Graph Analytics in the Big Data Era

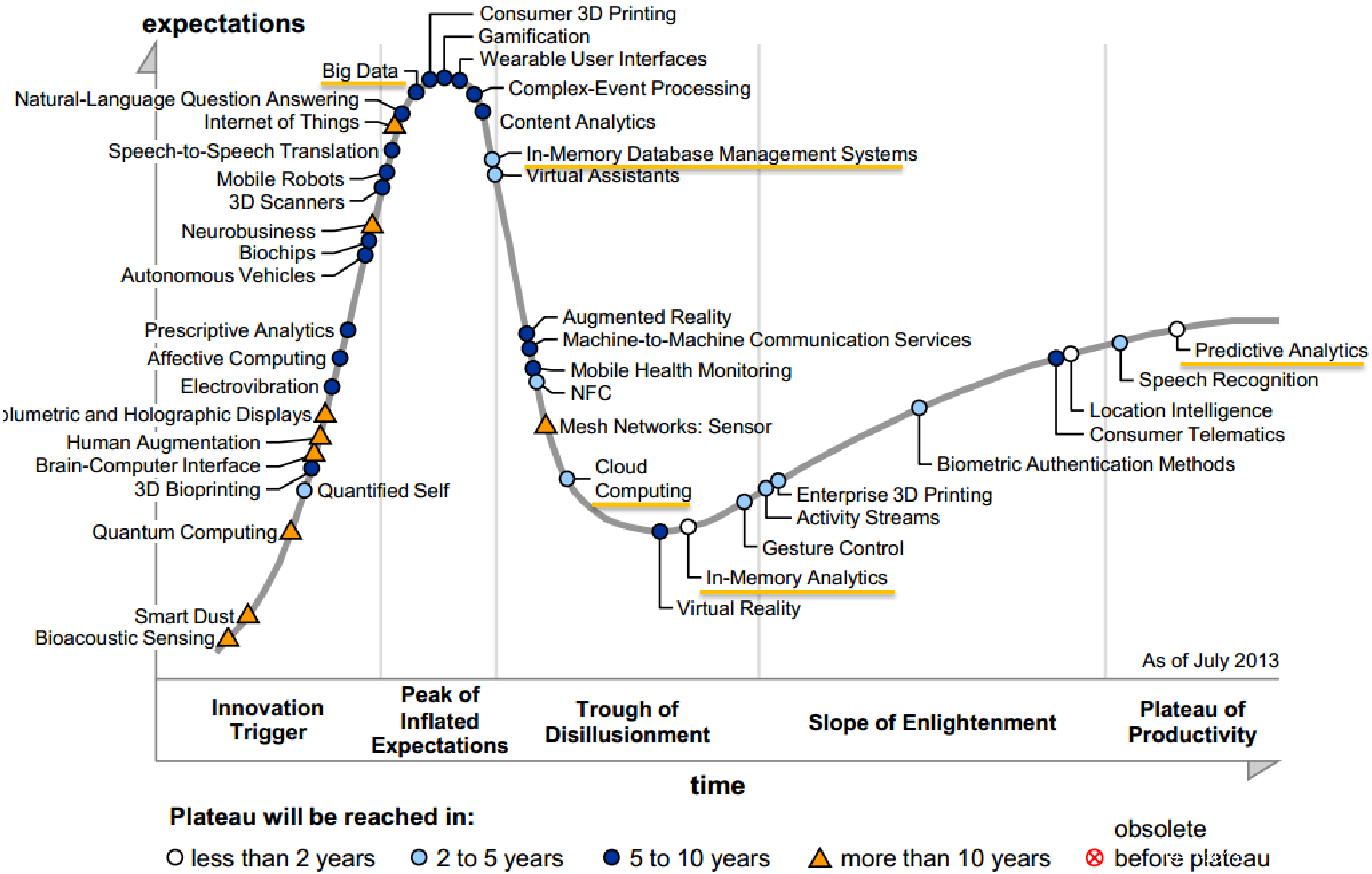
Yongming Luo, dr. George H.L. Fletcher

Web Engineering Group

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

What is really hot?



An old/new data model – graph data

- **Model entities and relations between entities**
- **Trending application space**
  - **Social network analysis (facebook, linkedin, …)**
  - **Bioinformatics (protein networks)**
  - **Recommendation (web graph, netflix, …)**
  - **Semantic Web (RDF data, Google knowledge graph)**

Technische Universiteit Eindhoven University of Technology

RDF as a data model for graph data

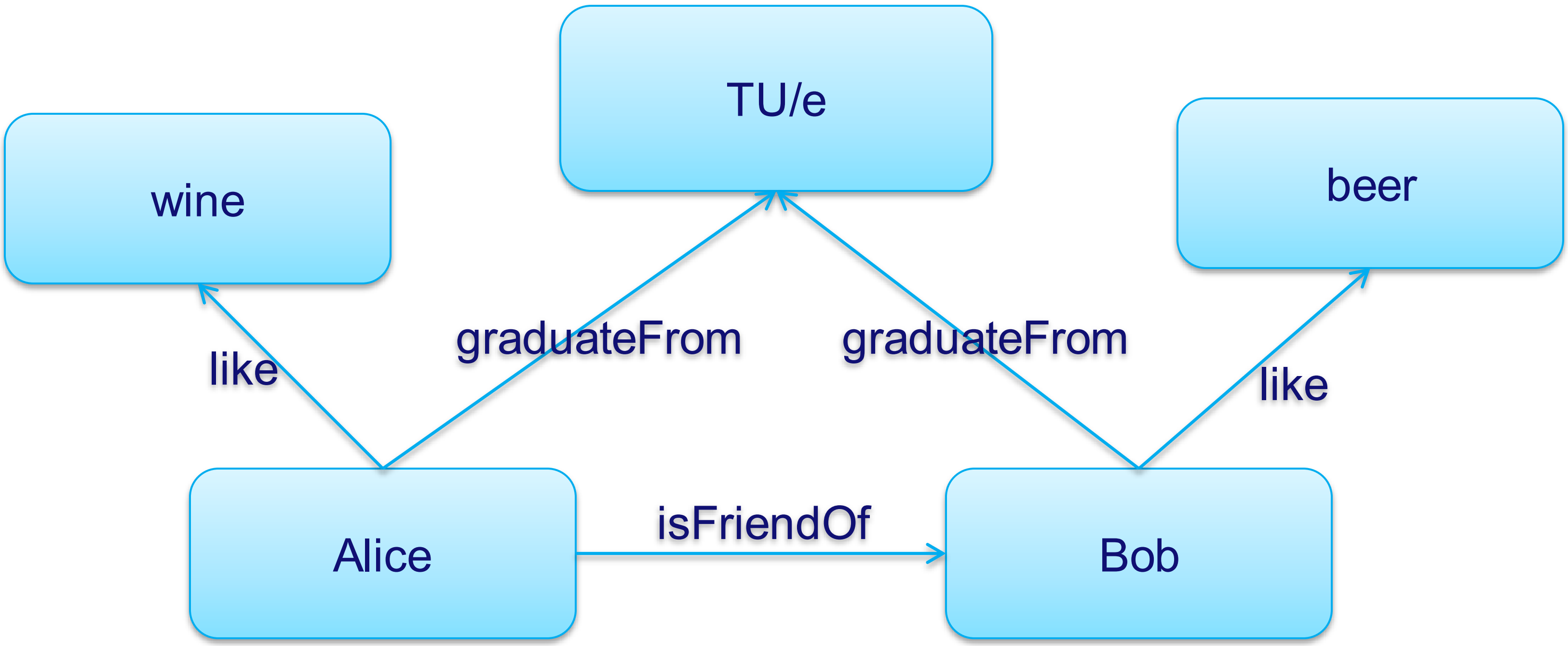
- **RDF Model (Resource Description Framework)**
  - **Describe “things (resources)” on the web**
  - **Part of the linked data vision, connect information on the web**
  - **Distributed way of managing *things***
  - **Schema-less feature**
  - **We will skip the strict format description for now**

