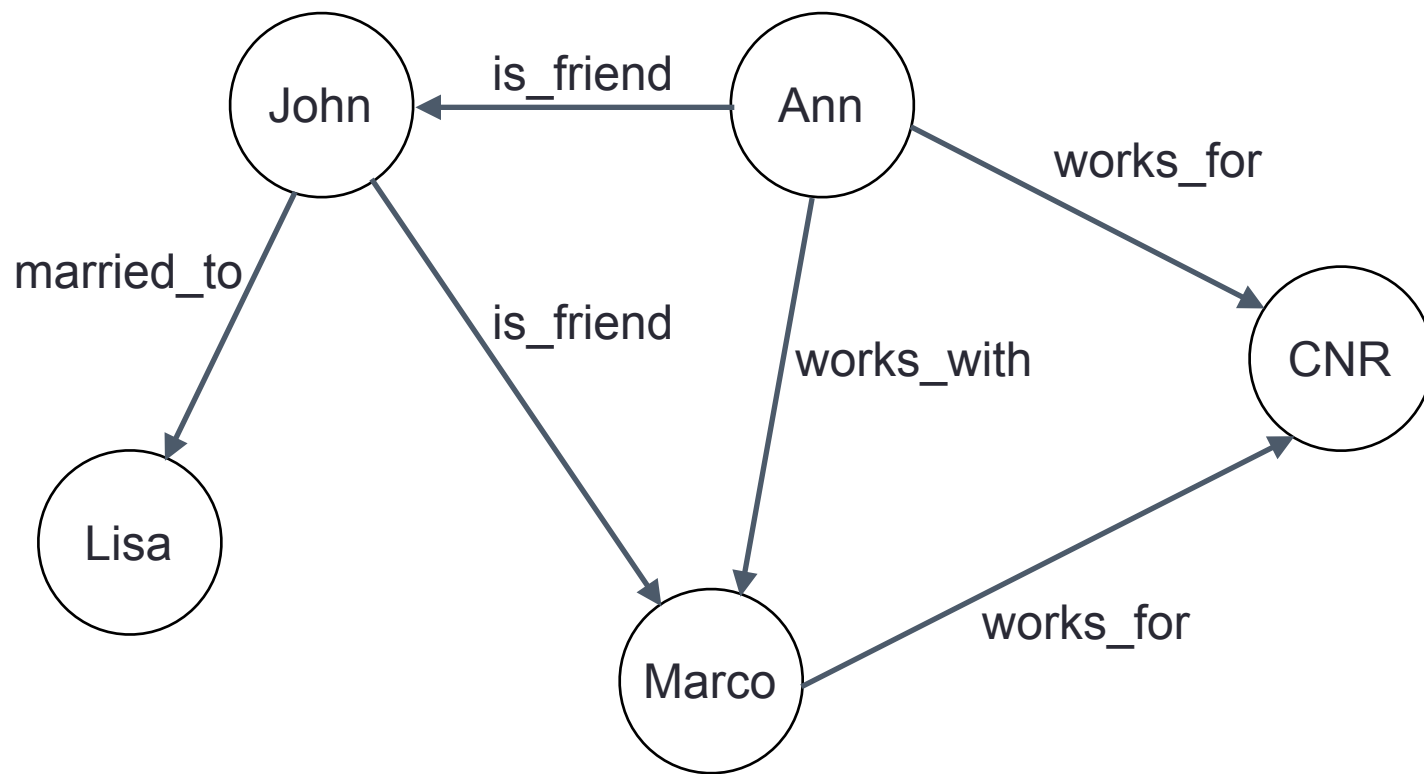
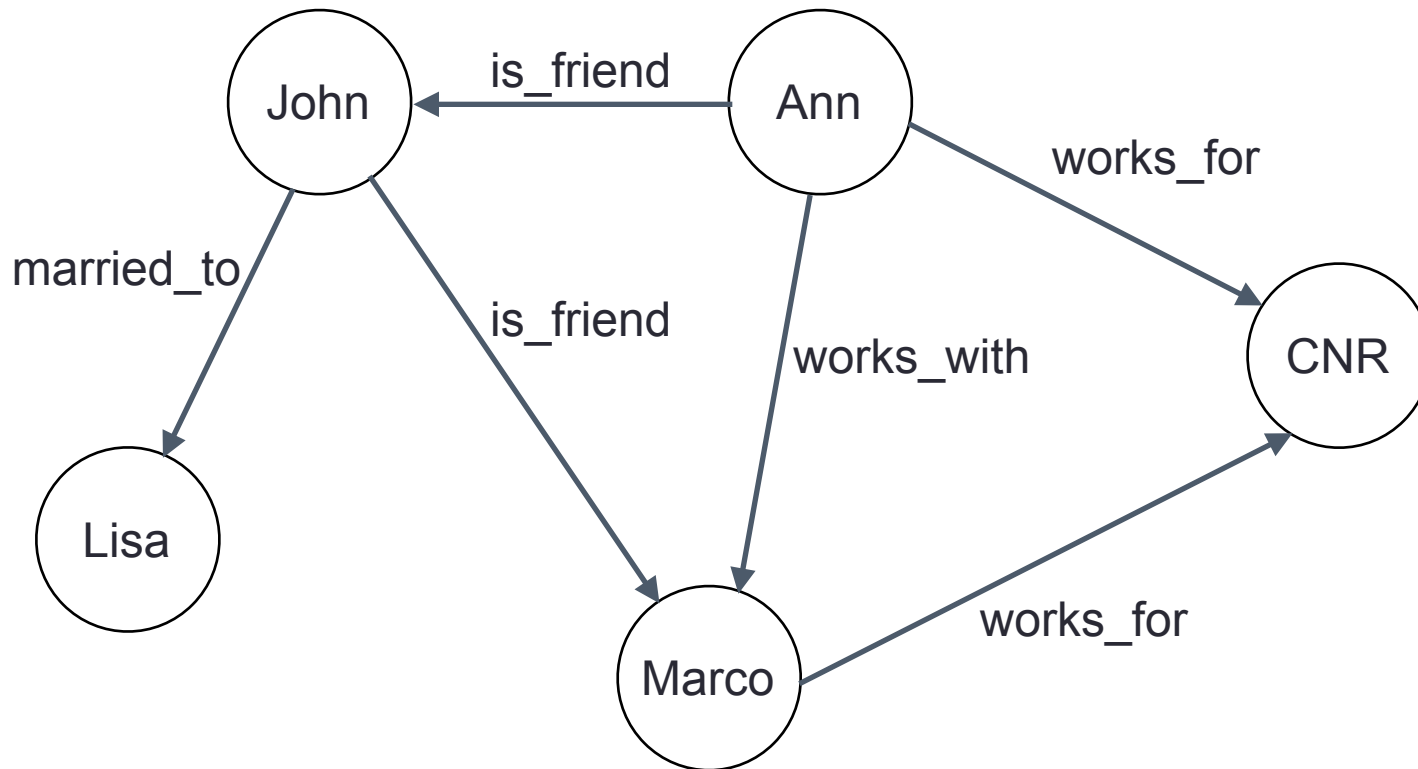


Graph Databases

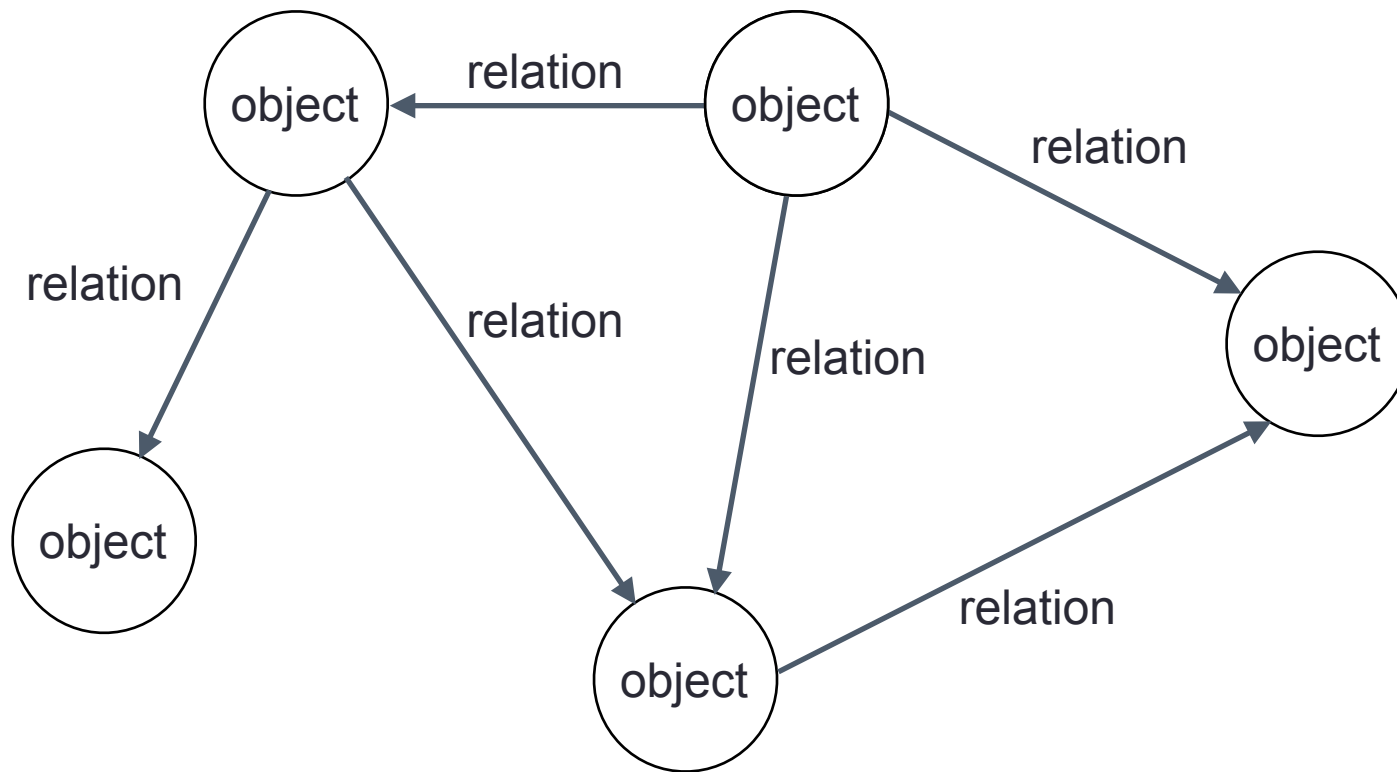


Graph Databases

- Social Networks
- Biological Databases
- Geographical Models
- ...



