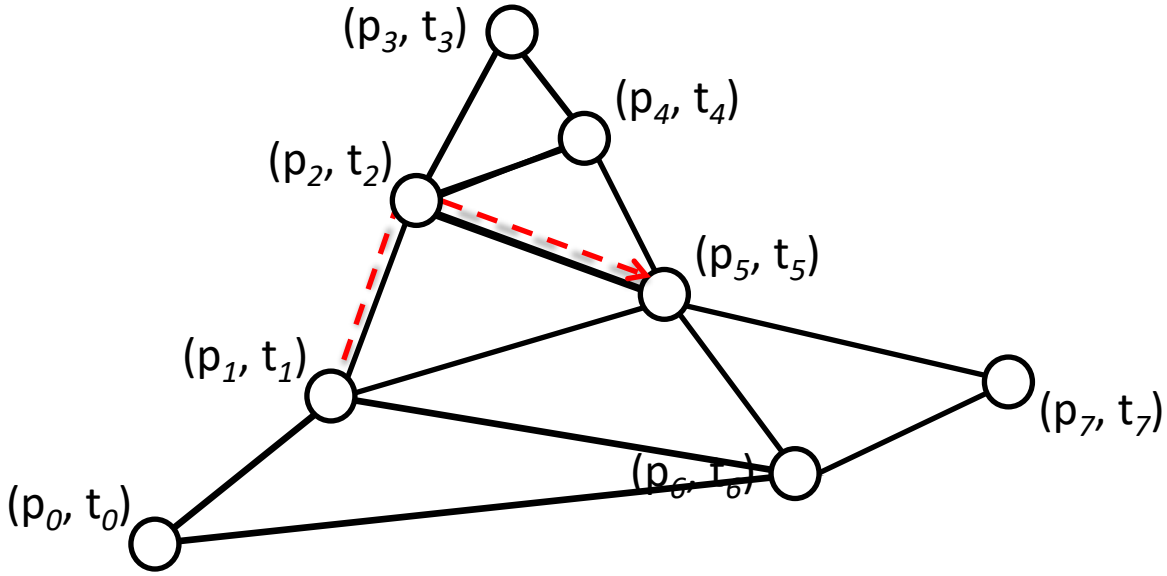
Question (cont.)

- … thus, making possible the computation of all pathways from a certain point to any other
  - e.g. Starting from (p_1, t_1) and arriving to (p_5, t_5)
    - $(p_1, t_1) \rightarrow (p_5, t_5)$



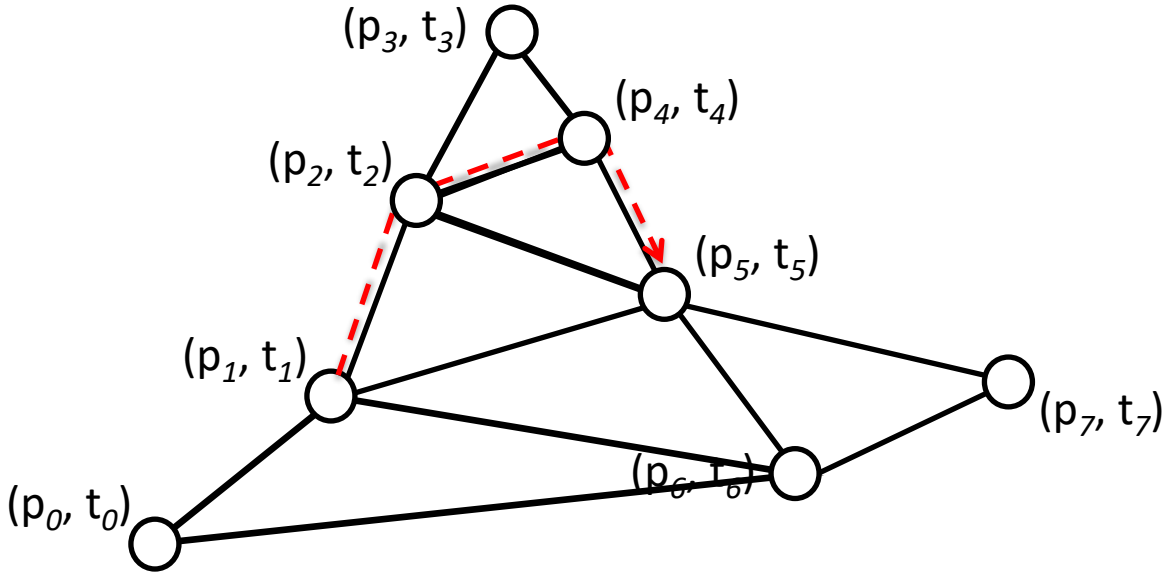
Question (cont.)

- … thus, making possible the computation of all pathways from a certain point to any other
  - e.g. Starting from (p_1, t_1) and arriving to (p_5, t_5)
    - $(p_1, t_1) \rightarrow (p_2, t_2) \rightarrow (p_5, t_5)$



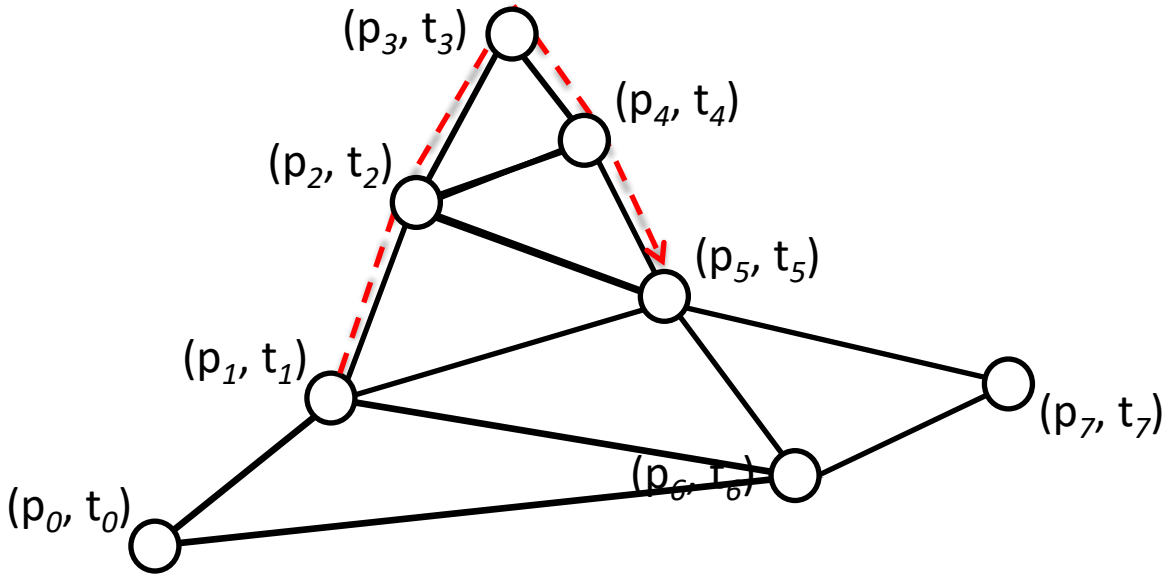
Question (cont.)

- … thus, making possible the computation of all pathways from a certain point to any other
  - e.g. Starting from (p_1, t_1) and arriving to (p_5, t_5)
    - $(p_1, t_1) \rightarrow (p_2, t_2) \rightarrow (p_4, t_4) \rightarrow (p_5, t_5)$



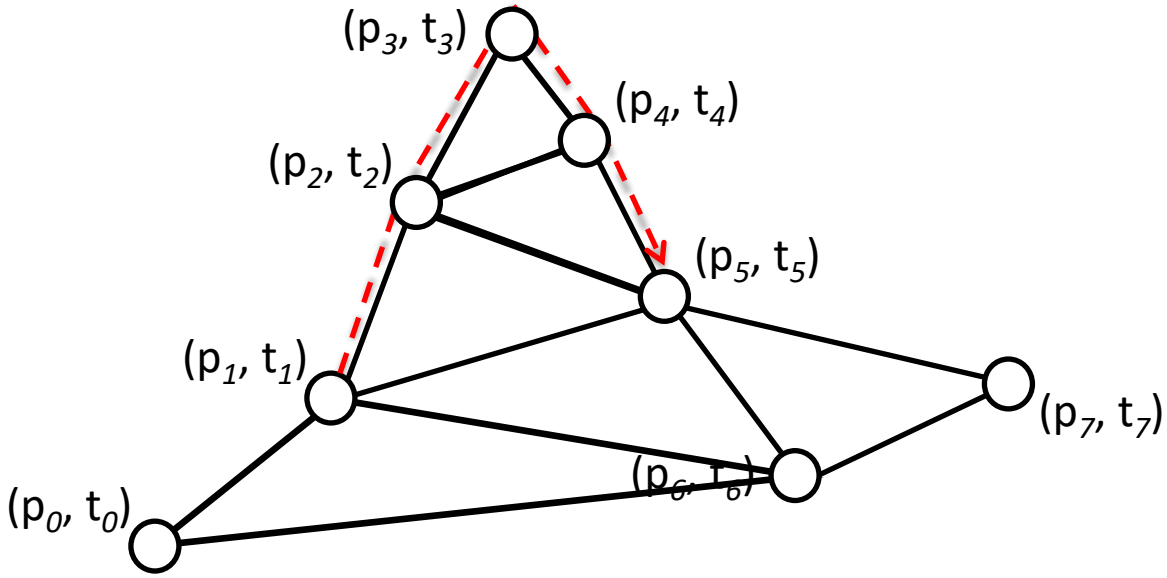
Question (cont.)