RDF Graph Model and Format

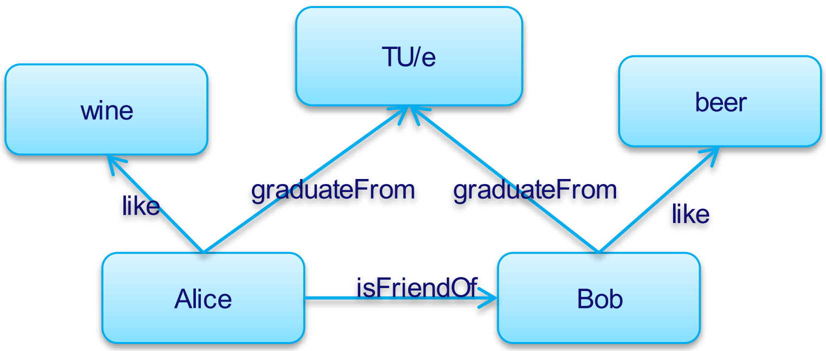
- Directed graph
- Triple format (subject, predicate, object)
- *Subjects* and *objects* are nodes/things in graph
- *Predicates* are edges
- (node, edge, node) format, nothing special

- No distinction between data and metadata
- Predicates can also be resources

RDF Graph Example

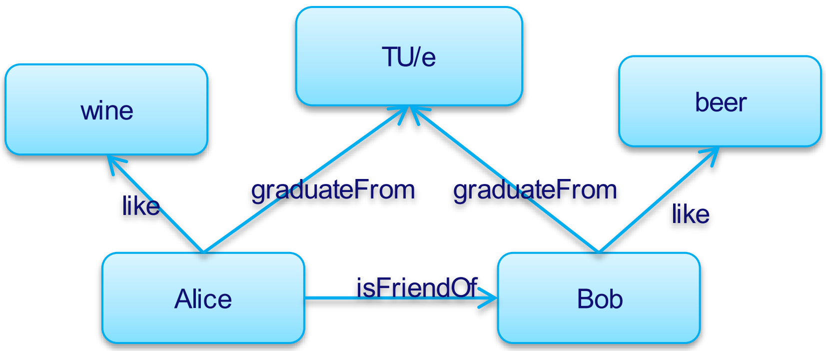


Example of Triples




```
<Alice>  <isFriendOf>          <Bob>
<Alice>  <like>               <wine>
<Alice>  <graduateFrom>       <TU/e>
<Bob>    <like>               <beer>
<Bob>    <graduateFrom>       <TU/e>
```

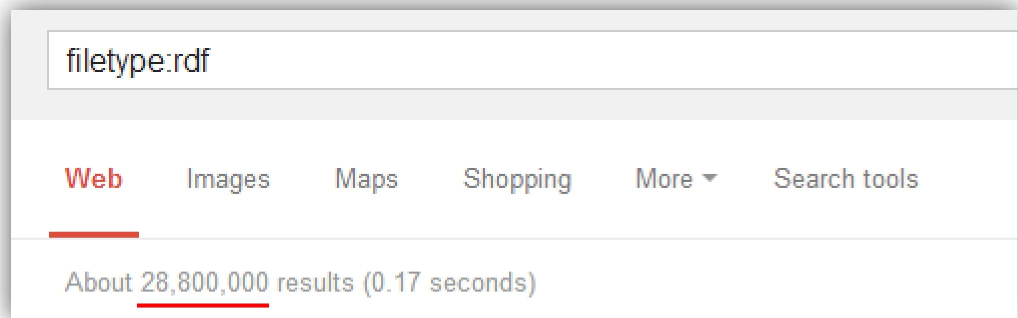
Example of Triples, expanded




```
<Alice>  <isFriendOf>      <Bob>
<Alice>  <like>            <wine>
<Alice>  <graduateFrom>    <TU/e>
<Bob>    <like>            <beer>
<Bob>    <graduateFrom>    <TU/e>
<like>         <isA>    <feeling>
<isFriendOf> <isA>     <relation>
```