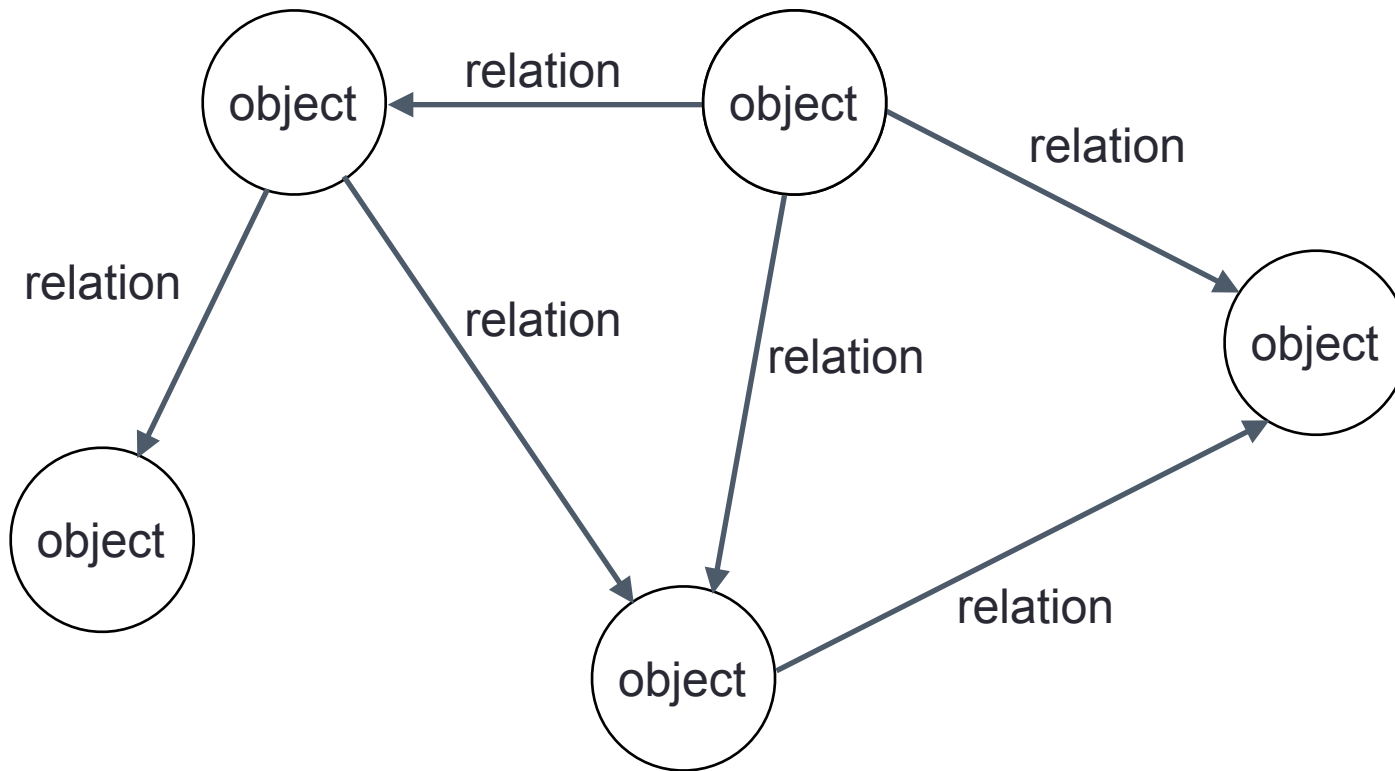
Graph Databases



Graph Databases

Basic model:

- Nodes represent objects
- Edges are relations



Querying graphs via pattern matching

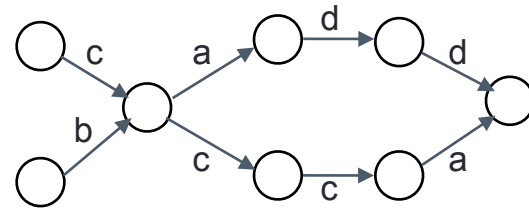
Querying graphs via pattern matching



Graph Database

Querying graphs via pattern matching

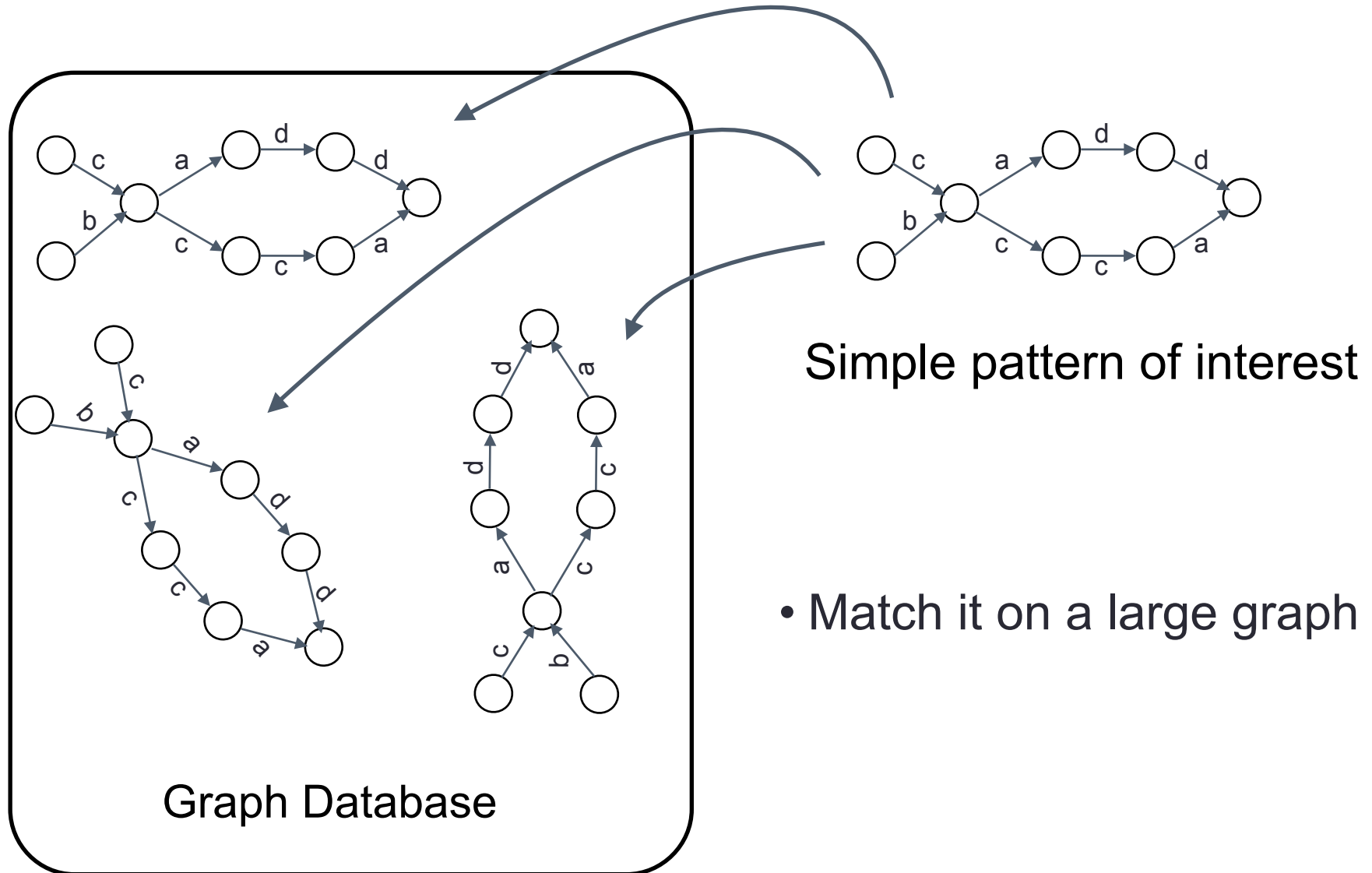
Graph Database



Simple pattern of interest

- Match it on a large graph

Querying graphs via pattern matching

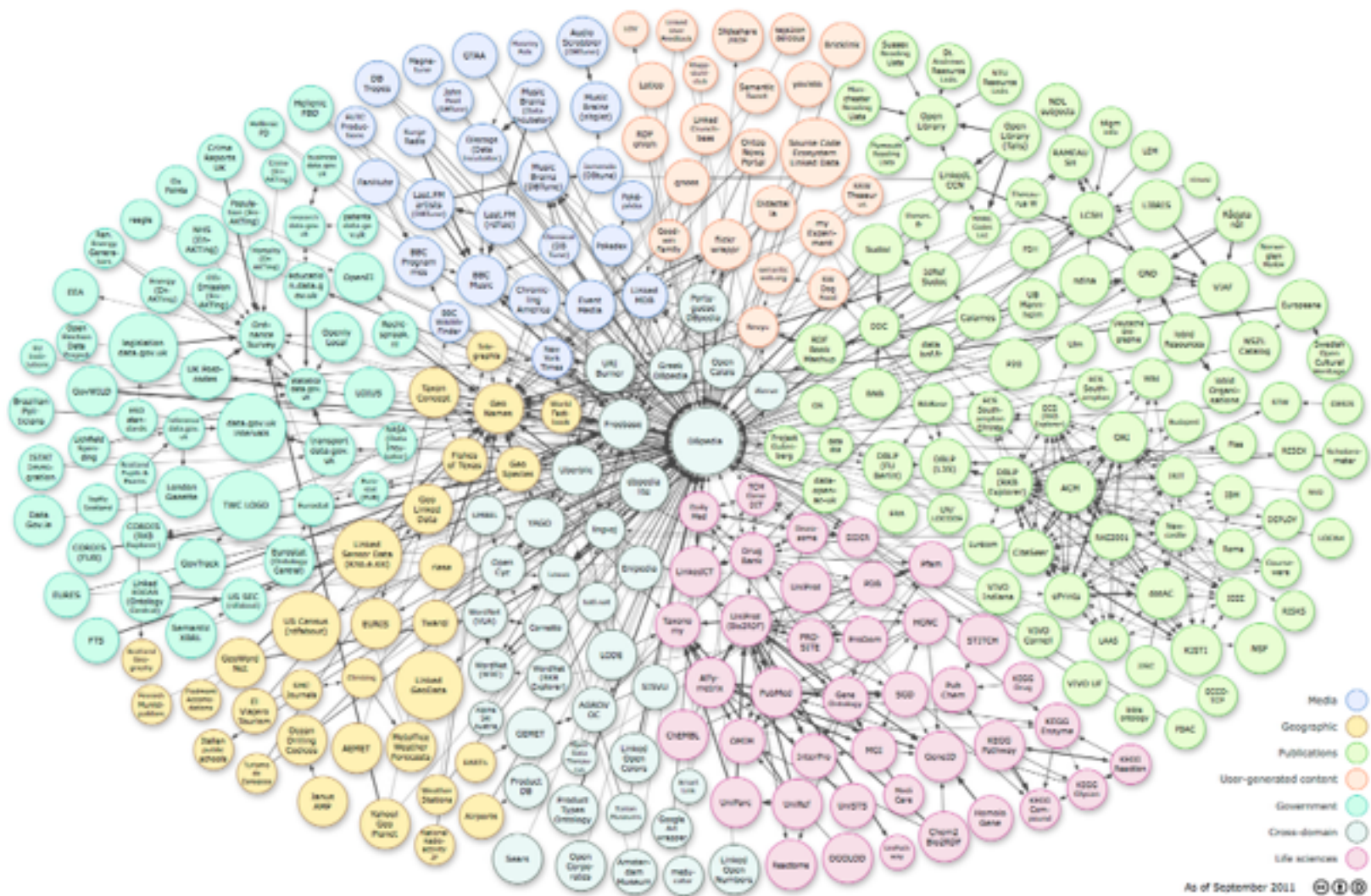


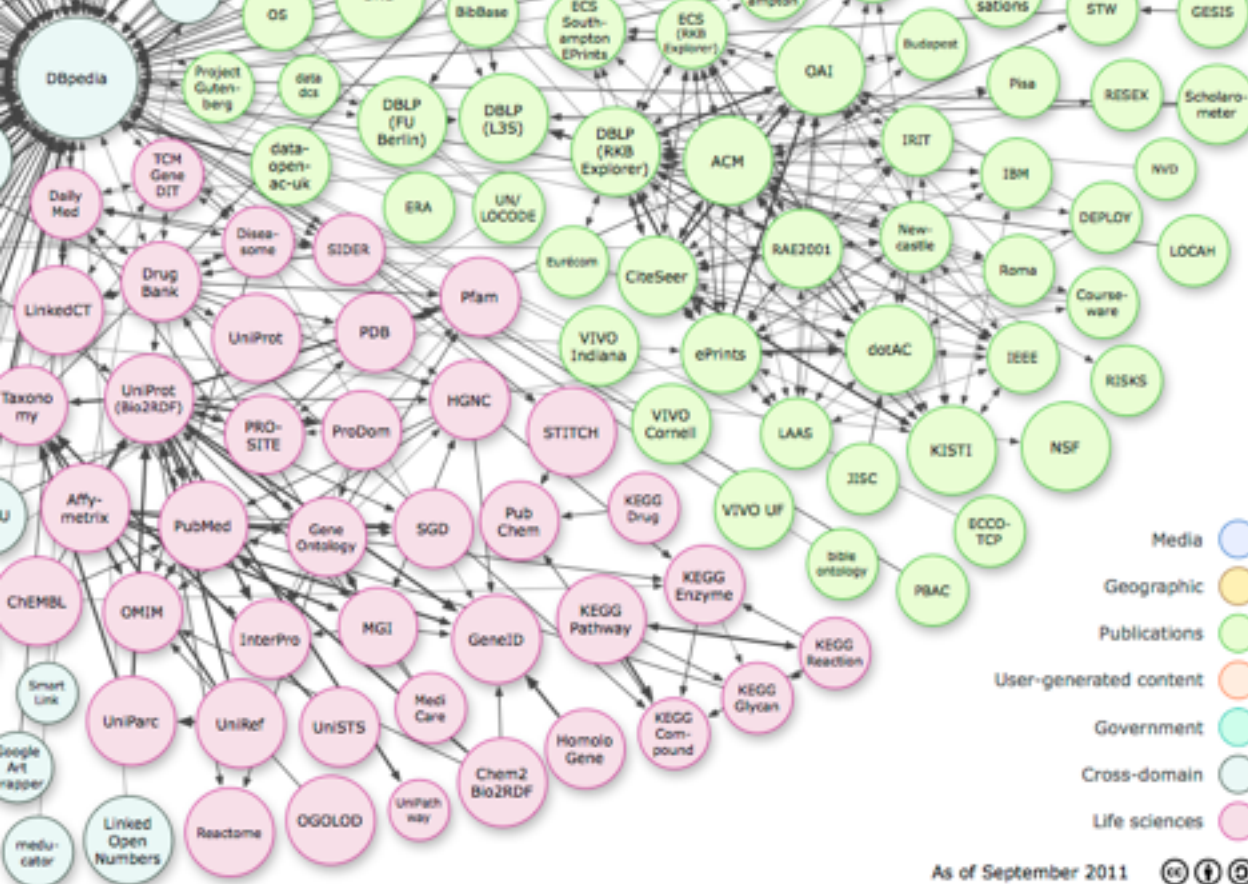
Pattern matching is a well studied problem