- … thus, making possible the computation of all pathways from a certain point to any other
  - e.g. Starting from (p_1, t_1) and arriving to (p_5, t_5)
    - $(p_1, t_1) \rightarrow (p_2, t_2) \rightarrow (p_3, t_3) \rightarrow (p_4, t_4) \rightarrow (p_5, t_5)$



Remark

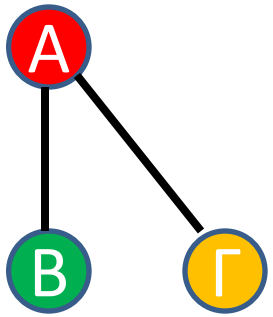
- It is worth noticing that
  - Computing all possible pathways from a certain point to another requires *recursive* processing



Social data

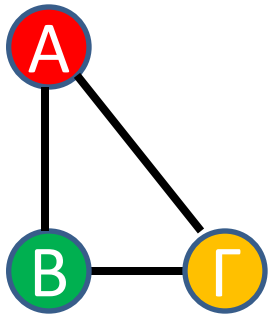
Definition

- Specialized category of data
  - They have come to the surface with the rise of social networking sites
- Example (three users)
  - User A is friend of B and Γ



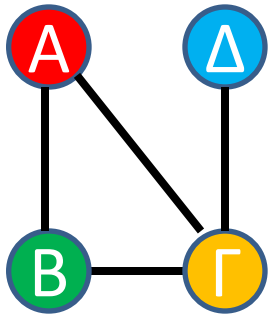
Definition (Cont.)

- Specialized category of data
  - They have come to the surface with the rise of social networking sites
- Example (three users)
  - User A is friend of B and Γ
  - User B is friend of Γ



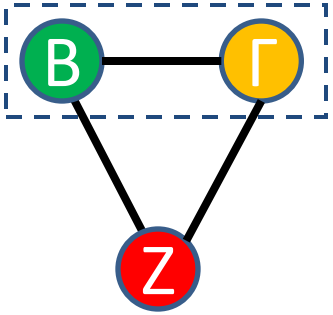
Definition (Cont.)

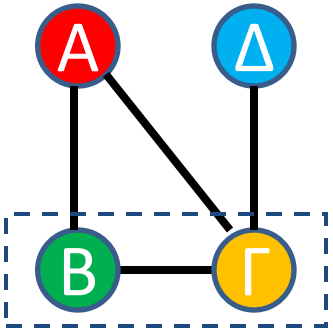
- Specialized category of data
  - They have come to the surface with the rise of social networking sites
- Example (three users)
  - User A is friend of B and Γ
  - User B is friend of Γ
  - User Γ is friend of Δ



Definition (Cont.)

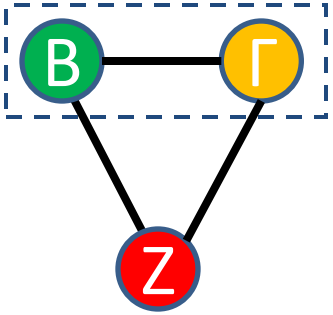
- If another user Z is friend of B and Γ
- What can we make out of this?
  - There are common friends

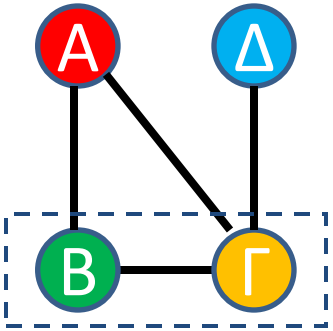




Definition (Cont.)

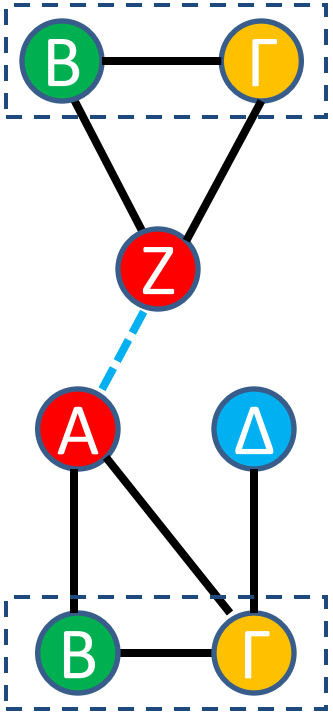
- If another user Z is friend of B and Γ
- What can we make out of this?
  - There are common friends
  - There can be recommendations





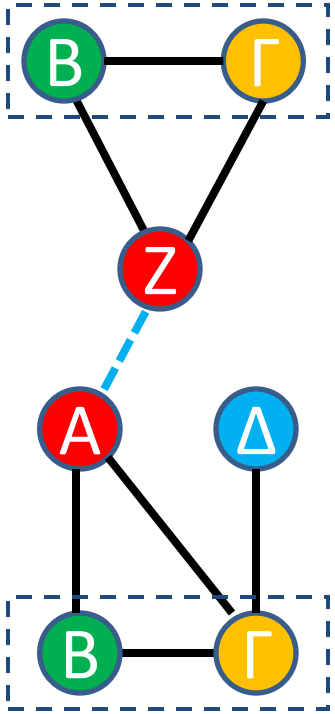
Definition (Cont.)

- If another user Z is friend of B and Γ
- What can we make out of this?
  - There are common friends
  - There can be recommendations
    - Z to be connected with A



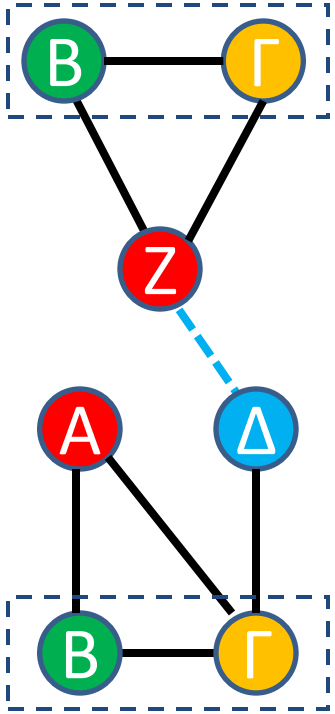
Definition (Cont.)

- If another user Z is friend of B and Γ
- What can we make out of this?
  - There are common friends
  - There can be recommendations
    - Z to be connected with A
    - A to be connected with Z



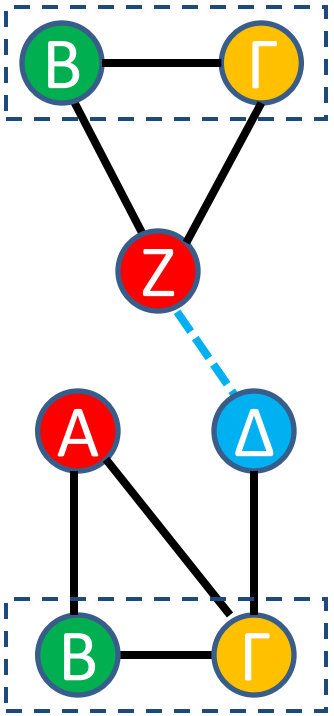
Definition (Cont.)

- If another user Z is friend of B and Γ
- What can we make out of this?
  - There are common friends
  - There can be recommendations
    - Z to be connected with A
    - A to be connected with Z
    - Z to be connected with Δ



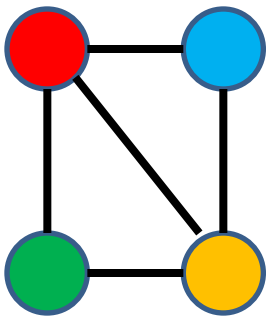
Conclusion

- A pre-requisite
  - Given a set of users we need to be able to compute the overlap between the networks of friends
- One definition if overlap
  - For any two users A and Z, there is overlap in their networks S_A and S_Z if $S_Z \cap S_A \neq \varnothing$ holds
- Moreover,
  - If $S_Z \subseteq S_A$ then $S_Z < S_A$
  - Transitive property does not apply
    - ✓ i.e., if A is friend if Γ and Γ is friend of Δ, then it does not hold that A is friend of Δ

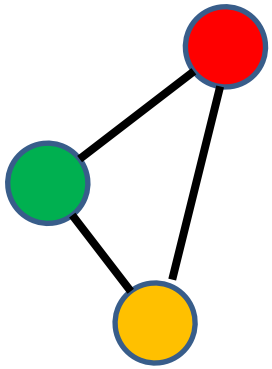


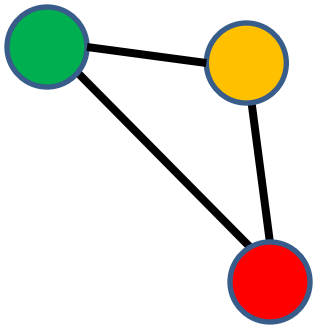
Additional useful operators

- Considering the following
  - Is there any kind of relationship between the vertices?