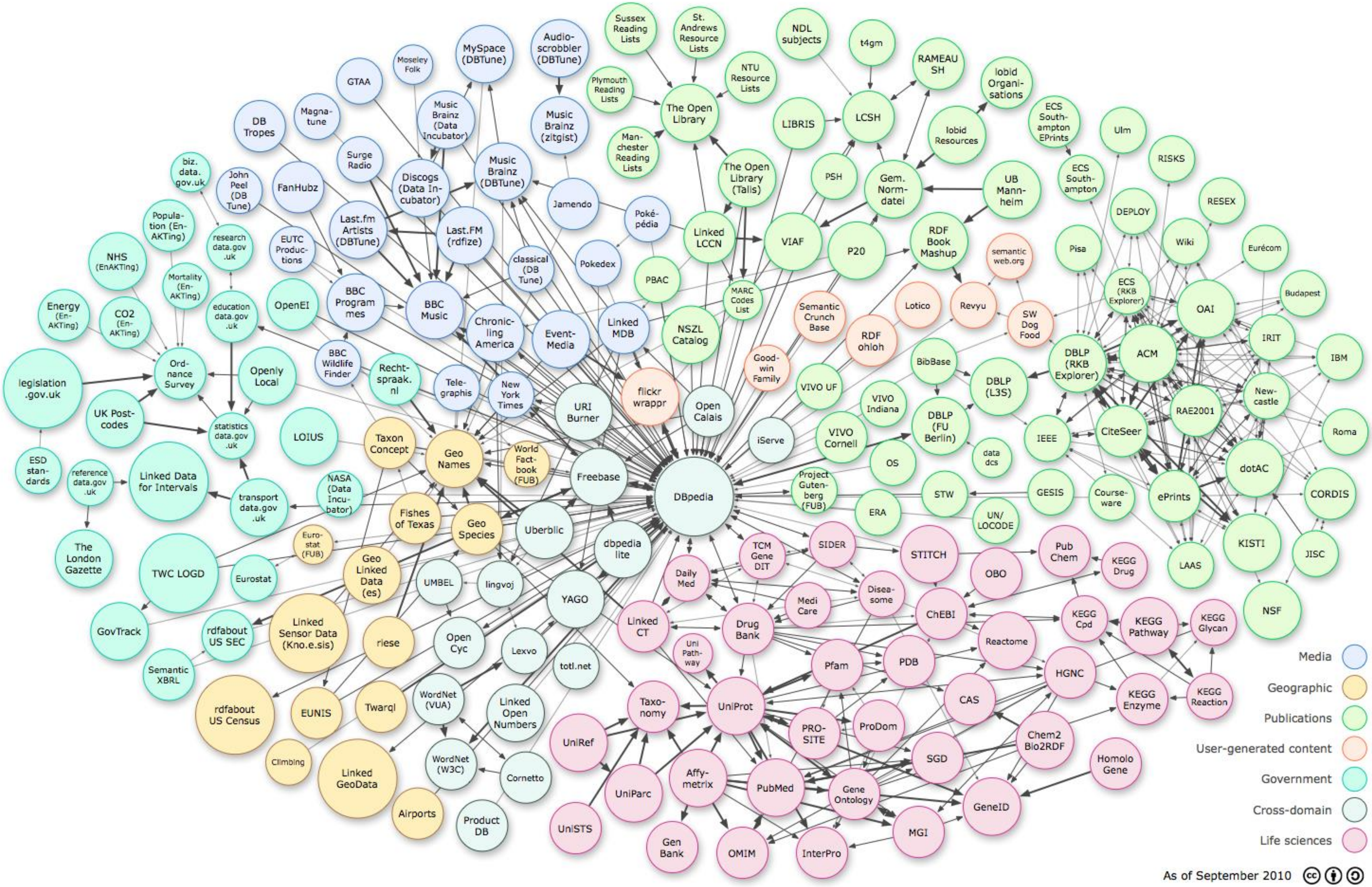
Data Explosion in the (RDF) world

- **How much data are we talking about?**
  - **Simply type filetype:rdf in google, 28M documents found**



  - **Billion Triple Challenge at ISWC**
  - **Easily reach a Trillion triples in commercial systems**
  - **More data join the open data project to connect datasets, and the famous graph (next page)**

Technische Universiteit Eindhoven University of Technology



http://richard.cyganiak.de/2007/10/lod/

TU/e Technische Universiteit Eindhoven University of Technology

Facebook graph

Graph analytics

- What to do with graphs?
  - Pattern matching
    - graph query languages
  - Graph algorithms
    - Classical algorithms
    - PageRank-style algorithms

Pattern Matching - SPARQL Query for RDF

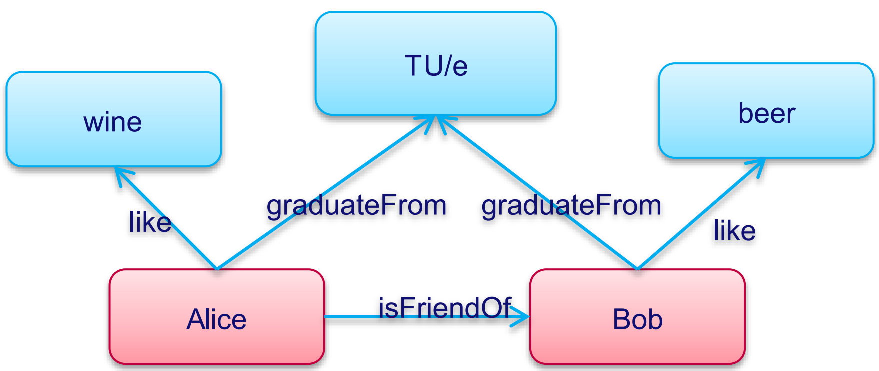
- **The query language for RDF data**
- **Similar to SQL for relational database**
- **Similar grammar, select, where, filter clauses and more**
- **Example**

```
select ?a ?b where {
   ?a <isFriendOf>    ?b .
   ?a <graduateFrom> ?c .
   ?b <graduateFrom> ?c .
}
```

Technische Universiteit **Eindhoven** University of Technology

SPARQL Example

```
select ?a ?b where {
  ?a <isFriendOf>     ?b .
  ?a <graduateFrom> ?c .
  ?b <graduateFrom> ?c .
}
```



Two people who are friends and graduate from the same university

SPARQL Query can be Complex

```
SELECT ?gn ?fn WHERE {
    ?gn <givenNameOf>                ?p.
    ?fn <familyNameOf>               ?p.
    ?p  <type>                       "scientist";
        <bornInLocation>             ?city;
        <hasDoctoralAdvisor>         ?a.
    ?a  <bornInLocation>             ?city2.
    ?city  <locatedIn>               "Switzerland".
    ?city2 <locatedIn>               "Germany".
}
```

rdf3x / Yago / A1.sparql

A few more words on query language

- Essentially ad-hoc graph algorithm execution
- Pattern matching as the backbone, with possibly many features added
  - E.g., regular expression, keyword search, aggregation
- Some special cases get special treatment
  - Triangle counting – community detection, graph measurement