Pattern matching is a well studied problem, but

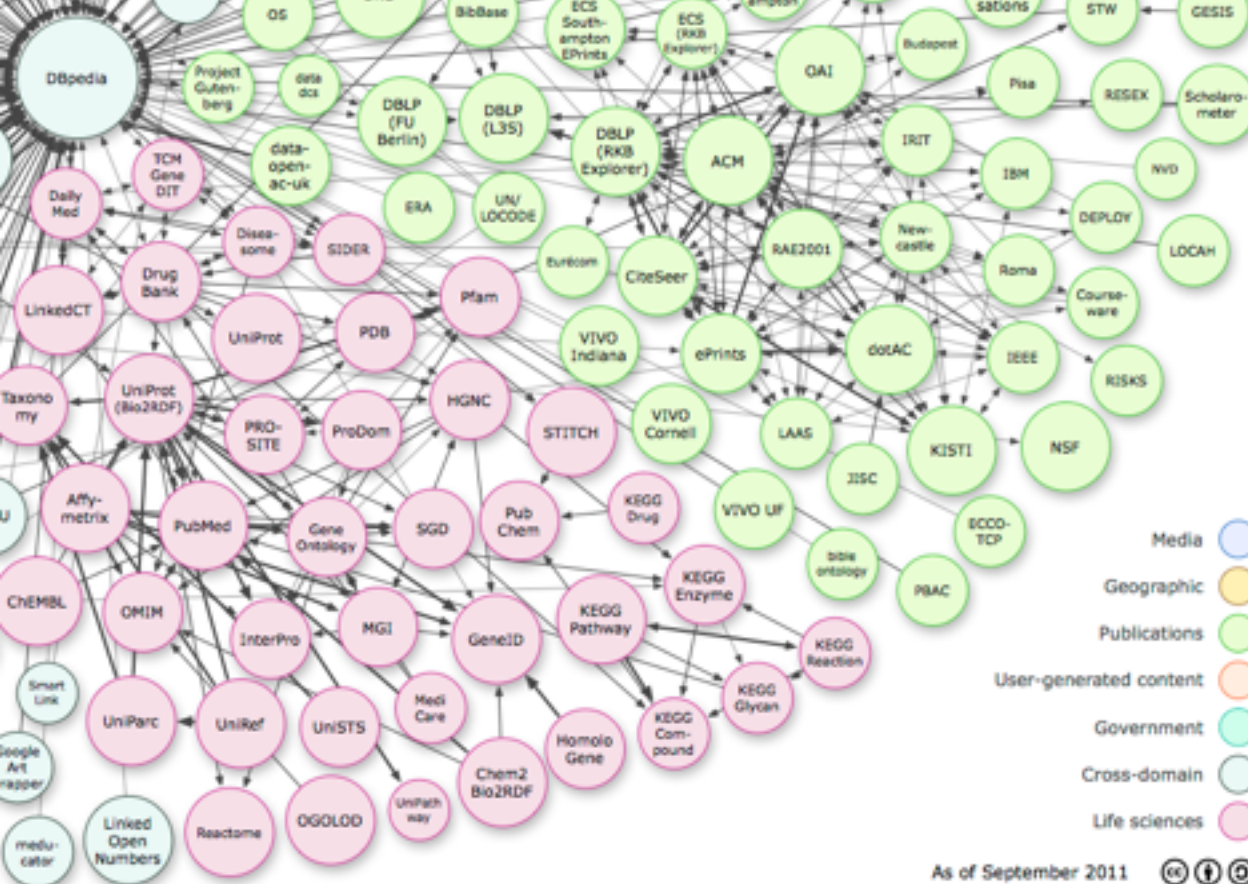
several applications **need more than pattern matching**

Need more than patterns: Linked Data





- Several graph databases connected together
- All of them store information in **different** ways



Need ways of **summarising and integrating graphs**

Idea:

Summarise the graph using a complex pattern

Idea:

Summarise the graph using a complex pattern

- Extract from the graph only the information that is relevant
- As if it was a **view of the graph**
- Then pose queries over this graph pattern

Querying Graph Databases: beyond graph patterns

Juan L. Reutter

Pontificia Universidad Católica de Chile

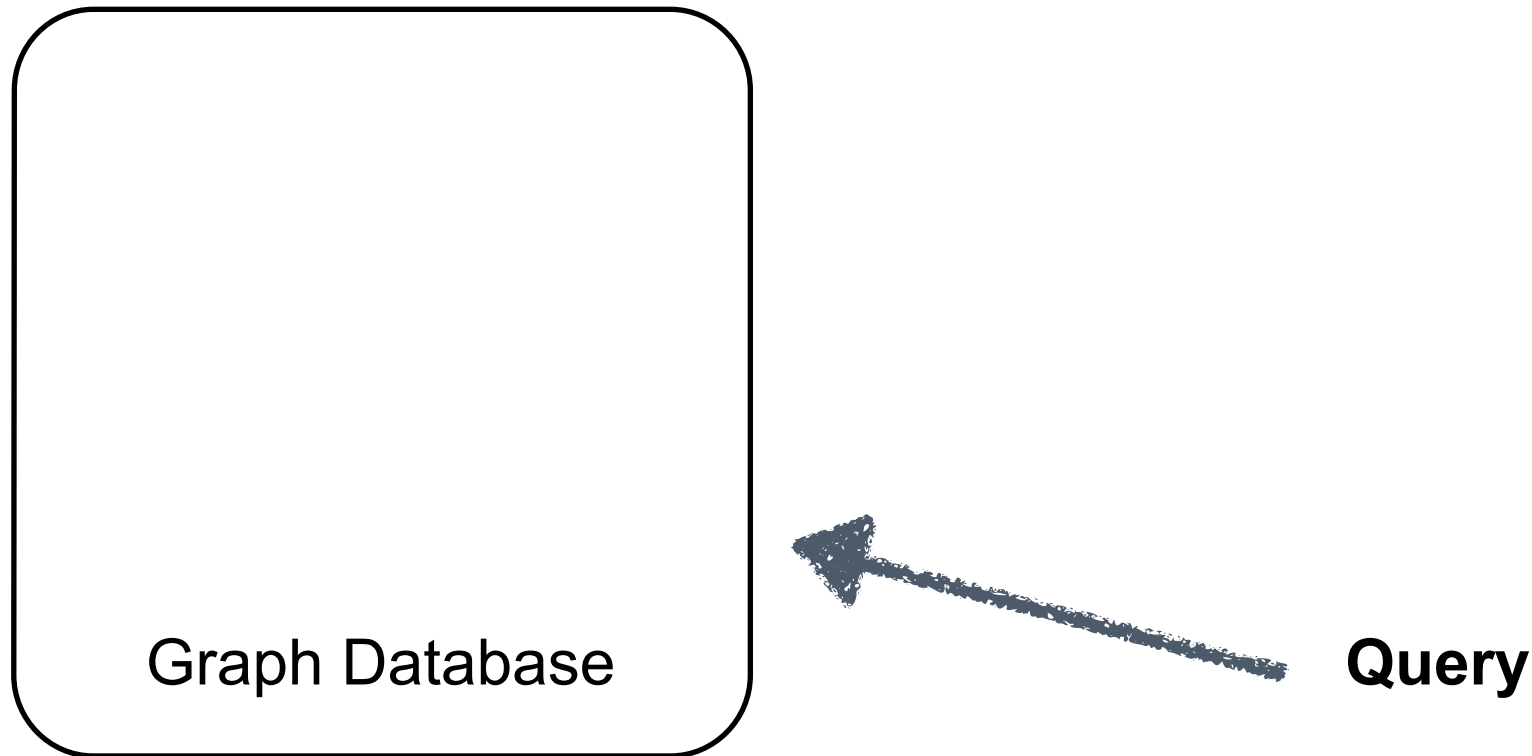
Center for **Semantic Web** Research



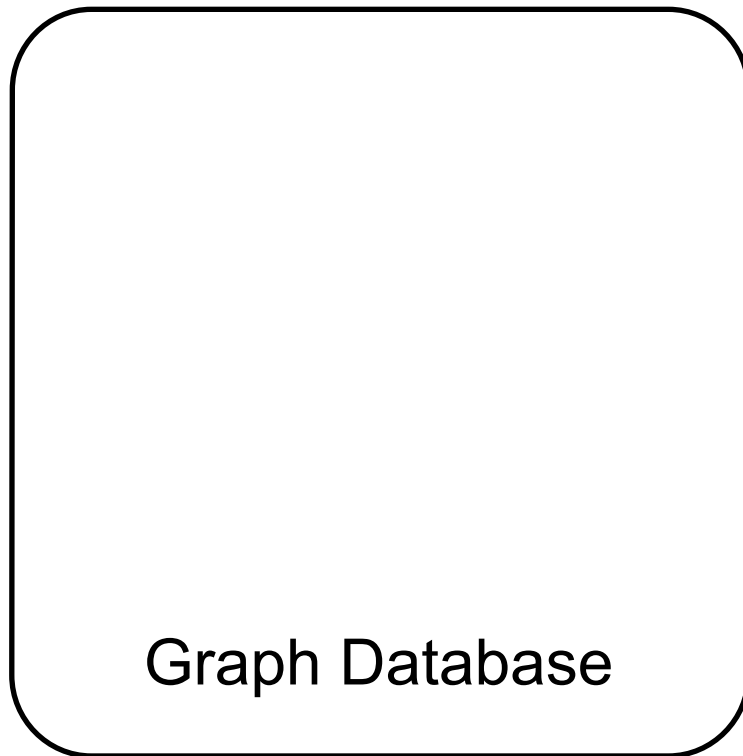
Outline

- Querying graph patterns
- Integrating multiple databases with underlying ontologies