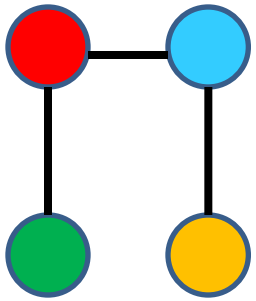


G_1





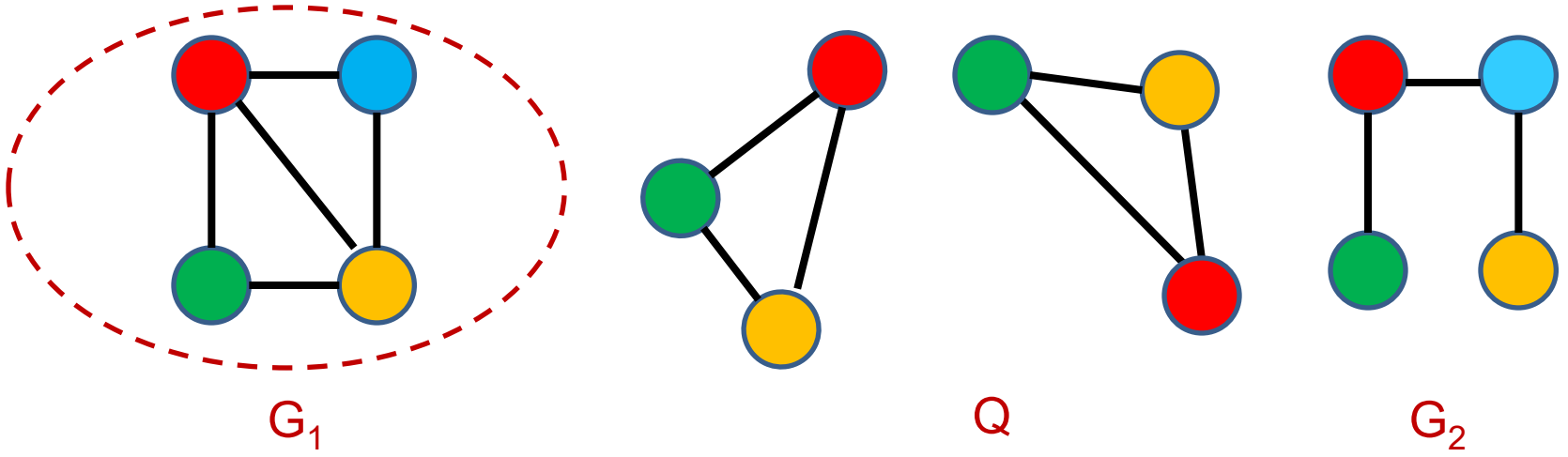
Q



G_2

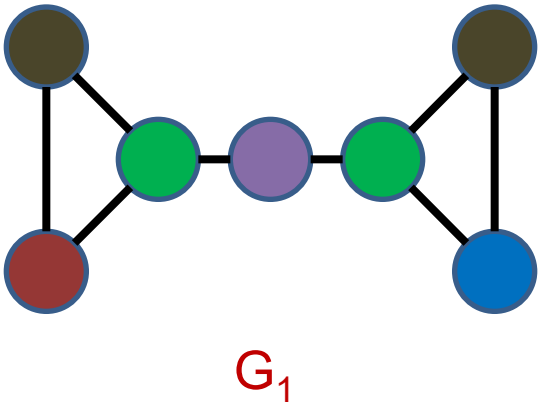
Additional useful operators (cont.)

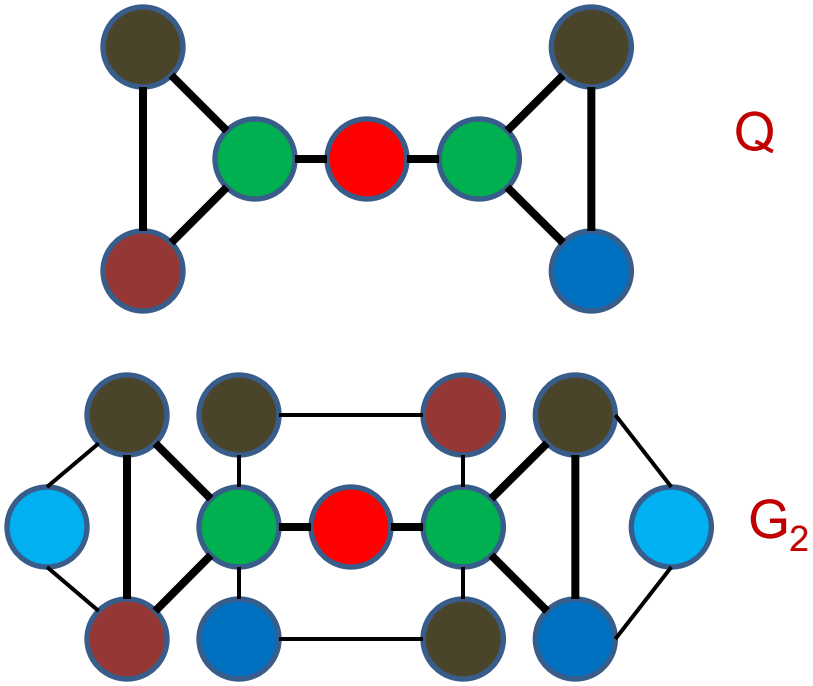
- Containment
  - Is there any kind of relationship between the vertices?
    - G_1 *CONTAINS* Q



Additional useful operators (cont.)

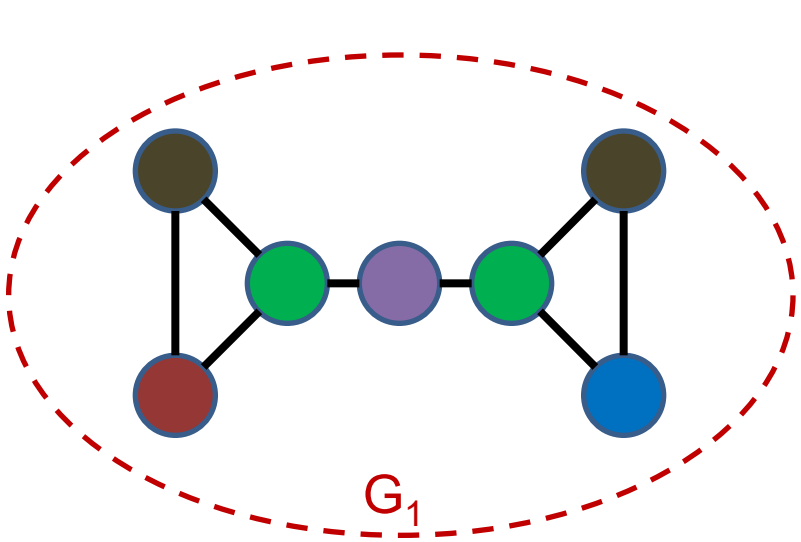
- Similarity
  - Is there any kind of relationship between the vertices?

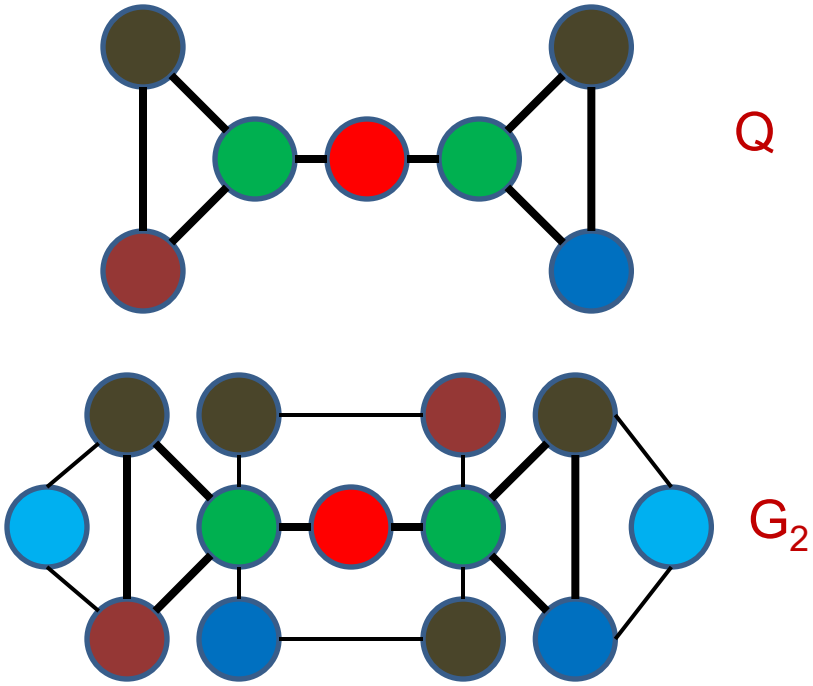




Additional useful operators (cont.)