Graph Algorithms

- Breadth First Search
- Single Source Shortest Path
- Bipartite Matching
- PageRank, SimRank and more
  - So called diffusion based techniques

Technische Universiteit **Eindhoven** University of Technology

Where are we

- Graph model
- Operations on graph
  - Pattern matching (queries) -- indexes
  - Algorithms -- platforms

Indexes to accelerate query processing

- **Value-based indexes**
  - **Relational approaches**
  - **Document-oriented approaches**
- **Structural indexes**
  - **Seeqr**
  - **Frequent patterns, …**

Storing and Indexing Massive RDF Datasets. Yongming Luo, Francois Picalausa, George H. L. Fletcher, Jan Hidders and Stijn Vansummeren. In: De Virgilio, R., et al. (eds.) Semantic Search over the Web, Data-Centric Systems and Applications, pp. 31–60. Springer, Heidelberg (2012).

TU/e Technische Universiteit Eindhoven University of Technology

Relational approaches, RDF as an example

- **Treat triples as a three-column table**
  - **Relational DB, row store, column store**

Subject	Predicate	Object
<Alice>	<isFriendOf>	<Bob>
<Alice>	<like>	<wine>
<Alice>	<graduateFrom>	<TU/e>
<Bob>	<like>	<beer>
<Bob>	<graduateFrom>	<TU/e>
…	…	…

- **One entity one row**

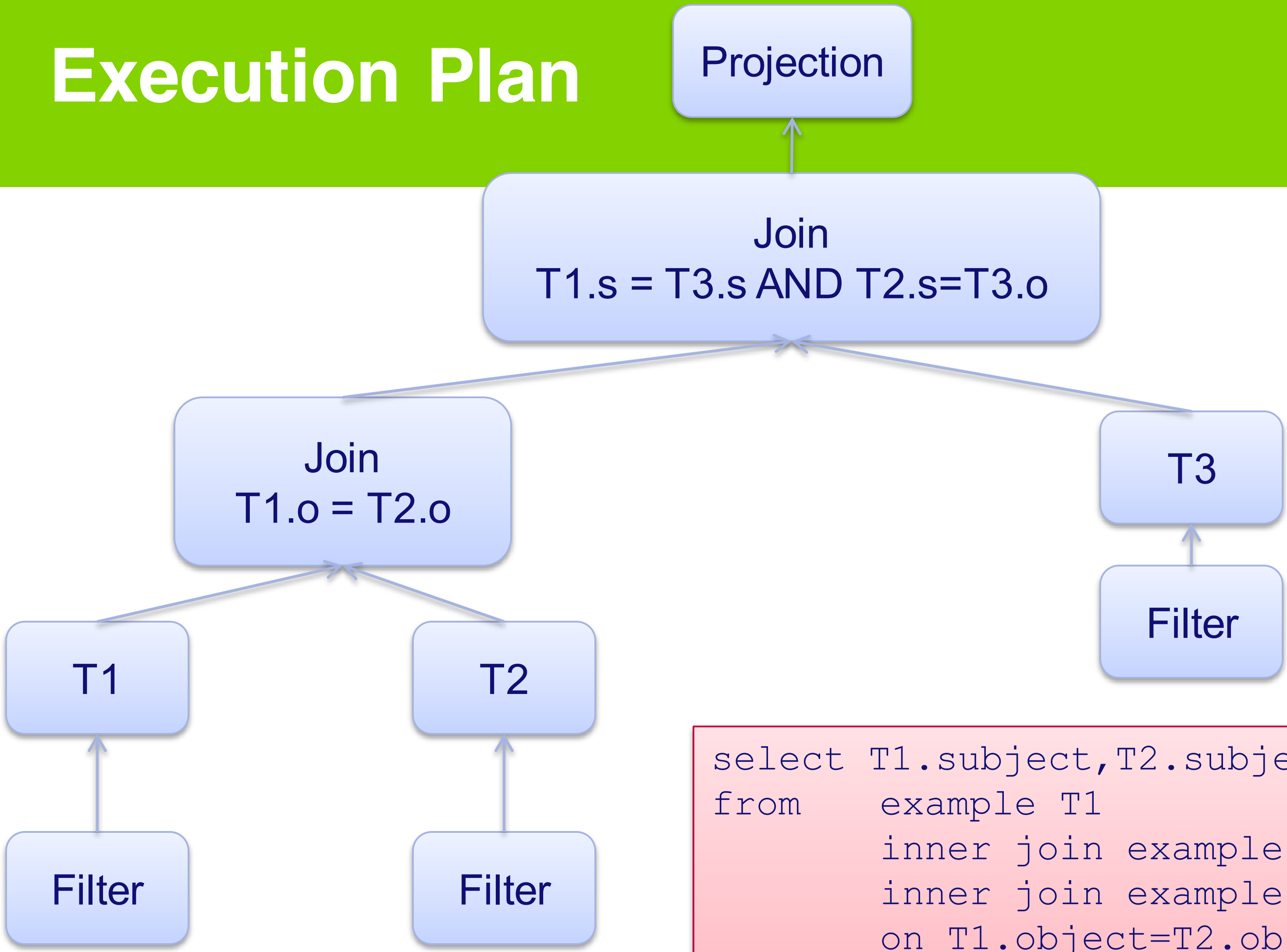
Relational Approaches – Query Processing

- **For relational DB**
- **SPARQL -> SQL**

```
select ?a ?b where {
  ?a <isFriendOf>    ?b .
  ?a <graduateFrom> ?c .
  ?b <graduateFrom> ?c .
}
```

```
select T1.subject,T2.subject
from  example T1
      inner join example T2
      inner join example T3
      on T1.object = T2.object
      AND T1.subject = T3.subject
      AND T2.subject = T3.object
where T1.predicate = 'graduateFrom'
      AND T2.predicate = 'graduateFrom'
      AND T3.predicate = 'isFriendOf';
```

Execution Plan




```
select T1.subject,T2.subject
from   example T1
       inner join example T2
       inner join example T3
       on T1.object=T2.object
       AND T1.subject=T3.subject
       AND T2.subject=T3.object
where  T1.predicate='graduateFrom'
       AND T2.predicate='graduateFrom'
       AND T3.predicate='isFriendOf';
```