Instead of accessing all of the graph,
summarise it as a graph pattern

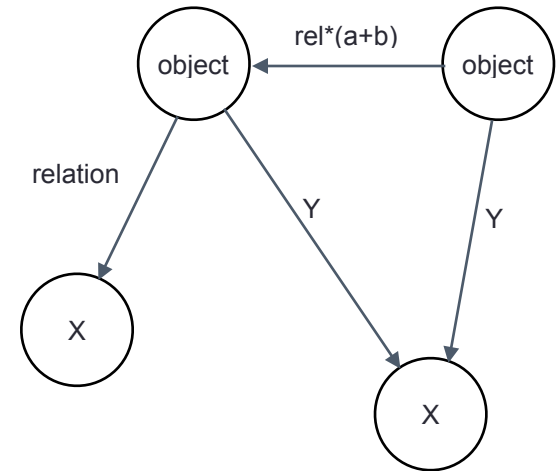
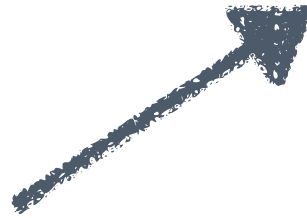
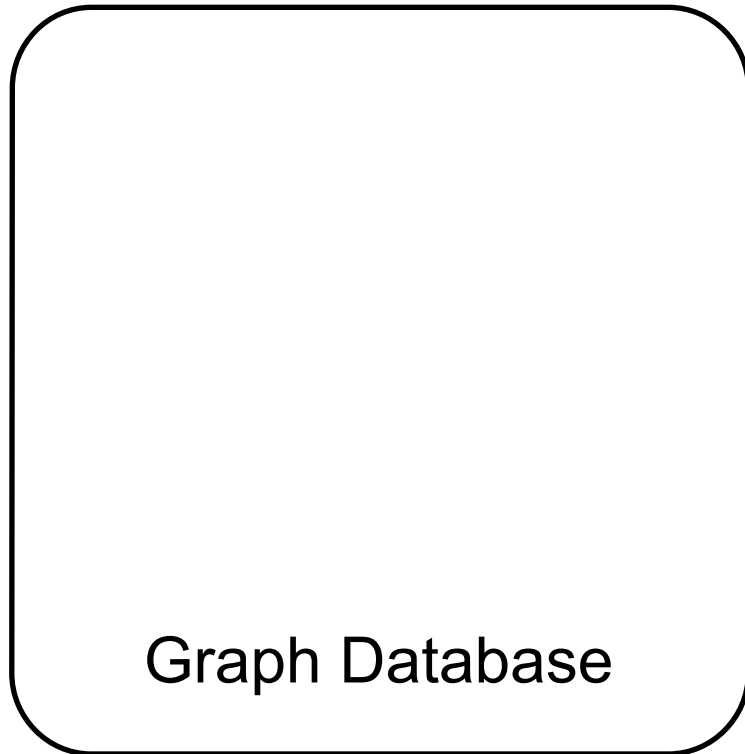


Instead of accessing all of the graph,
summarise it as a graph pattern

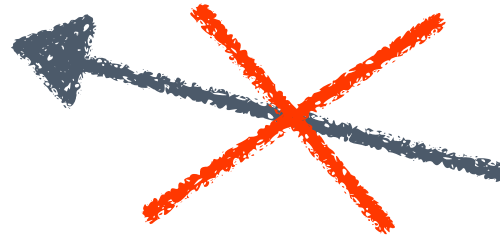


Query

Instead of accessing all of the graph,
summarise it as a graph pattern

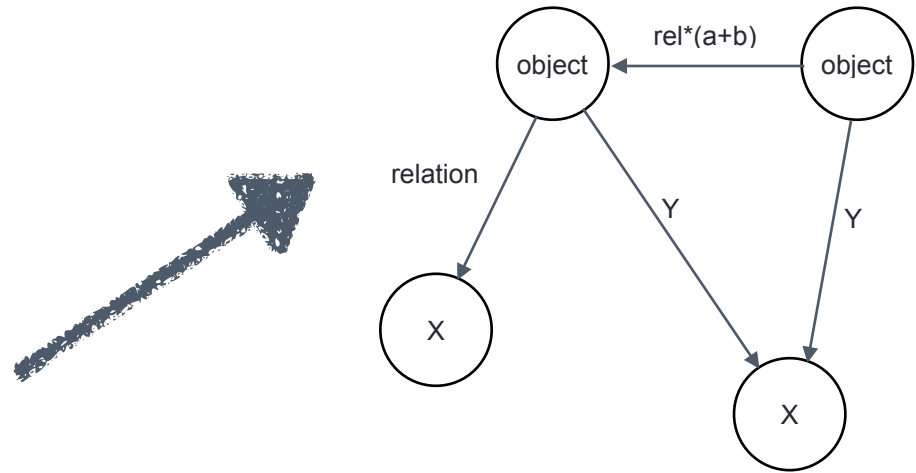
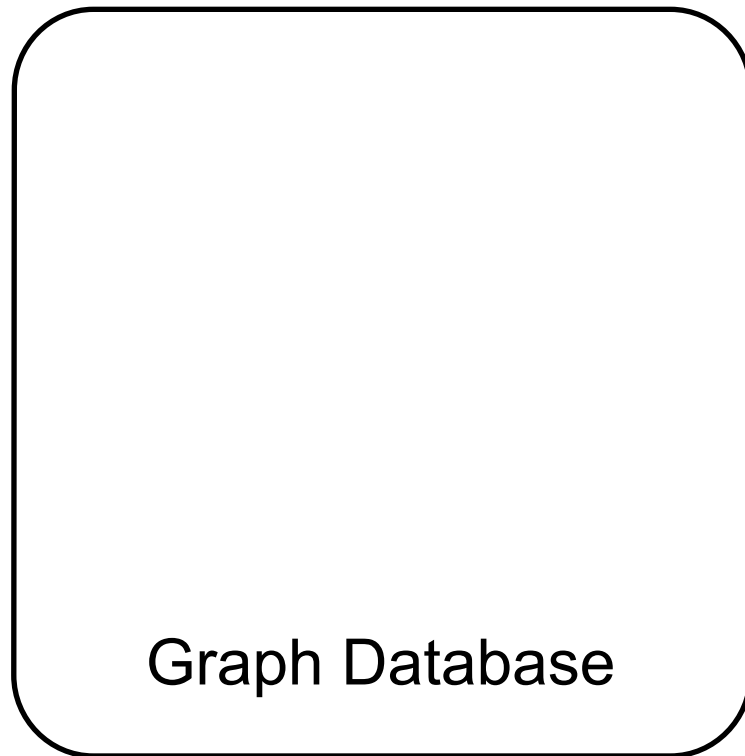


Graph Pattern



Query

Instead of accessing all of the graph,
summarise it as a graph pattern



Graph Pattern



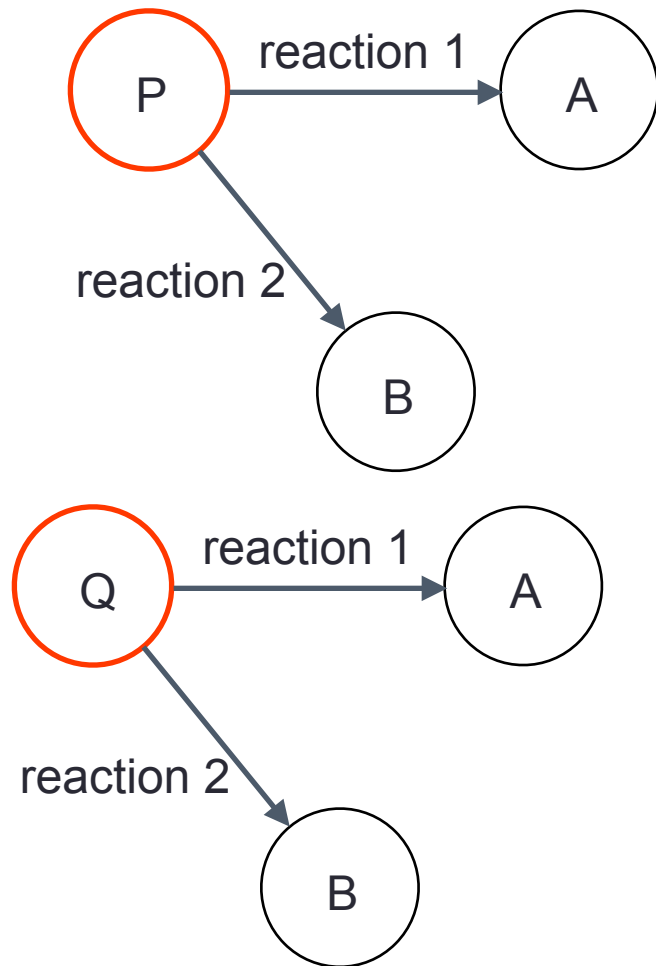
We study patterns with more expressive power

Important features:

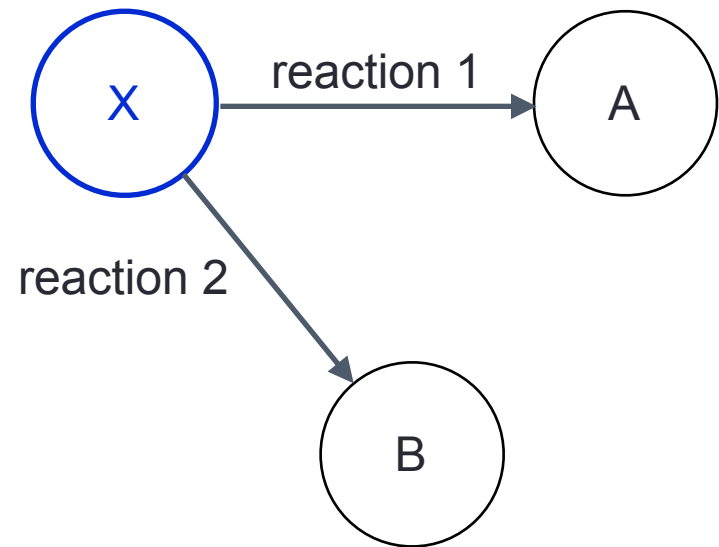
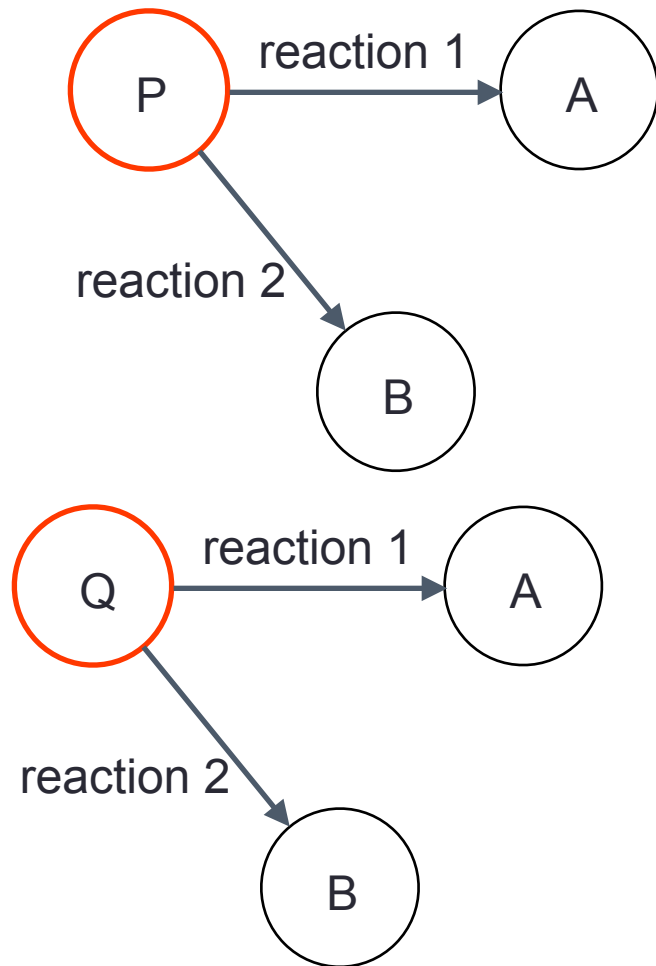
- **Node variables** to represent **objects** with same properties