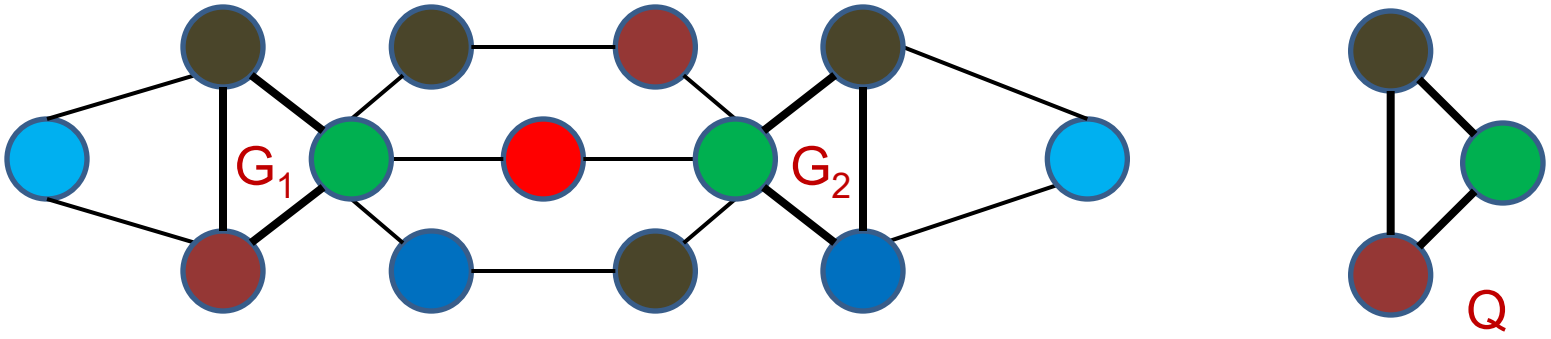
- Similarity
  - Is there any kind of relationship between the vertices?
    - G_1 *SIMILAR TO* Q





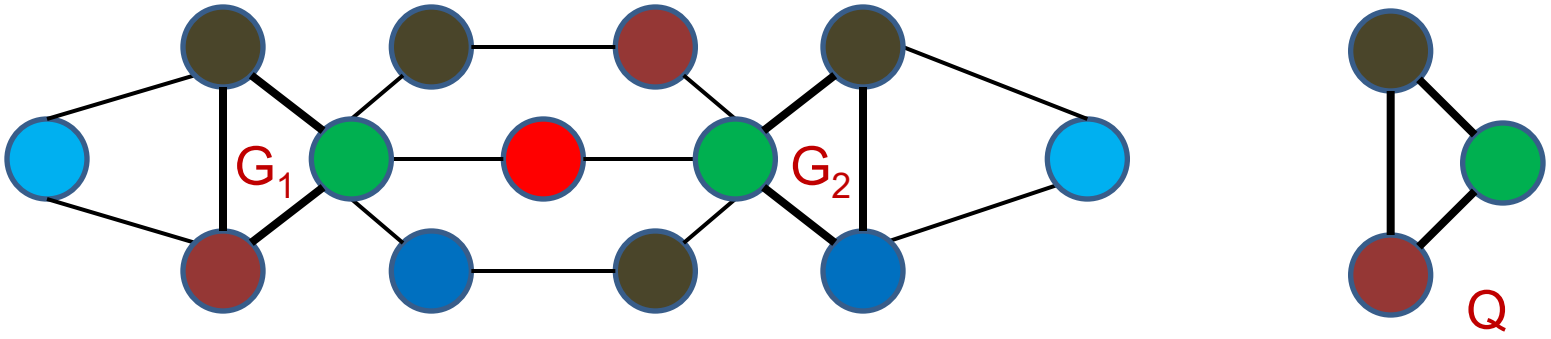
Additional useful operators (cont.)

- Matching
  - Is there any kind of relationship between the vertices?



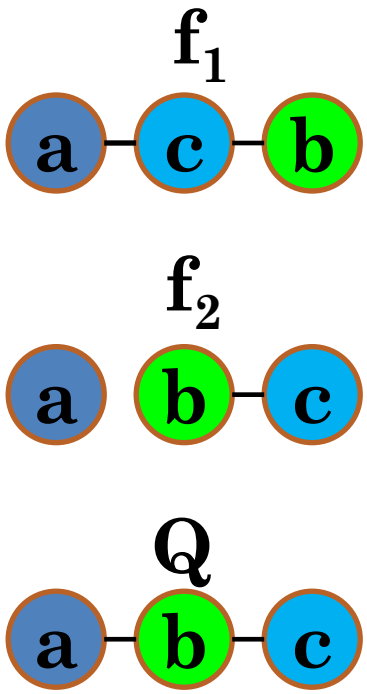
Additional useful operators (cont.)

- Matching
  - Is there any kind of relationship between the vertices?
    - Q *MATCHES* <G_1, G_2>



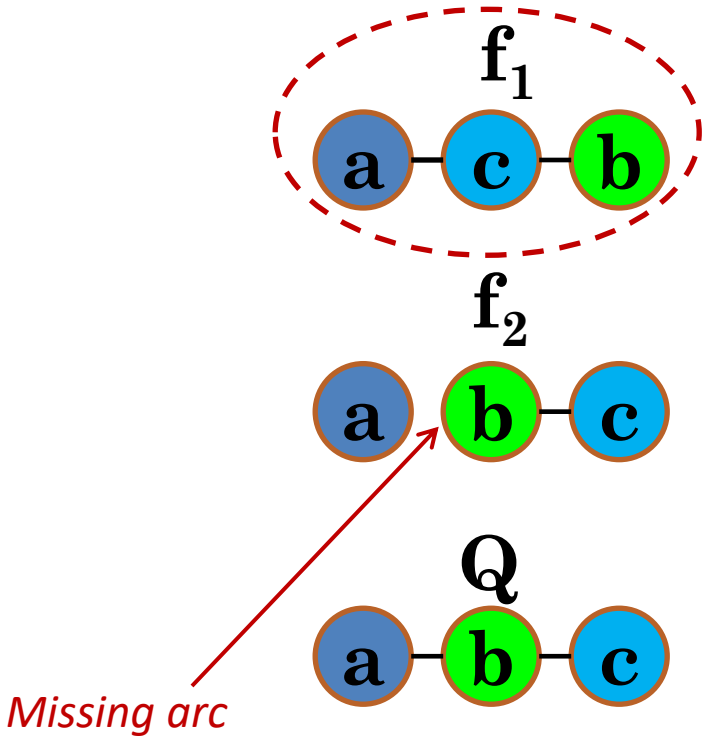
Additional useful operators (cont.)

- Comparison
  - Ποιο είναι περισσότερο κοντά;