Relational DB is not enough

- **Problems**
  - **Reduce data size -> ID mapping**

String	ID
<Alice>	1
<Bob>	2
<like>	3
…	…

  - **How to build indexes?**
  - **Too many self-joins**
  - **Several alternative SQLs, which one to choose?**

Native Approaches - RDF-3X

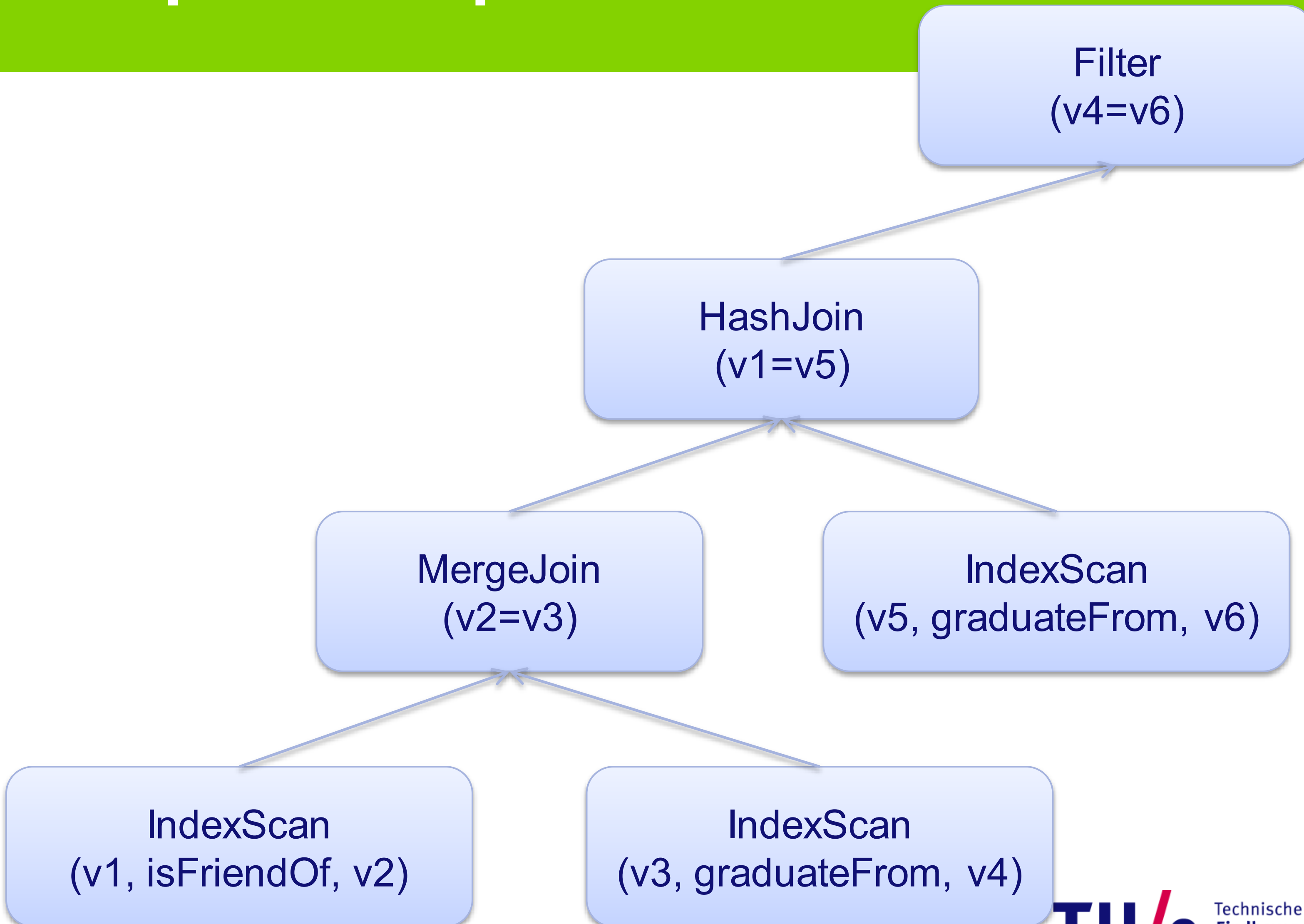
- **B-Tree index**
- **Index several (if not all) permutations of triples**
  - **(s, p, o) (p, s, o) (o, s, p) (s, o, p) …**
- **prefer merge join rather than hash join**

```
select ?a ?b where {
  ?a <isFriendOf>    ?b .
  ?a <graduateFrom> ?c .
  ?b <graduateFrom> ?c .
}
```

Predicate	Subject	Object
<isFriendOf>	<Alice>	<Bob>
<like>	<Alice>	<wine>
<like>	<Bob>	<beer>
<graduateFrom>	<Alice>	<TU/e>
<graduateFrom>	<Bob>	<TU/e>
…	…	…

TU/e Technische Universiteit Eindhoven University of Technology

One possible plan from RDF-3X



More techniques

- Query optimization
  - Statistics, query history, cache, all information helps
- Compression
  - Standard compression techniques
  - Works better for column store
- Update
  - Save changes first, merge them later

Technische Universiteit
Eindhoven
University of Technology

One Entity One Row

- In the spirit of E-R model, or adjacency list of graphs
- Reduce self-joins

Entity	isFriendOf	like	graduateFrom
<Alice>	<Bob>	<wine>	<TU/e>
<Bob>		<beer>	<TU/e>
<wine>			
<beer>			
<TU/e>			
	…	…	…

TU/e Technische Universiteit Eindhoven University of Technology

One Entity One Row - Cons

- Null values
- Multi-value property
- Too many properties
- Schema required
- Schema update

Entity	isFriendOf	like	graduateFrom
<Alice>	<Bob>	<wine>	<TU/e>
<Bob>		<beer>	<TU/e>
<wine>			
<beer>			
<TU/e>			
	…	…	…