Node variables

to represent objects with same properties



Node variables to represent objects with same properties



We study patterns with more expressive power

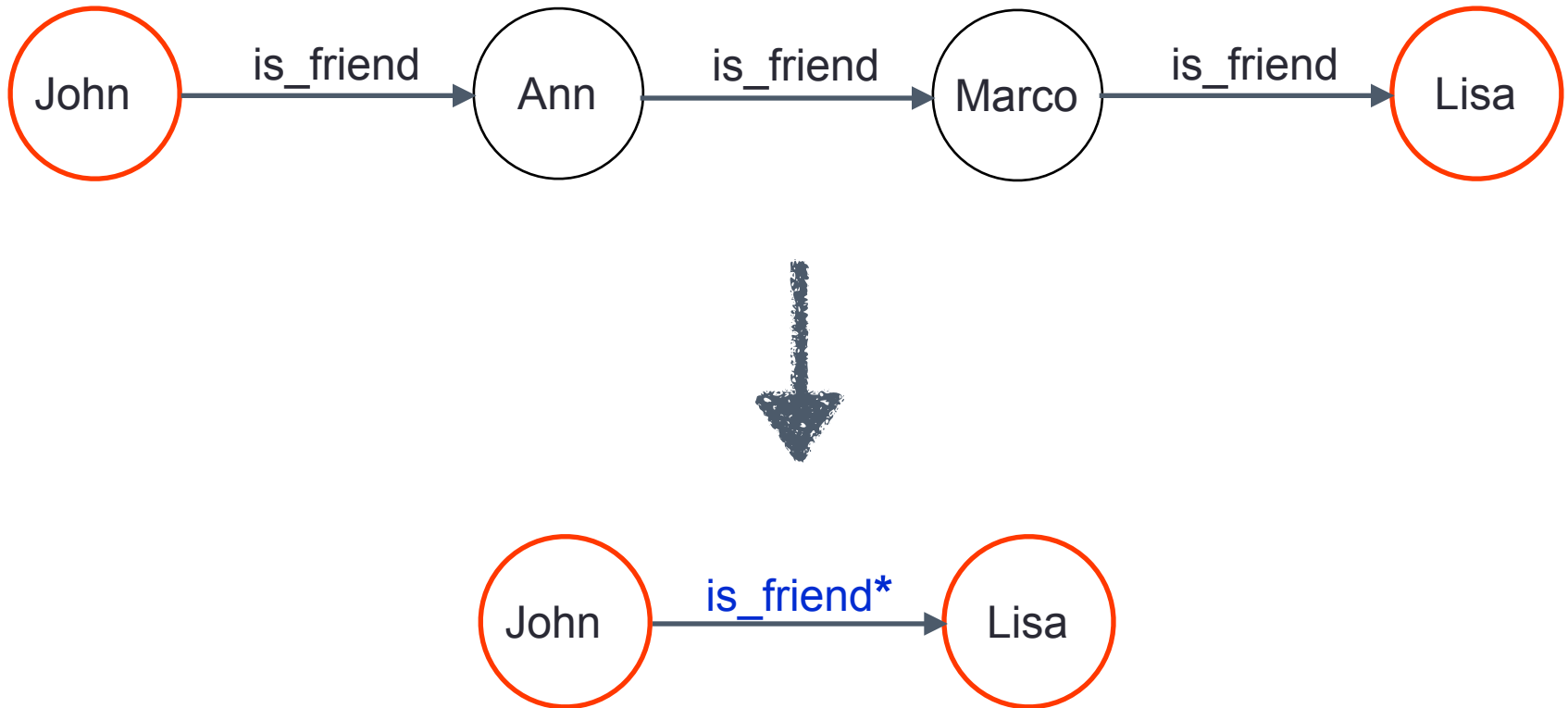
Important features:

- **Node variables** to represent **objects** with same properties
- **Regular expressions** to represent complex **paths**

Regular expressions to represent complex paths



Regular expressions to represent complex paths

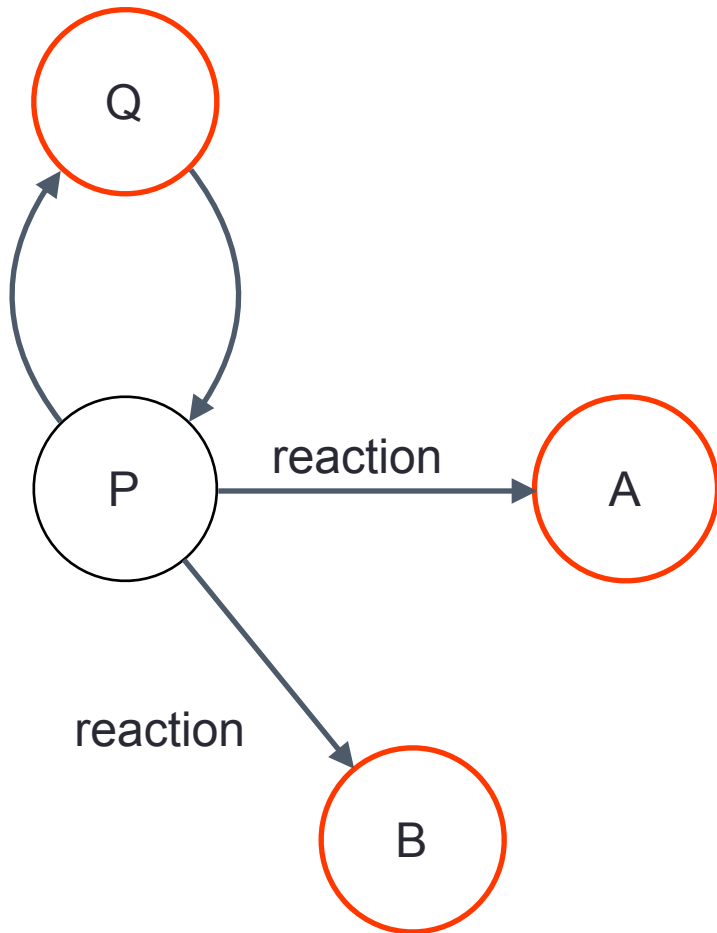


We study patterns with more expressive power

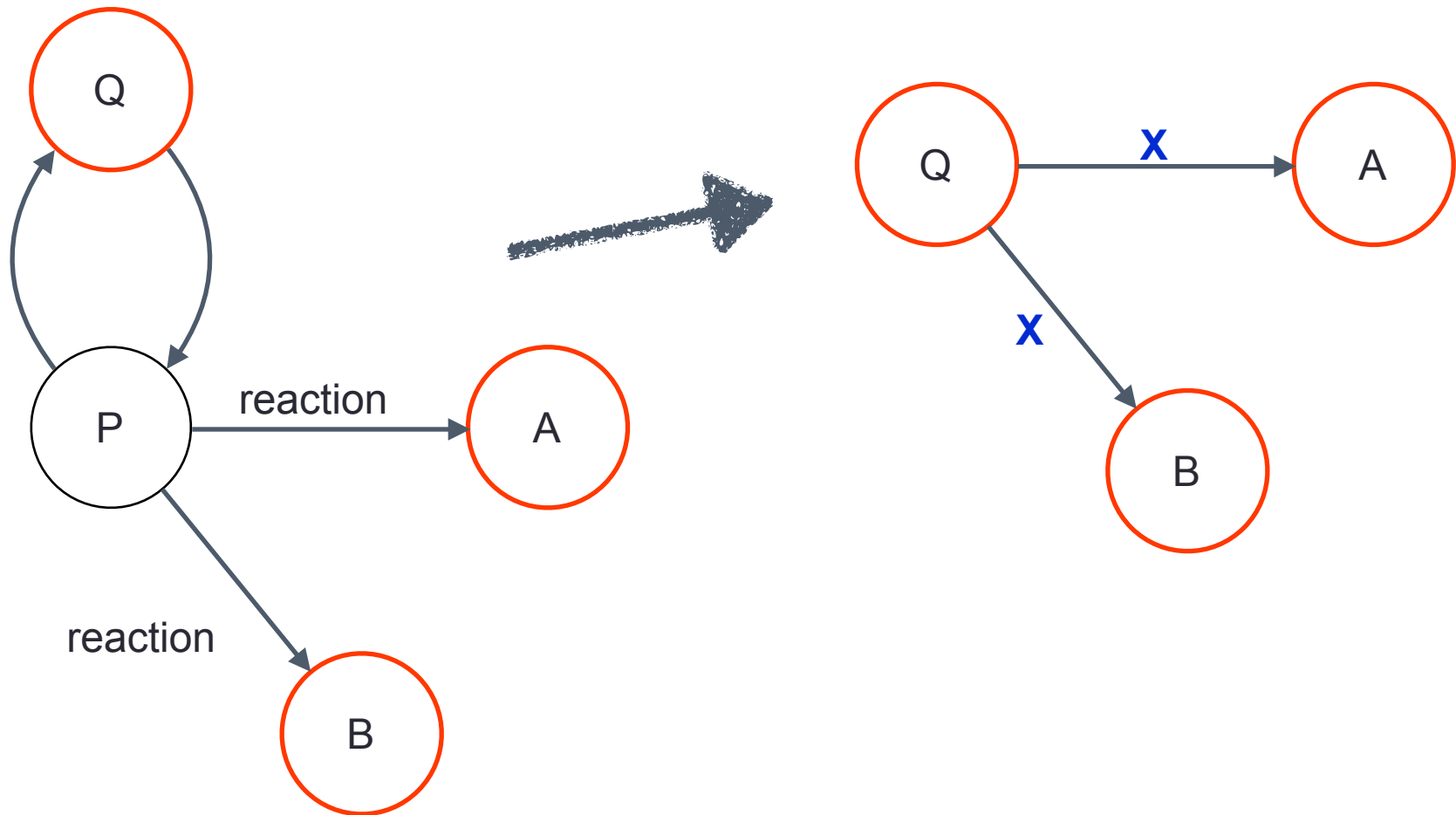
Important features:

- **Node variables** to represent **objects** with same properties
- **Regular expressions** to represent complex **paths**
- **Edge label variables** to represent **relations** with same properties

Edge label variables to represent relations with same properties



Edge label variables to represent relations with same properties



We study patterns with more expressive power

Important features:

- **Regular expressions** to represent complex **paths**
- **Node variables** to represent **objects** with same properties
- **Edge label variables** to represent **relations** with same properties

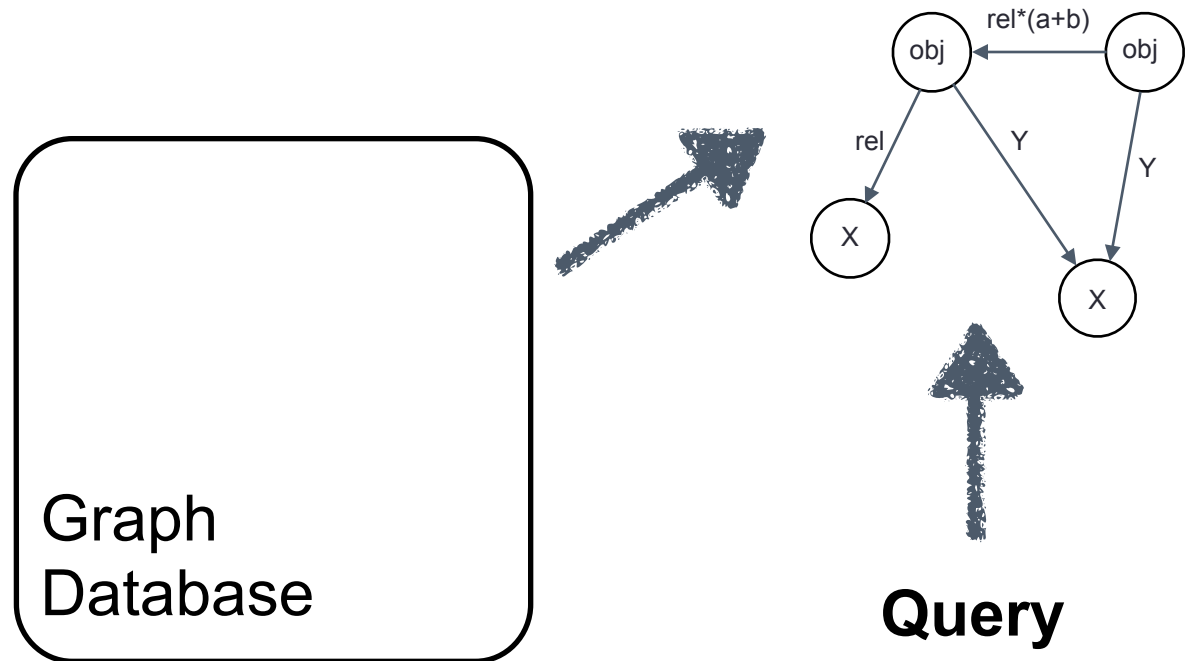
We study patterns with more expressive power

Important features:

- **Regular expressions** to represent complex **paths**
- **Node variables** to represent **objects** with same properties
- **Edge label variables** to represent **relations** with same properties