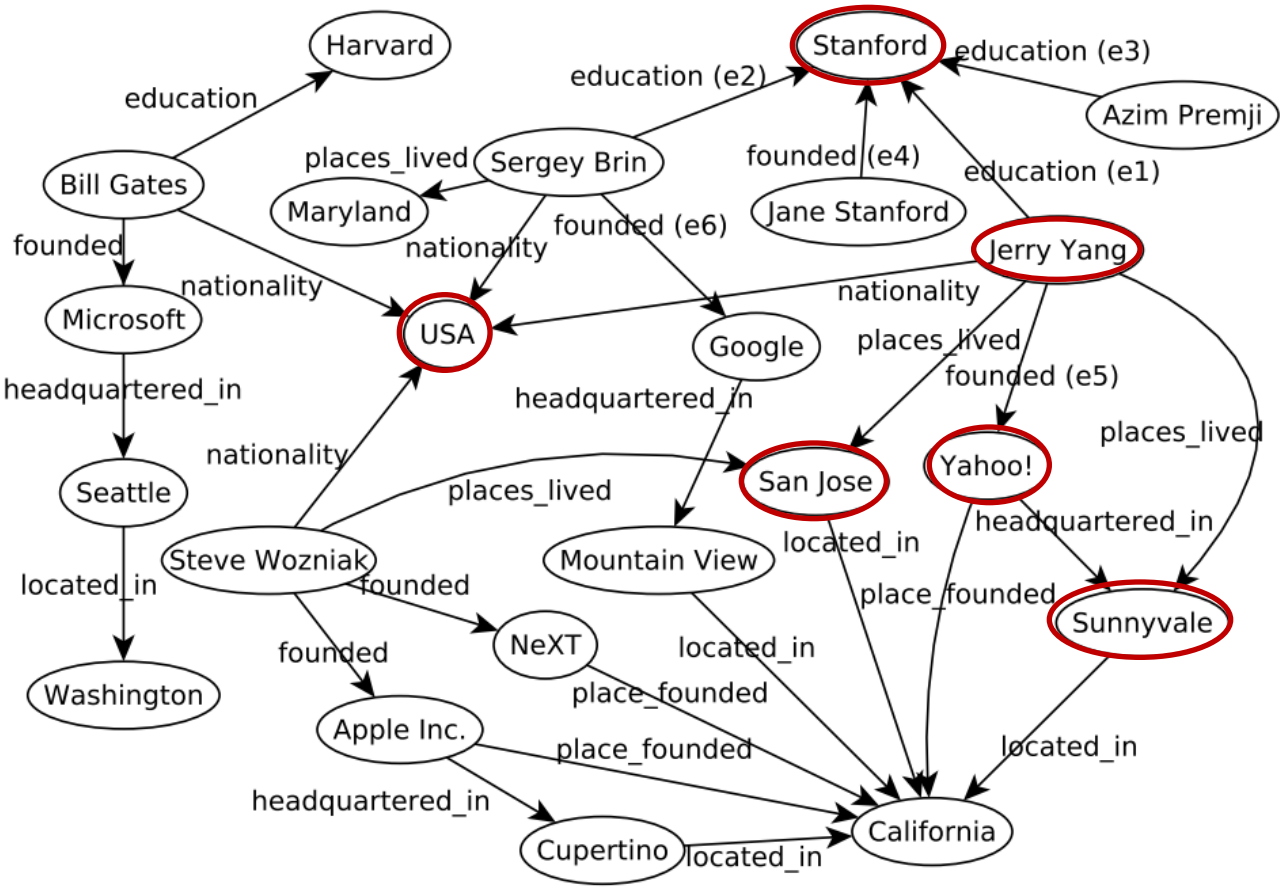
Additional useful operators (cont.)

- Comparison
  - Which is the better match?
    - f_1 *BETTER MATCH THAN* f_2



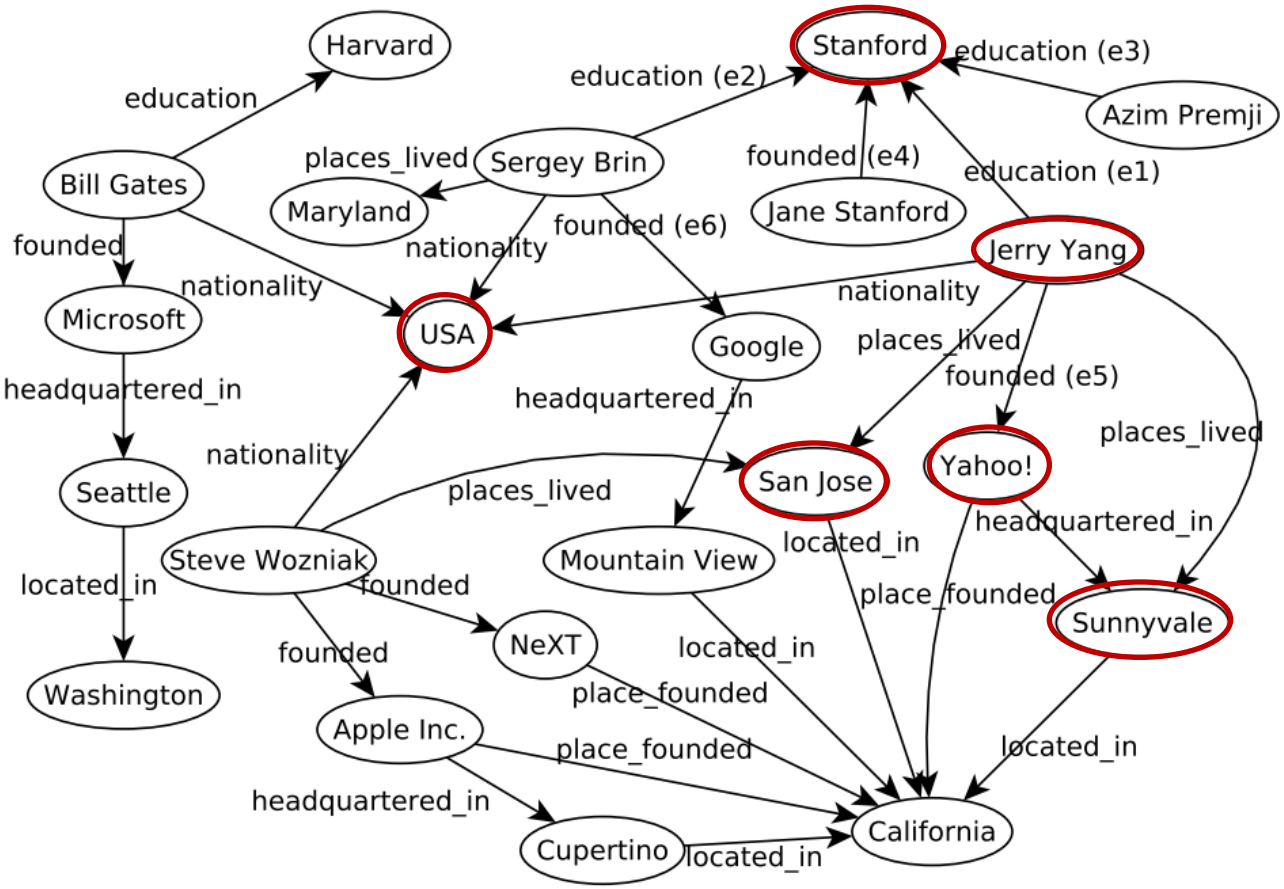
In class exercise

- What do we know about Jerry Yang?



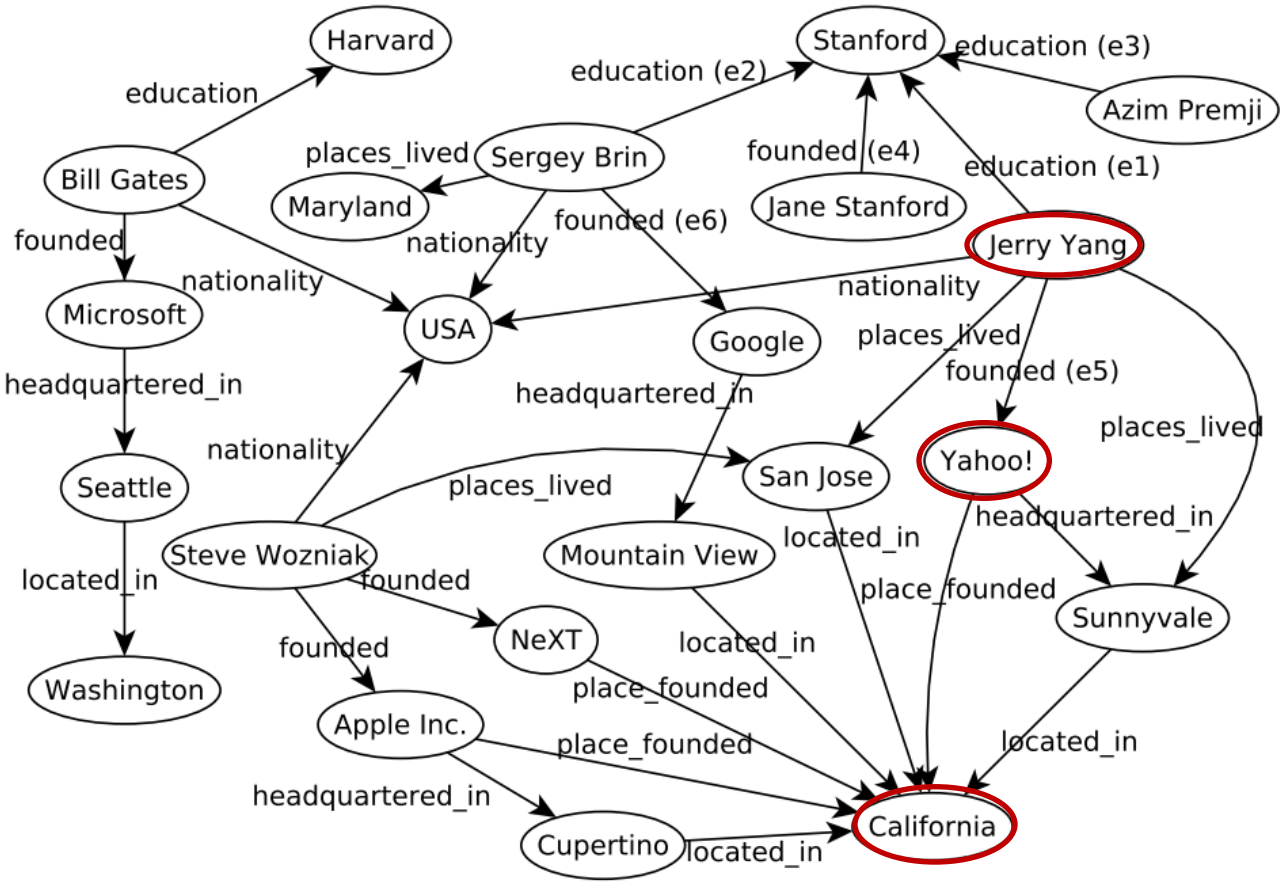
Observation

- Querying = traversing the graph



In class exercise

- Find entrepreneurs (like Jerry Yang) who have established high-tech start ups (like Yahoo!) based in California