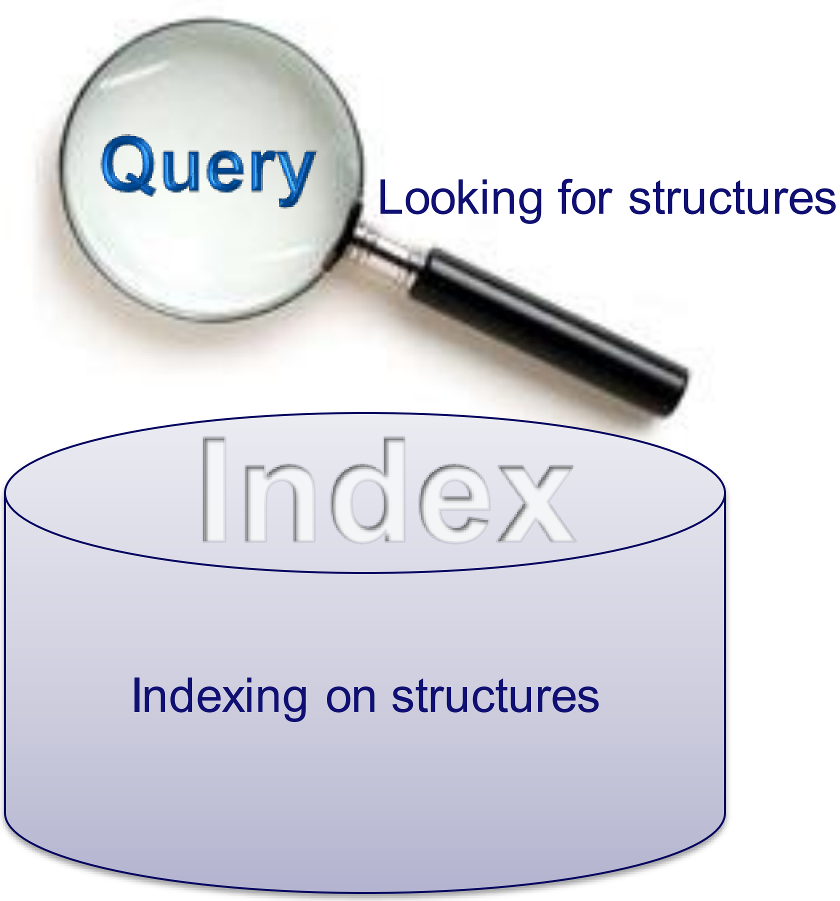
Structural Index - Motivation

- **Value indexes are good, but there is more regularity/structure we can get from data**
- **Query/index mismatch**

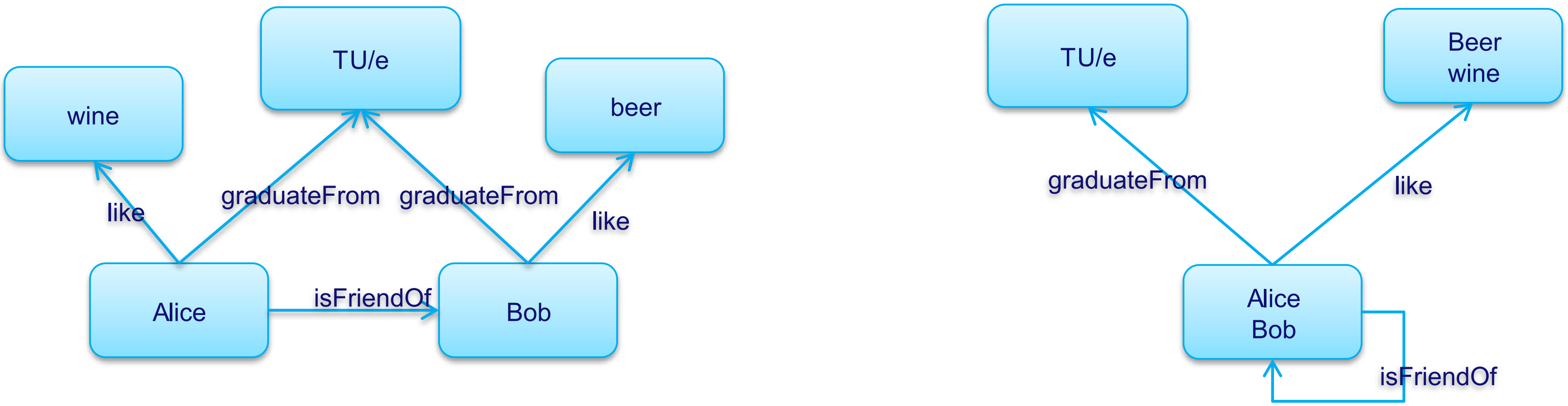
Looking for structures

Structural Index

- **Preserve structures in index**
- **Give the queries what they are looking for, not less, not more**



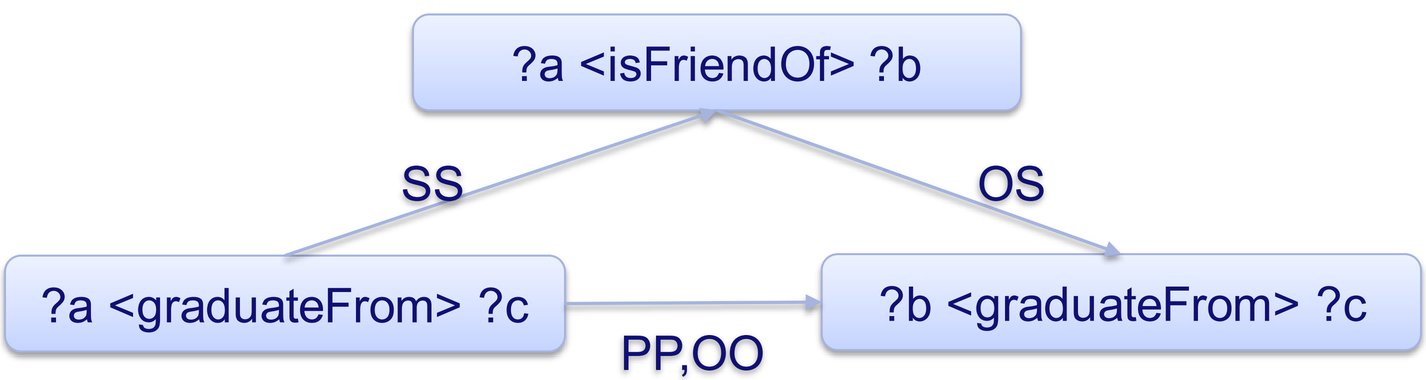
It will be better if we have something like