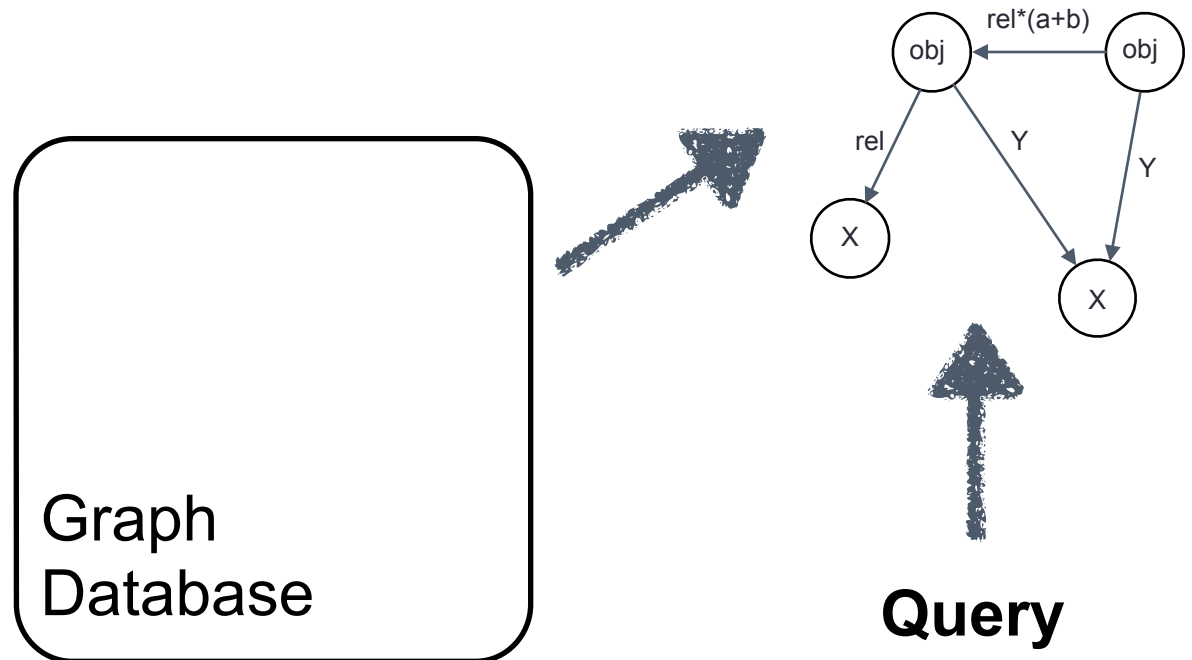
Semantics: **Patterns represent several graphs at once**

Querying graph patterns allows one to:



Querying graph patterns allows one to:

- Extract information from summarised graphs
- Deal with **incomplete or missing information** in graphs
- **Transform** or **integrate** several graph databases



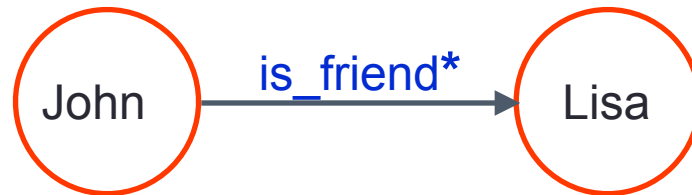
Contributions

We **define semantics** of patterns and querying patterns:

- Each pattern represents a set of graph databases
- Queries extract information common to all these graphs

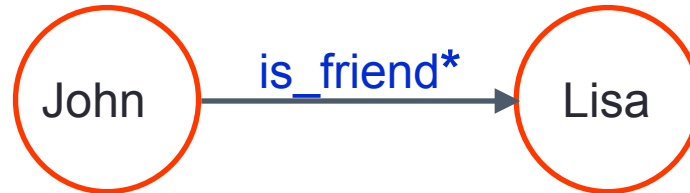
We define semantics of patterns and querying patterns:

- Each pattern represents a set of graph databases
- Queries extract information common to all these graphs



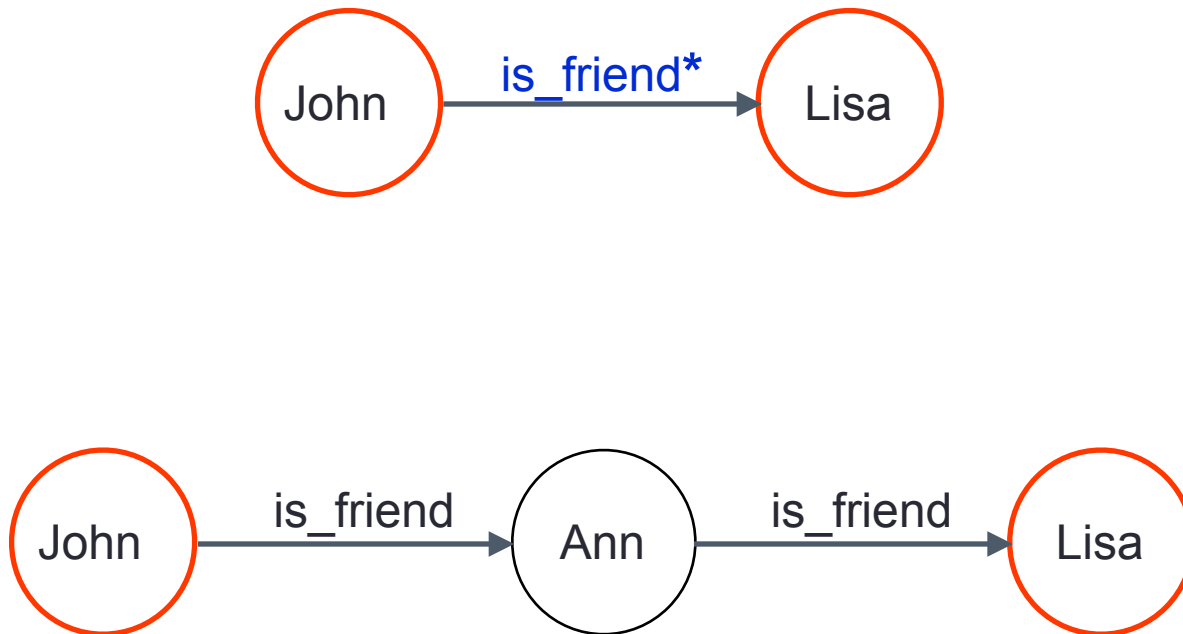
We define semantics of patterns and querying patterns:

- Each pattern represents a set of graph databases
- Queries extract information common to all these graphs



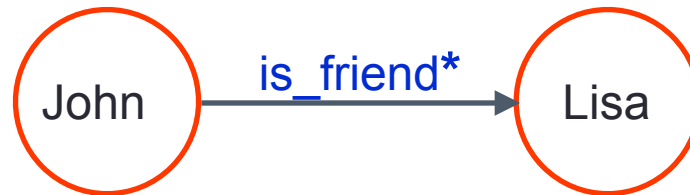
We define semantics of patterns and querying patterns:

- Each pattern represents a set of graph databases
- Queries extract information common to all these graphs



We define semantics of patterns and querying patterns:

- Each pattern represents a set of graph databases
- Queries extract information common to all these graphs



The information is that John is connected to Lisa,
via a chain of **is_friend**

Main conclusion

Tradeoff between expressiveness of patterns
and complexity of querying

Main conclusion

Tradeoff between expressiveness of patterns
and complexity of querying

- Querying gets **harder** as patterns get **more expressive**
- If one wants to use patterns specifying complex paths,
querying could become very costly

Main conclusion

Tradeoff between expressiveness of patterns
and complexity of querying

We can still do it

- Automata techniques gives us **heuristics**
- Finding **islands of tractability**
- Connections with constraint satisfaction problem for practical implementations

Main technique

Incomplete automata

- Standard finite automata with transitions partially defined

Main technique

Incomplete automata

- Standard finite automata with transitions partially defined
- Problems associated to querying graph patterns can be casted as standard language theoretic problems
- Applications beyond databases (e.g. program analysis)

Outline

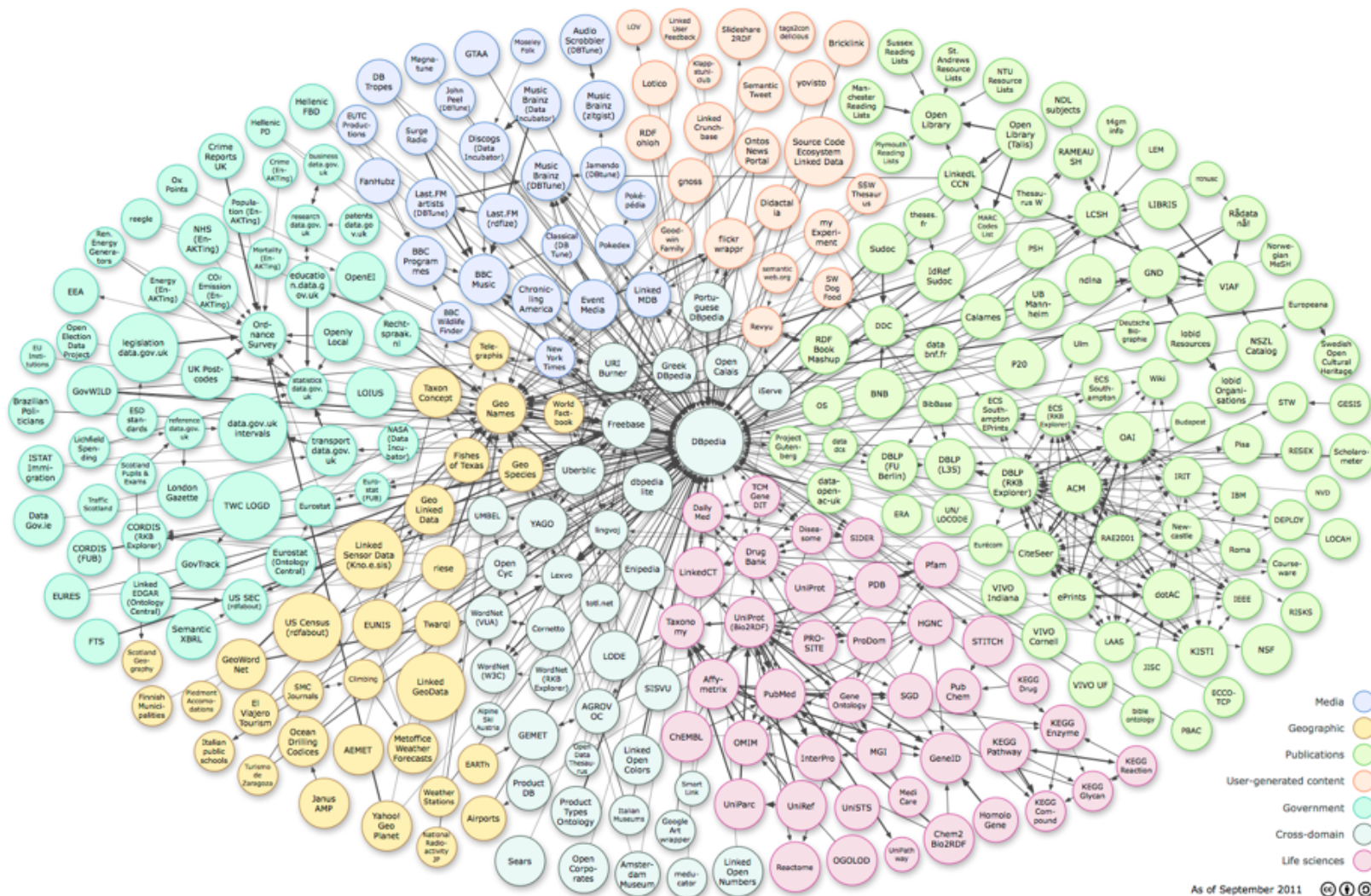
- Querying graph patterns
- Integrating multiple databases with underlying ontologies

Integrating databases with ontologies

Patterns are not the end of the story:

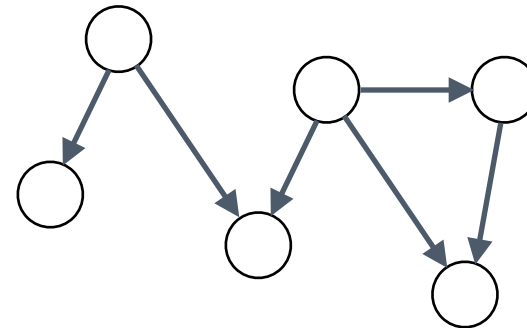
- In Linked Data databases are coupled with ontologies

As of September 2011



Ontology

\vdots
Airline \sqsubseteq Company
Company \sqsubseteq Organisation
Organization $\sqsubseteq \exists \text{ceo}$
ceo \sqsubseteq Person
 \vdots