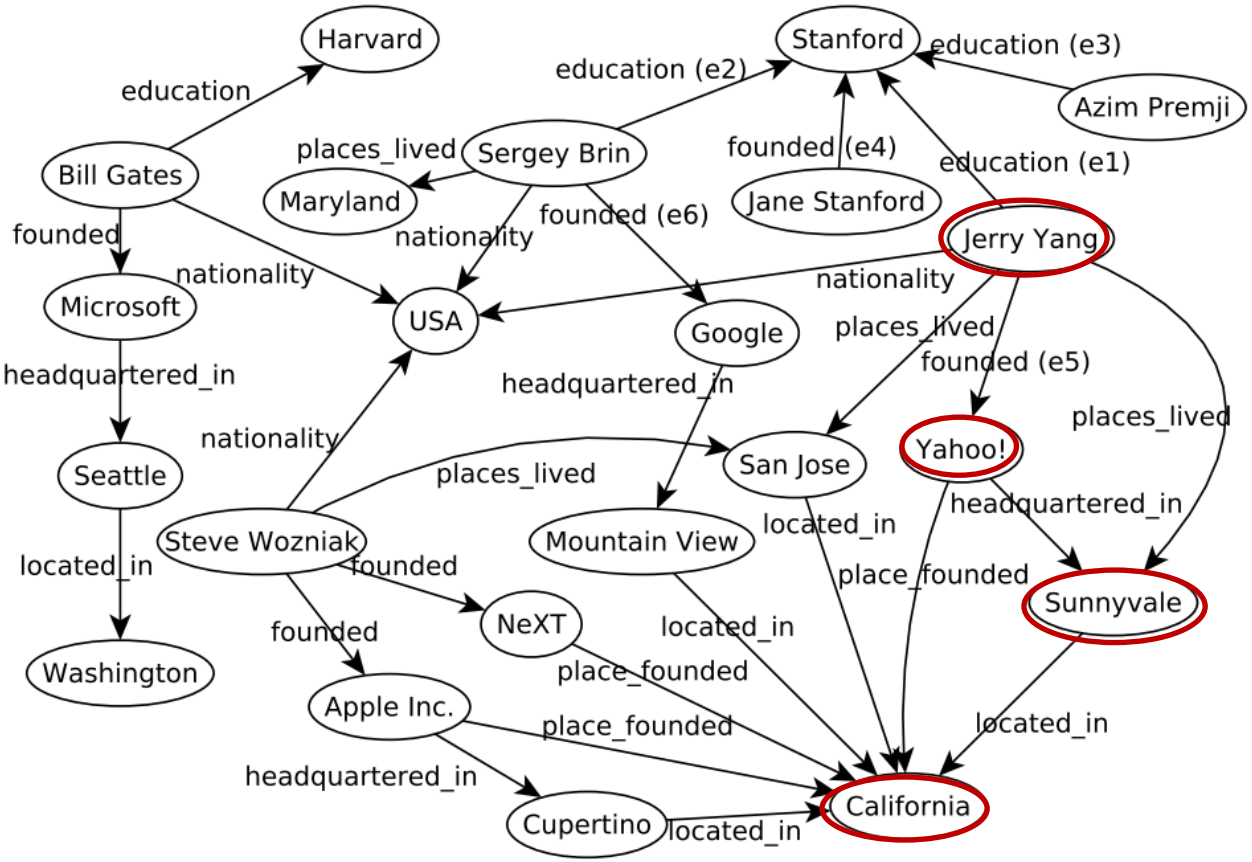


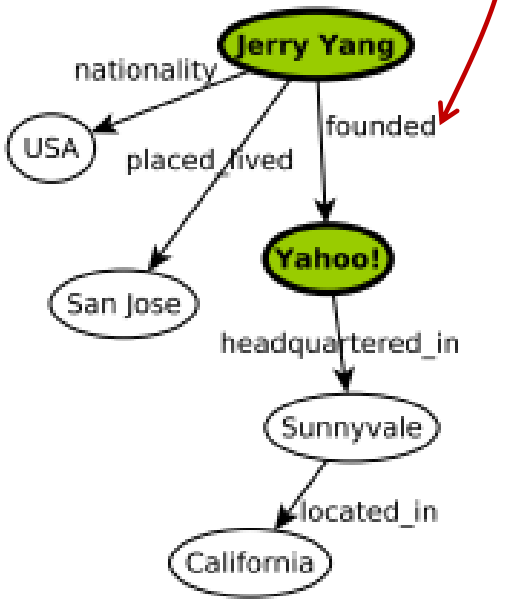
Find results that match <Jerry Yang , Yahoo!, CA>

In class exercise (cont.)

- Find entrepreneurs (like Jerry Yang) who have established high-tech start ups (like Yahoo!) based in California

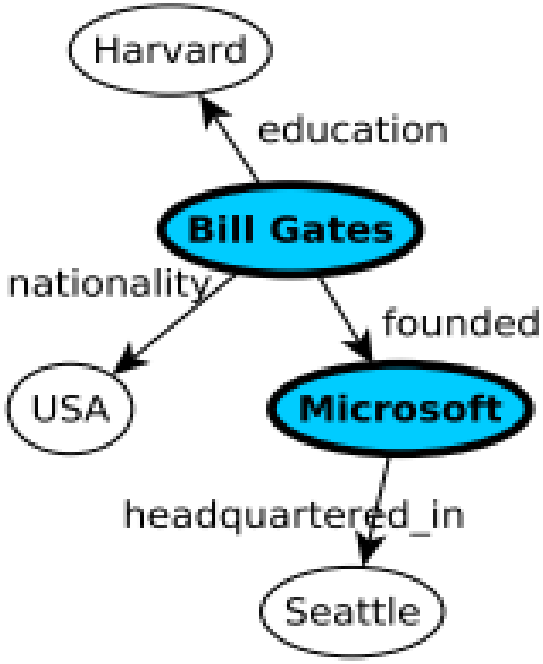
Find results that match
<Jerry Yang , Yahoo!, CA>