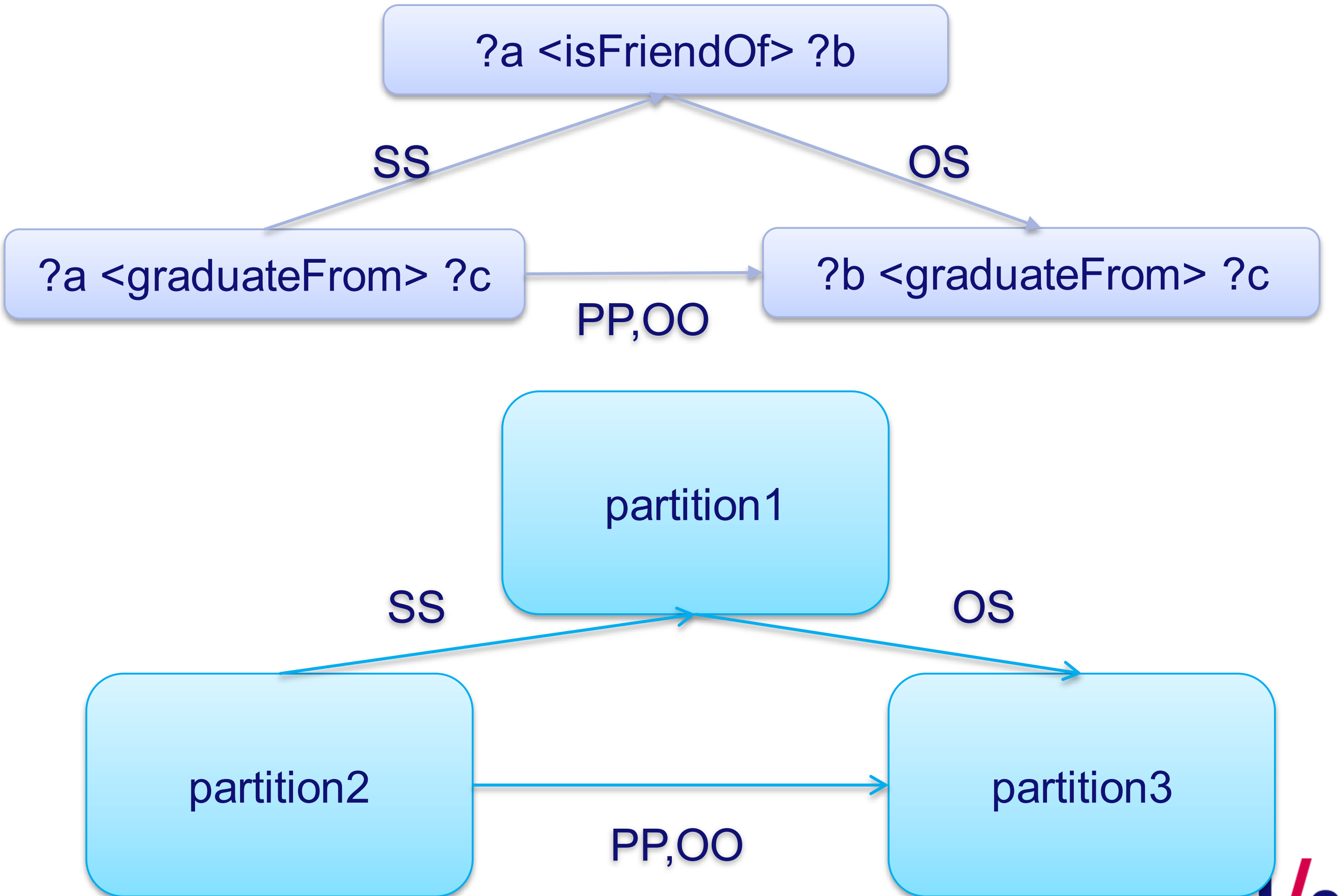
Structural Index - Example

```
select ?a ?b where {
  ?a <isFriendOf>    ?b .
  ?a <graduateFrom> ?c .
  ?b <graduateFrom> ?c .
}
```

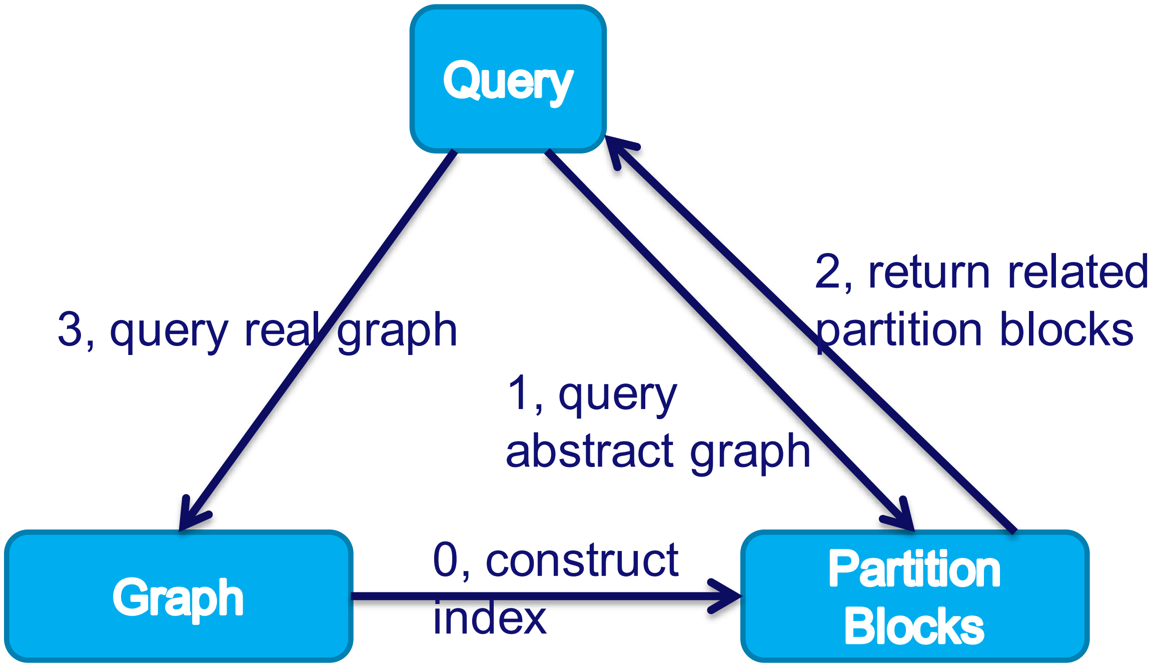
If two triples share a subject
Draw an SS edge between them
Same for SO, OS, SP, PS, …



Ideal case


?a <isFriendOf> ?b
SS
OS
?a <graduateFrom> ?c
?b <graduateFrom> ?c
PP,OO
partition1
SS
OS
partition2
partition3
PP,OO

Workflow



A Structural Approach to Indexing Triples. Francois Picalausa, Yongming Luo, George H. L. Fletcher, Jan Hidders and Stijn Vansummeren. ESWC 2012, Heraklion, GR. LNCS 7295, pp. 406–421, 2012, Springer-Verlag Berlin Heidelberg.