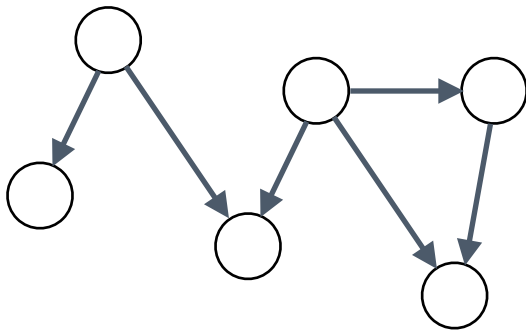
Graph Database

DBpedia

Knowledge
Base

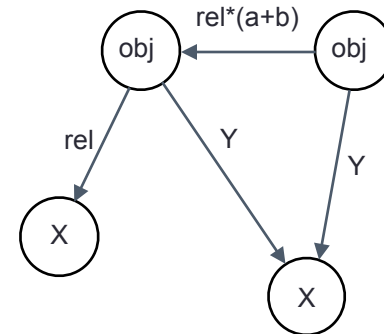
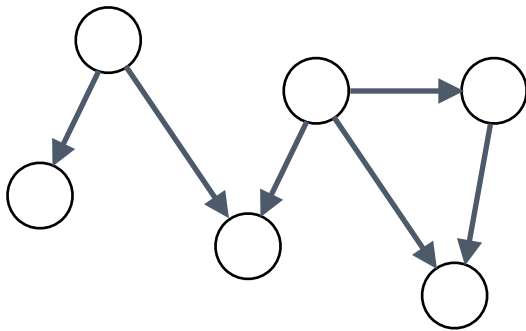
Integrating databases with ontologies

⋮
Airline \sqsubseteq Company
Company \sqsubseteq Organisation
Organization $\sqsubseteq \exists \text{ceo}$
ceo \sqsubseteq Person
⋮

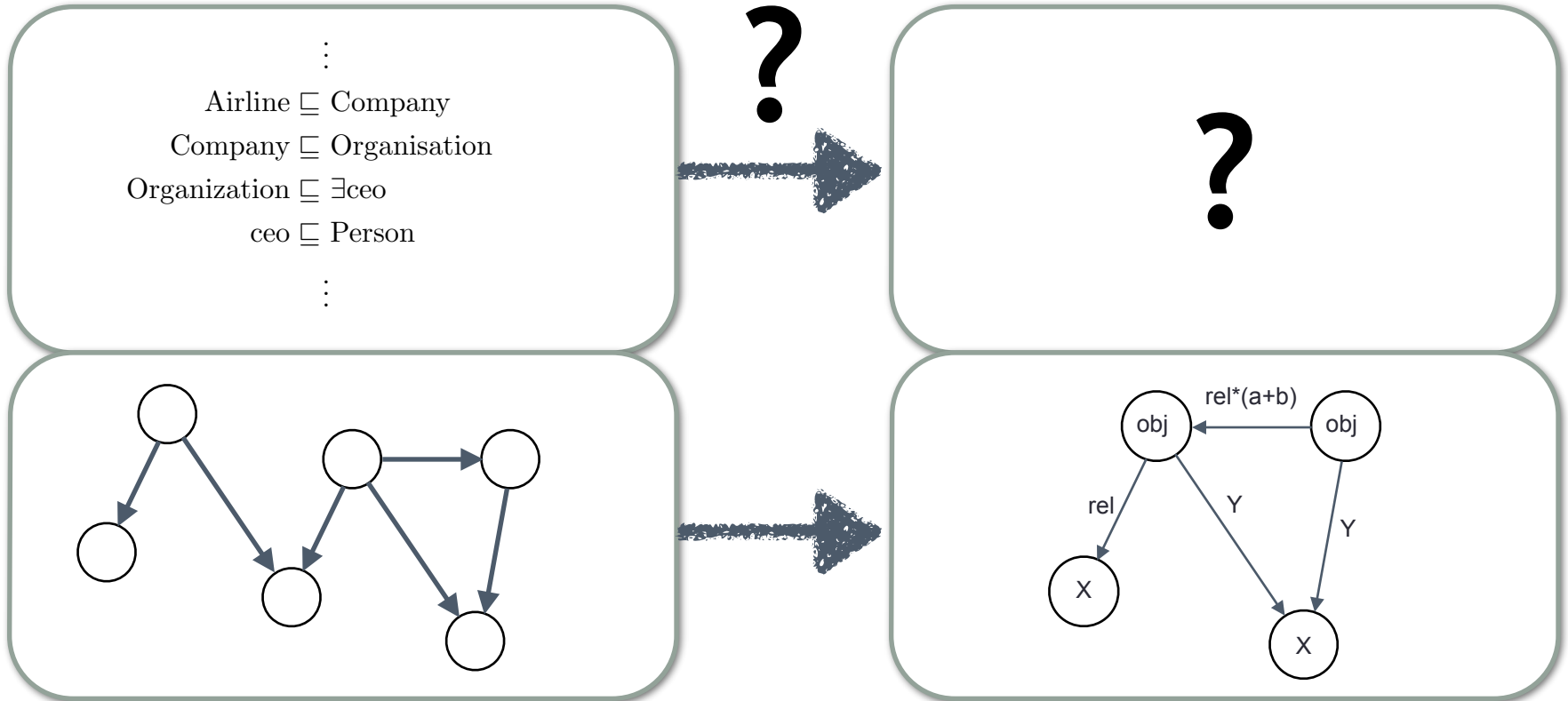


Integrating databases with ontologies

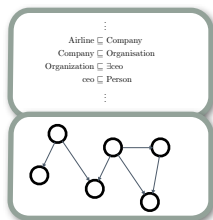
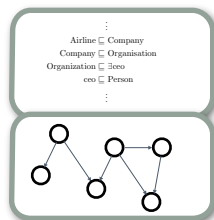
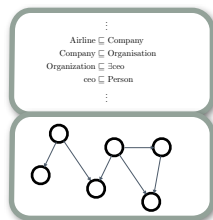
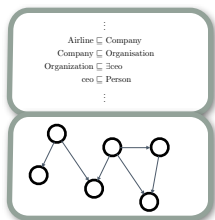
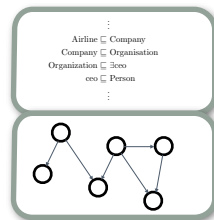
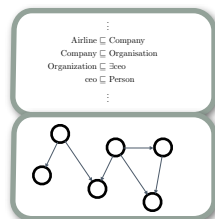
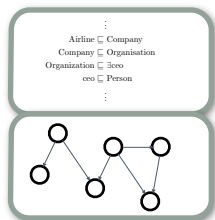
⋮
Airline \sqsubseteq Company
Company \sqsubseteq Organisation
Organization $\sqsubseteq \exists \text{ceo}$
ceo \sqsubseteq Person
⋮