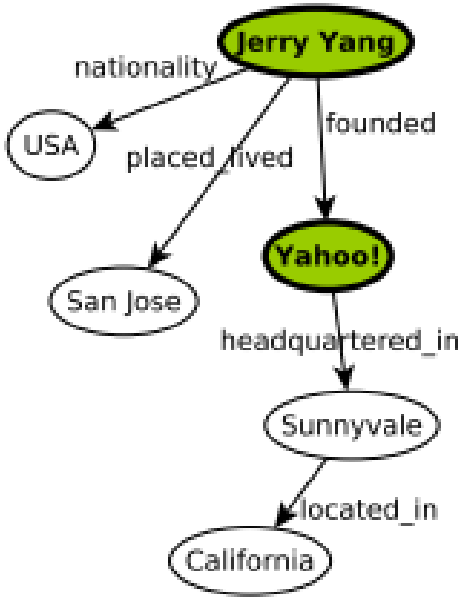


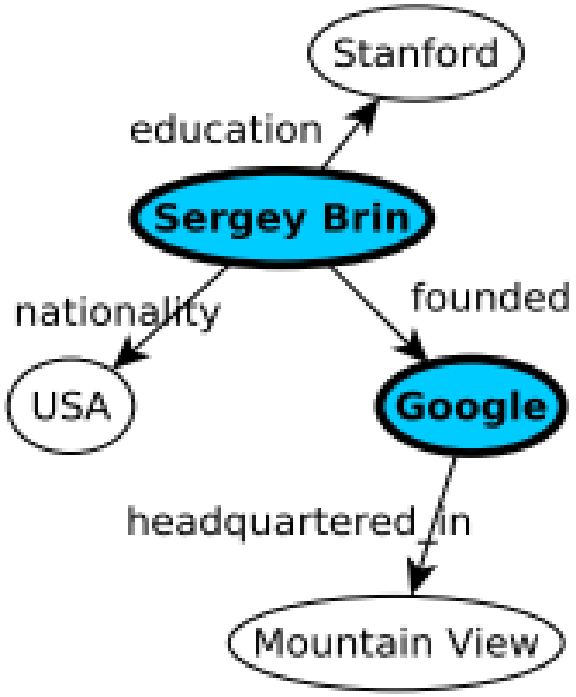


In class exercise

- Find others like Jerry Yang …

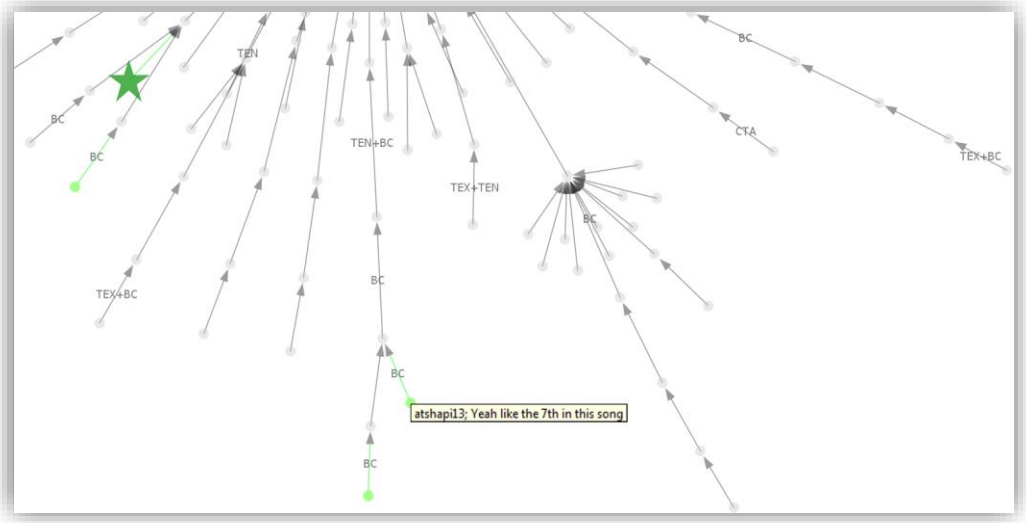






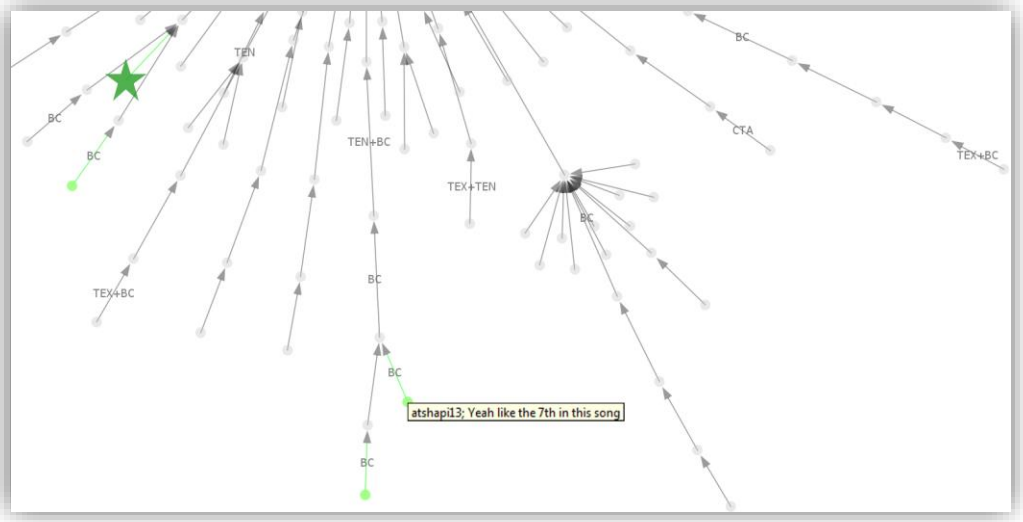
Example from a past dissertation

- How would you study the intensity of bonding as a result of *number of posts* exchanged by users



Example from a past dissertation

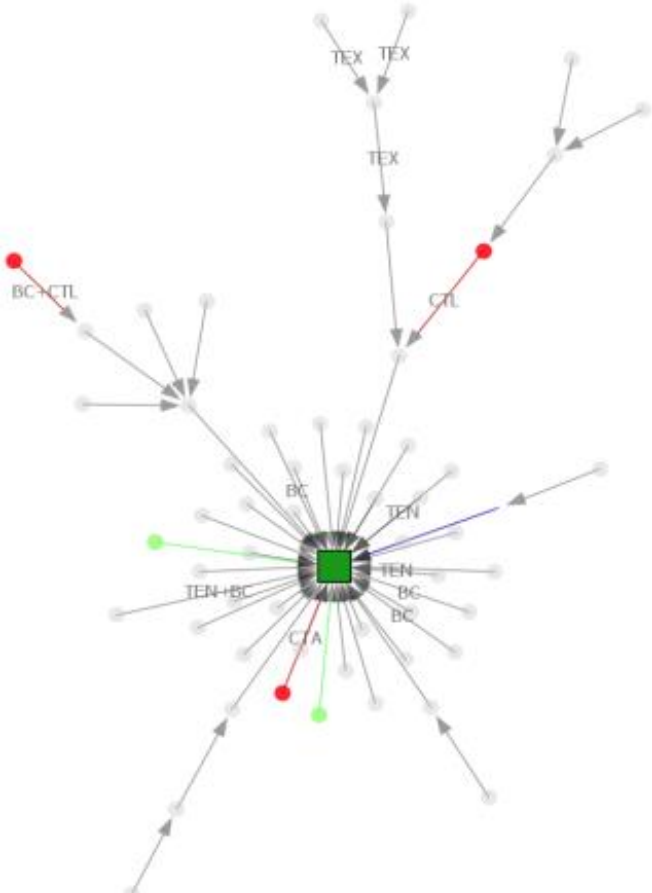
- How would you study the intensity of bonding as a result of *number of posts* exchanged by users



- Maybe you distinguish *users who exchange posts regularly* from those doing it sporadically

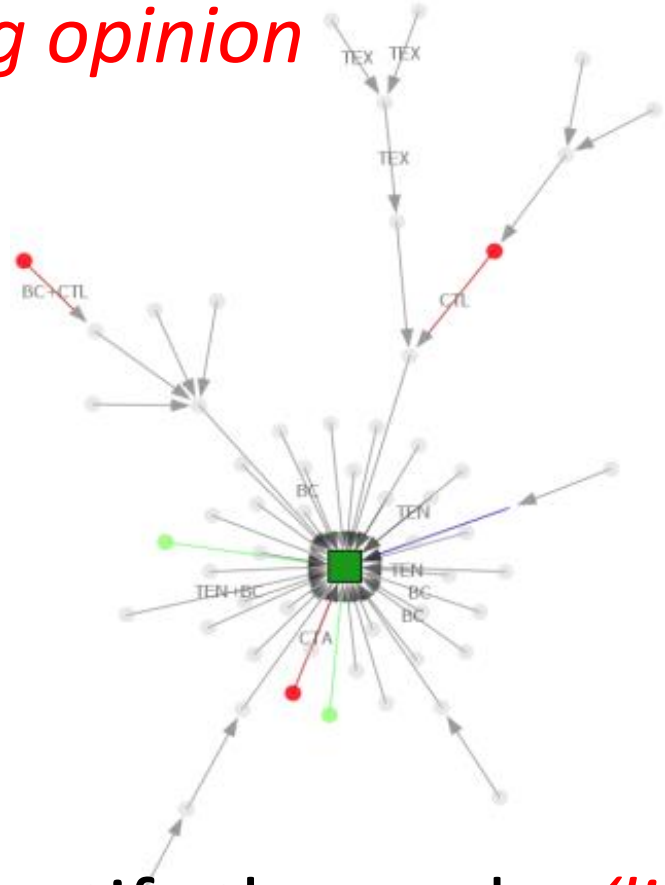
Example from a past dissertation

- How would you study the intensity of bonding as a result of *expressing opinion*



Example from a past dissertation

- How would you study the intensity of bonding as a result of *expressing opinion*



- Maybe of interest to identify those who *'like'* certain posts

Streaming data

Definition

- Streaming data is the process by which a continuous flow of data from single or multiple sources (sensors, apps, users) is processed in near real-time
  - The concept of *streaming* refers to how data is captured and delivered rather than the content of the data
- Important
  - Stream *processing* techniques (in data management) allow continuously ingesting, analyzing, and transforming high-volume, real-time data as it is generated, rather than in batches

Data sources

- Data sources include
  - financial systems
  - third-party data providers
  - social media platforms
  - IoT devices
  - SaaS apps
  - on-premises business applications like enterprise resource planning (ERP) and customer relationship management (CRM)