How to build partition blocks (structural index)?

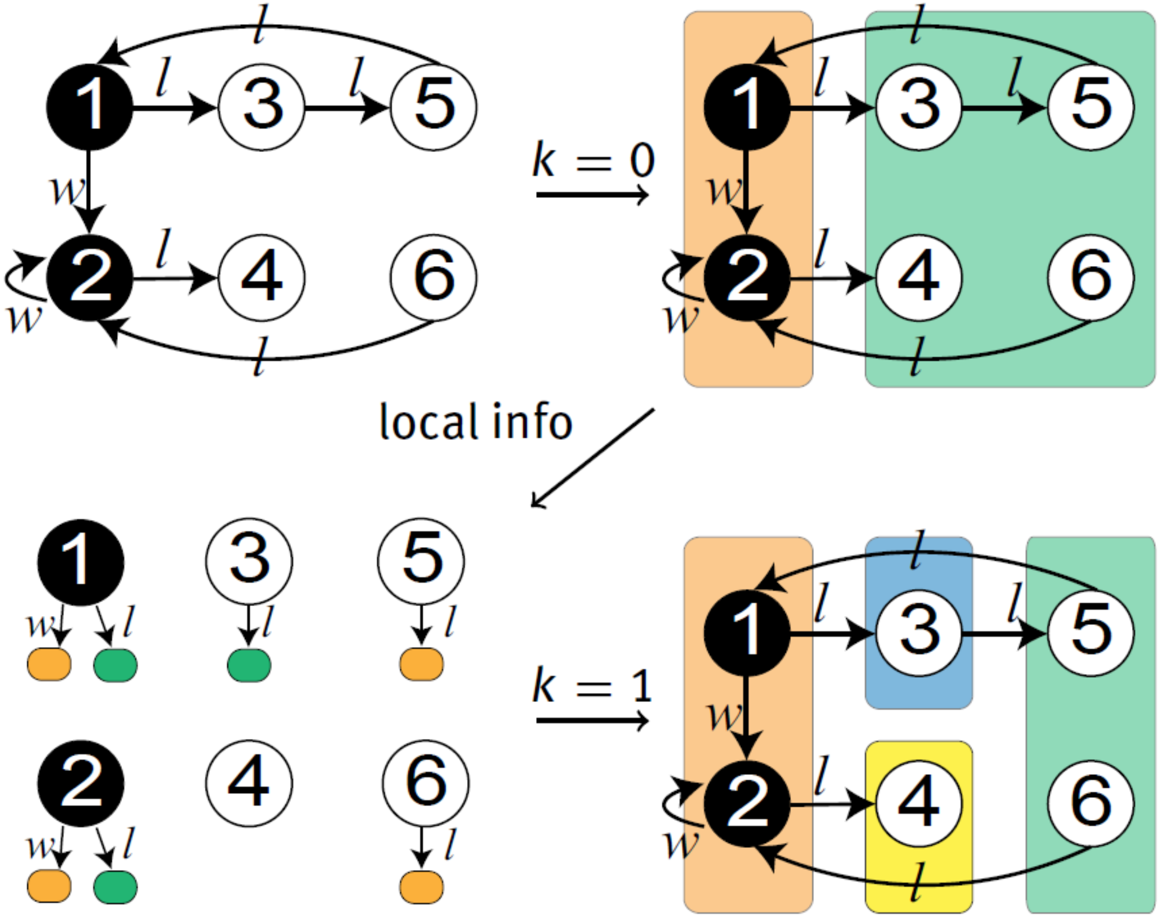
- E.g., according to k-bisimulation, denoted $\approx^k$

$x \approx^k y$ when the following holds:

- $x \approx^0 y$ if the node labels of x and y are the same
- If $x \rightarrow x'$, then there is some $y \rightarrow y'$, such that $x' \approx^{k-1} y'$
- If $y \rightarrow y'$, then there is some $x \rightarrow x'$, such that $x' \approx^{k-1} y'$

Algorithm for k-bisimulation computation

- Create a signature for each node in the graph
- Nodes are partitioned by their signature values



Where are we

- Graph model
- Operations on graph
  - Pattern matching (queries) -- indexes
  - Algorithms -- platforms

Platforms for (graph) algorithms

- **Single machine, out-of-core**
  - **I/O-efficient algorithms**
  - **GraphChi**
- **Shared-nothing architecture**
  - **MapReduce (Hadoop)**
  - **Pregel/GraphLab/Giraph**
  - **More to come and play**

Project Ideas

Project Idea: Bisimulation-friendly Big Graph Generator

- **In recent research, we see that power-law distribution in bisimulation results are not preserved in current synthetic graph generators.**
- **We want to change that.**
- **The task includes:**

  0. Pick a distributed programming framework, map-reduce (or other distributed framework as you like, spark, hyracks), get comfortable with the programming environment.

  1. Design an algorithm that generates big graphs (billions of edges) that are

    1.1. power-law

    1.2. bisimulation friendly

    ***1.3 other properties, such as small diameter**

  2. Test, compare with other approaches, both in efficiency and in quality (e.g., socialbench, graph500)

TU/e Technische Universiteit Eindhoven University of Technology

Project Idea: External-Memory Giraph

- Giraph is an Apache copy of Pregel, a BSP-like computation framework for distributed environment. A very new and hot platform to play with.
- It is proved that BSP algorithms can be simulated in an external memory environment in an efficient way.
- In this research, we want to use external memory environment as a backend for Giraph, enabling its efficiency on single machine.
- The task includes:
    1. try out Giraph
    2. write a few classical algorithms in Giraph
    3. write the external memory backend, API-compatible
    4. compare the result on medium to large graph datasets (~1billion edges)

Other related topics

- **If you have some related ideas in mind, just come and talk to me.**

Thank you!

Q&A

y.luo@tue.nl