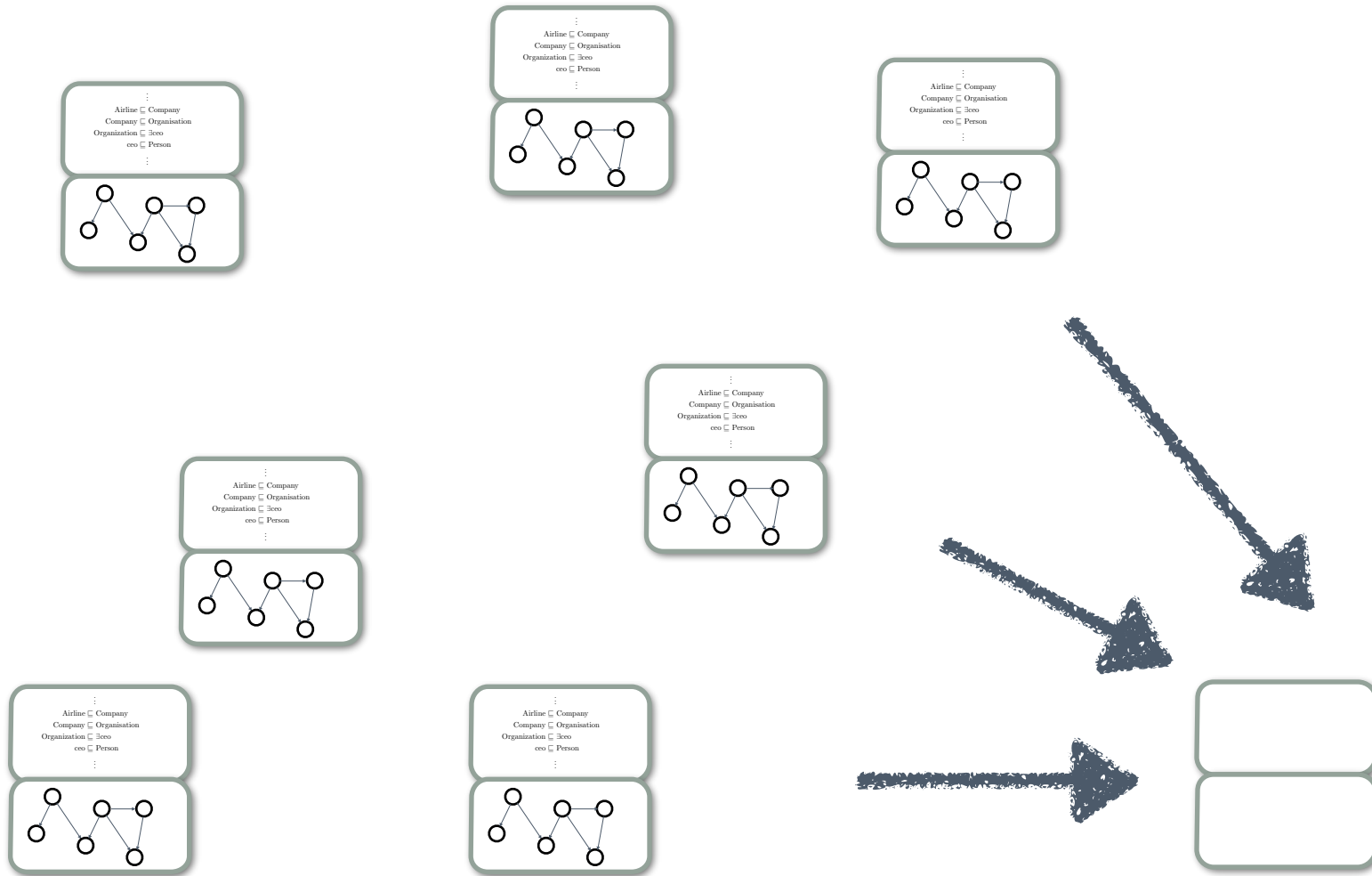


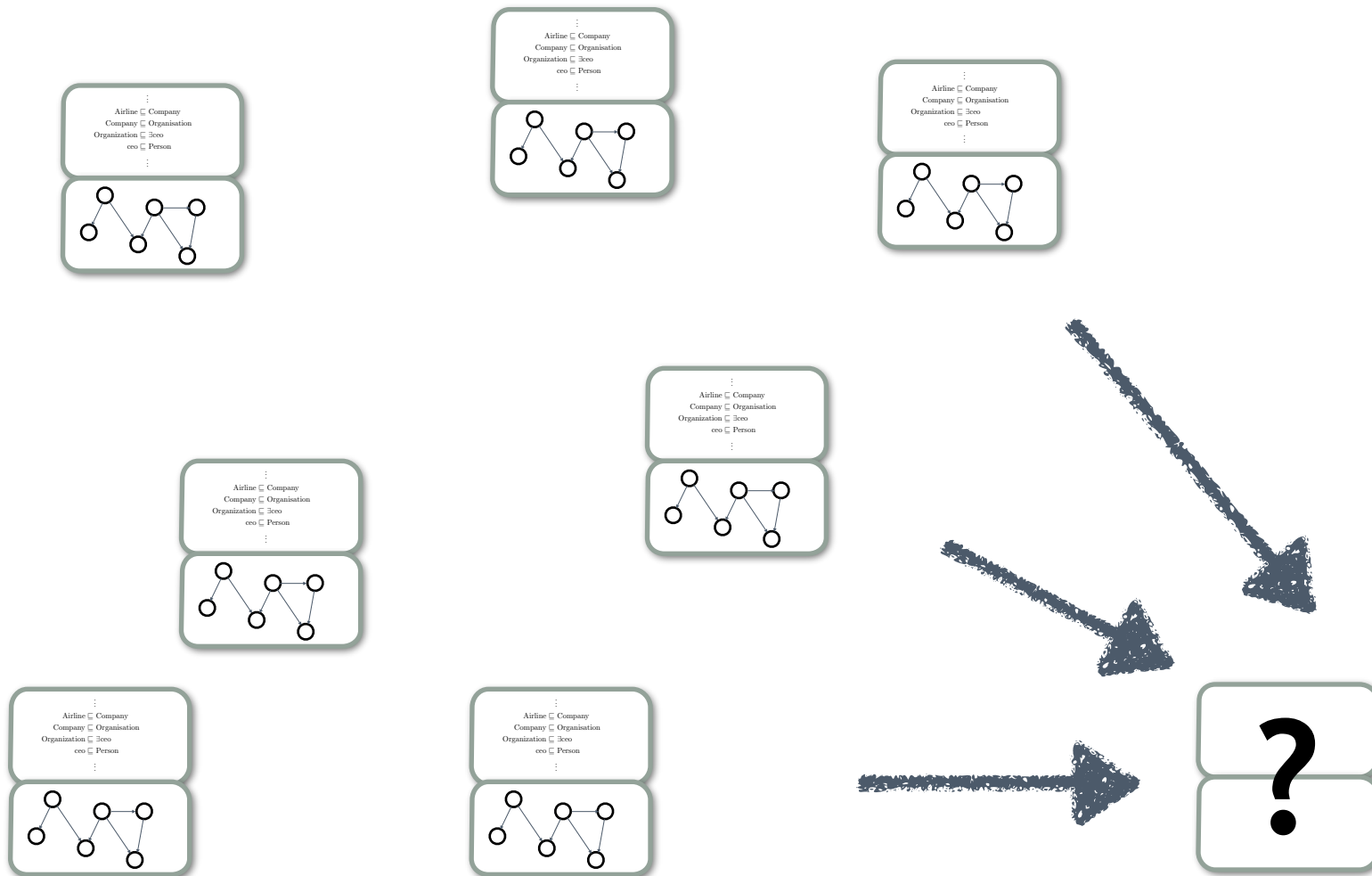
Integrating databases with ontologies



How do I summarise an ontology?







**How to integrate knowledge bases
into a single knowledge base?**

Contributions

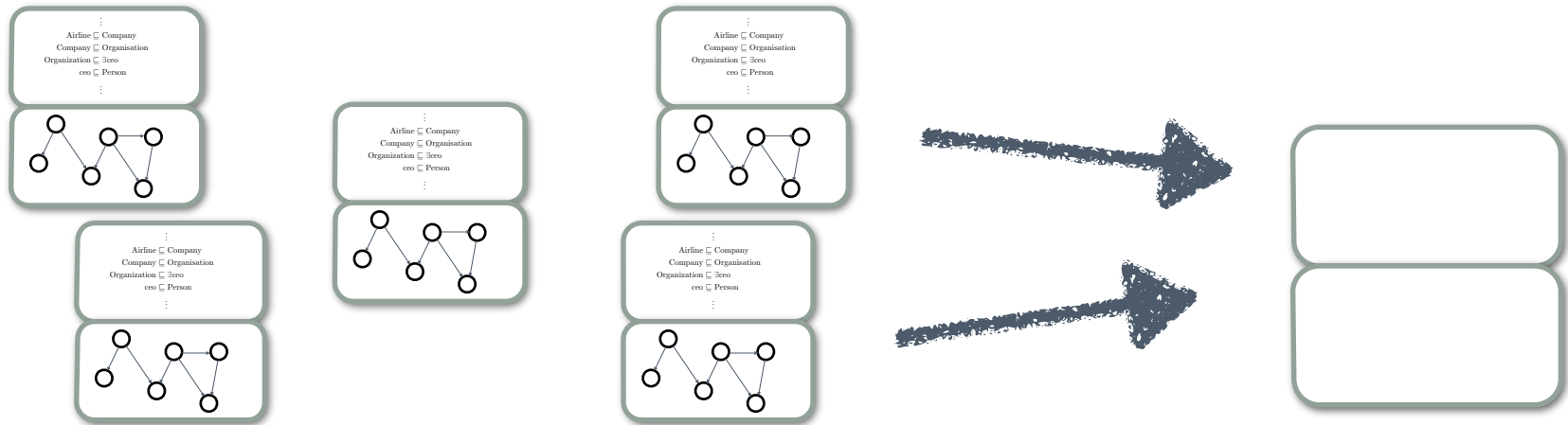
Formalised this problem:

Contributions

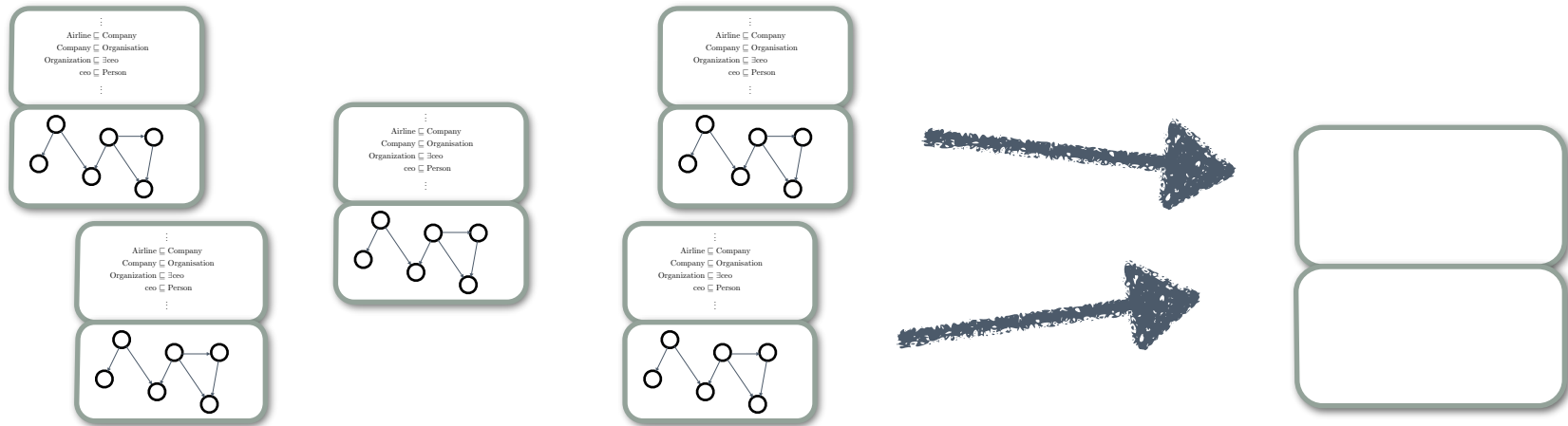
Formalised this problem:

- Theoretical framework
- Algorithms and Solutions for simple ontologies
- Framework can be used for other similar problems in databases such as dealing with incompleteness

Algorithms for solving this problem

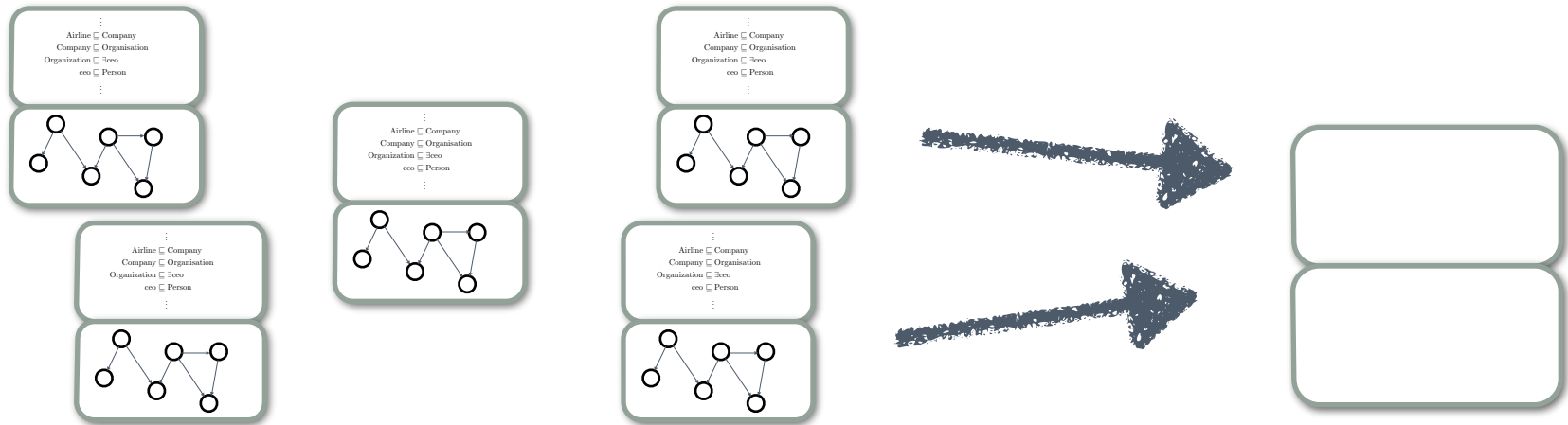


Algorithms for solving this problem



Can construct a new Knowledge Base that captures
information that is certain in the original data

Algorithms for solving this problem



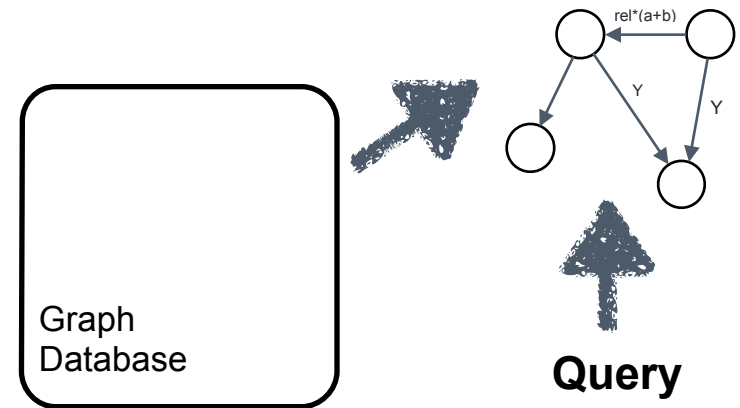
However, complex ontologies only allow for trivial solutions

- Need **notions of approximations** for ontologies

Conclusions

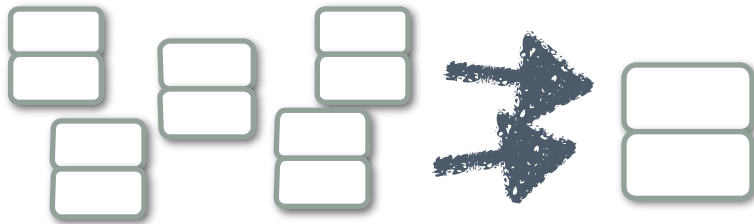
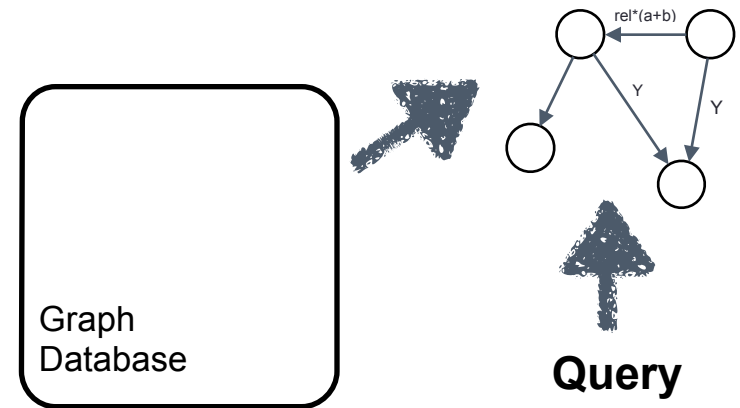
Conclusions

We now know how to
query complex graph patterns



Conclusions

We now know how to
query complex graph patterns



Can construct a new Knowledge Base that captures
information that is certain in the original data

Ongoing Work

Ongoing Work

We know how to query patterns, but how to **construct them**?

- Given a graph, create the **most representative** (small) **pattern**
- Perhaps coupled with an ontology

Ongoing Work

We know how to query patterns, but how to **construct them**?

- Given a graph, create the **most representative** (small) **pattern**
- Perhaps coupled with an ontology

Helping humans **understand** graph databases:

- DBpedia: all pages are stored in the same way
- can we represent this as a pattern
(perhaps with more features)?