Streaming data examples

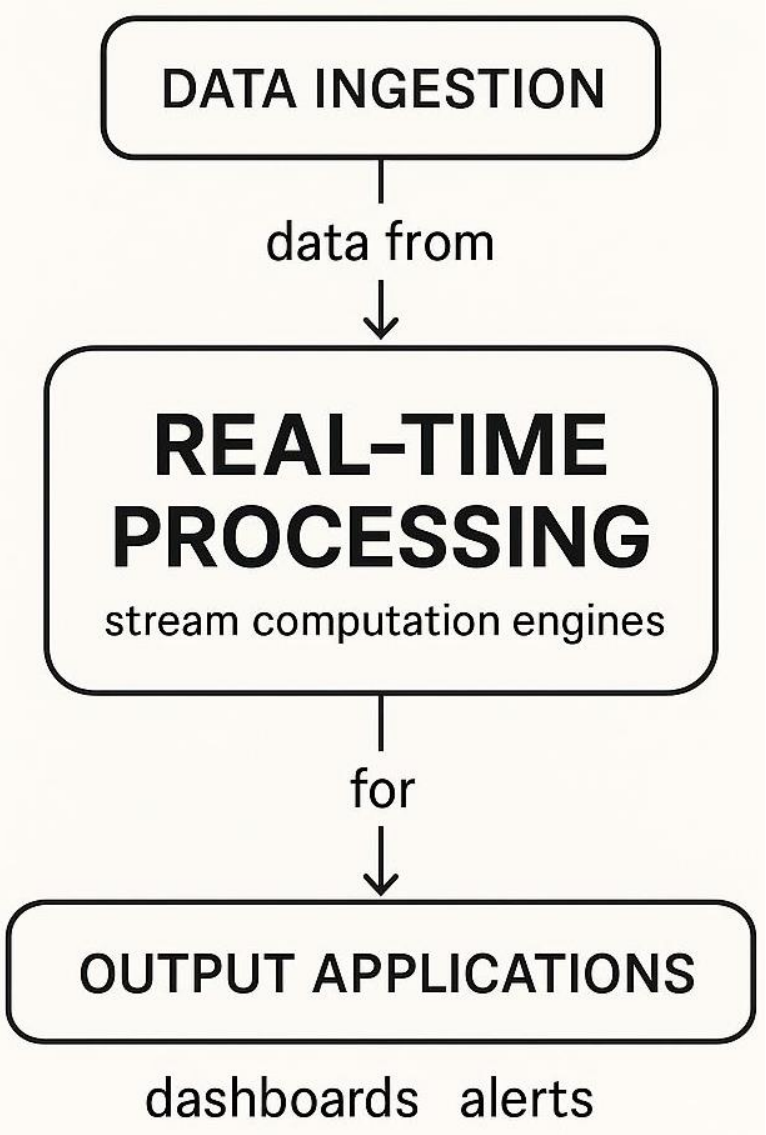
- Financial trading platforms use it to provide real-time price changes for stocks and currencies
  - Stock information services use streaming data to share company news as it breaks, helping institutional and individual investors make more informed trading decisions.
- Sales and marketing systems can use clickstream data to trigger an interaction with a chatbot or agent
- Gaming companies use in-game behavior analytics to suggest new games or offer the most relevant ads for in-game purchases
- Security systems use sensors to detect suspicious activity. Sensors collect video streams that are analyzed, and alerts are generated when potential threats are observed
- Autonomous driving uses real-time sensor input to control vehicle speed and safety systems. Cameras, sonar, and lidar sensors generate data streams for image processing software to analyze
- Industrial systems use sensors to monitor manufacturing systems for quality control and drive production. Digital streams enable manufacturers to remotely monitor the health of systems such as locomotive engines to make timing decisions for preventive maintenance, ordering parts, and alter performance to maximize the equipment's useful life
- Marketing systems use clickstream data to analyze what ads and web pages a prospect views so chatbots can offer the most compelling real-time engagement tactics
- Retail streamed data from in-store beaconing systems to inform text and email offers based on the shopper's location

Common goal, differ approaches

- Data ingestion
  - Data ingestion encompasses all the tools and processes responsible for collecting, extracting and transporting data from diverse sources for further processing or storage
- Extract, transform and load (ETL) is the process by which data is
  - extracted from its source system
  - transformed to meet the target system's requirements
  - loaded it into the designated data warehouse or data lake
- Extract, load and transform (ELT) is the process by which data is
  - extracted from its source
  - loaded into the target system
  - transformed on-demand and as needed for specific analyses

Ingestion engines

- Ingestion engines for stream processing are tools designed to continuously capture, buffer, and transport high-velocity, real-time data from diverse sources (IoT, logs, apps) to processing frameworks, ensuring low-latency data availability



Example

- Consider a data source that continuously produces data such as the following

A B C D E F G H I G K L M N O

- ✓ How does the *continuous* flow of data influence its processing?
  - Do we need to await until the flow completes and all data are made available?
    - ✓ What if there is no time or application is time-critical or the decision is data-driven?
    - ✓ What if the flow never completes?
  - Can processing take place with partial data
    - ✓ What kind?

Example (cont.)

- Consider a data source that continuously produces data such as the following

  A B C D E F G H I G K L M N O

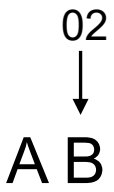
- ✓ One way (not the only one) is to *break* the flow into sub-flows based on parameters
  - Length of data stream
  - Shift in data stream

Example (cont.)

- An example with Length=5 and Shift=2

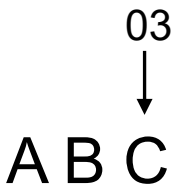
Example (cont.)

- An example with Length=5 and Shift=2



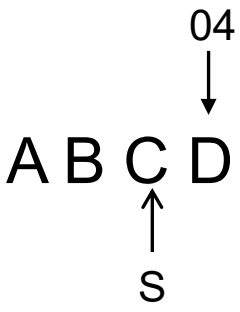
Example (cont.)

- An example with Length=5 and Shift=2



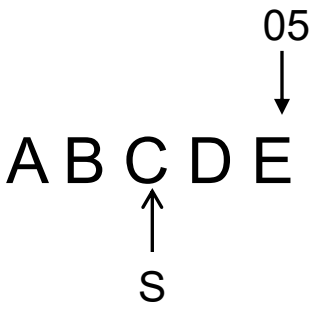
Example (cont.)

- An example with Length=5 and Shift=2



Example (cont.)

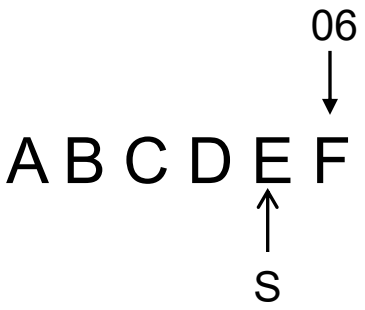
- An example with Length=5 and Shift=2



A B C D E
C D

Example (cont.)

- An example with Length=5 and Shift=2

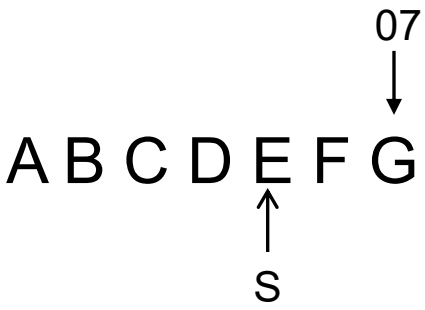


A B C D E

C D E

Example (cont.)

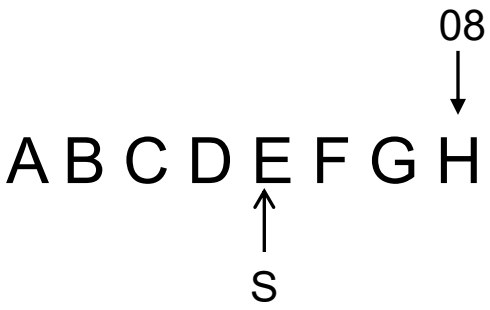
- An example with Length=5 and Shift=2



A B C D E
C D E F

Example (cont.)

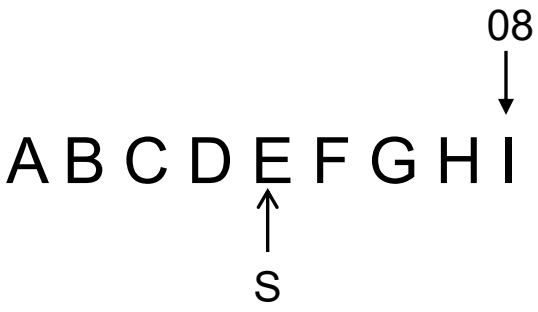
- An example with Length=5 and Shift=2



A B C D E
C D E F G

Example (cont.)

- An example with Length=5 and Shift=2



A B C D E
C D E F G
E

Example (cont.)

- At the end the result is something like the following

A B C D E F G H I G K L M N O

A B C D E
C D E F G
E F G H I
G H I J K
I J K L M
K L M N O

✓ Note

  - Stream processing can commence before the full set of data is available

Summary and conclusion

- As data change in nature new data types have come to facilitate their representation and processing
- Implications
  - The relational model is limited
  - New data models appear to address semantics challenges
  - Query languages follow the new data models

Next time

- New data types
  - Enumerated
  - 1D and multi-dimensional arrays
  - User defined datatypes
  - XML
  - JSON
- Specialization hierarchies & inheritance
- Practice and experience

This is the end