DEPARTMENT OF
COMPUTER
SCIENCE

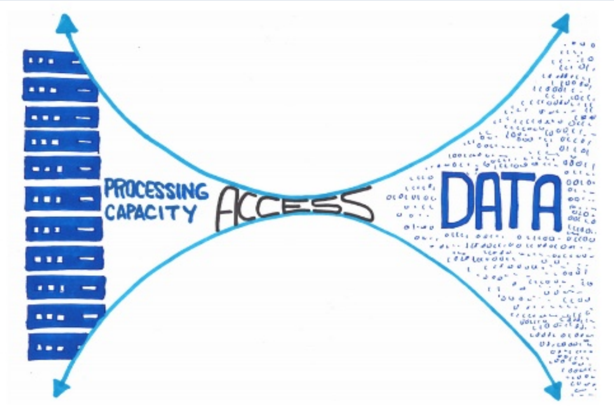
UNIVERSITY OF
OXFORD

Ontology Based Data Access (OBDA)

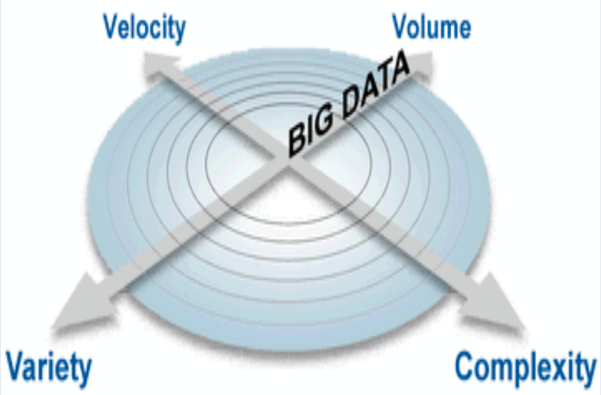
Ian Horrocks
Information Systems Group
Department of Computer Science
University of Oxford

Motivation

- Huge **quantity** of data increasing at an exponential rate
- Identifying & accessing **relevant** data is of critical importance



- Handling data **variety & complexity** often turns out to be main challenge
- **Semantic Technology** can seamlessly integrate heterogeneous data sources



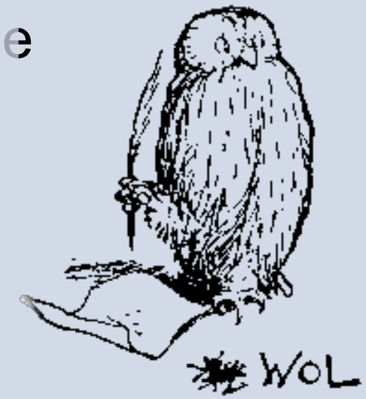
Semantic Technology

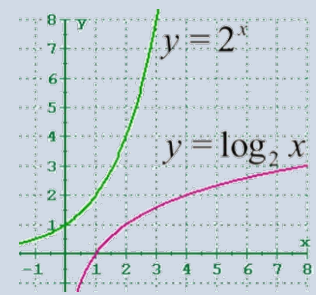
Rich **conceptual schemas** used to integrate heterogeneous sources

- **User Centric**
  - Schema modelled according to user intuitions
  - Independent of physical structure/storage of data
- **Declarative**
  - Improved understandability
  - Easier design, maintenance and evolution
- **Logic-based semantics**
  - Precise and formally specified meaning
  - Machine processable
- **Used at both design and query time**
  - Check validity and consequences of design
  - Easier query formulation and enriched query answers

Semantic Technology: Scalability Challenge

- Conceptual sche… …se OW… …ology language
- OWL based o… …cr… …gic S… …OIQ
  - ✔ Declarati…
  - ✔ Clear se… …ntic…
  - ✔ Well und… …too… …c… …ertie…
    (e.g., alg… …m… …d… …p… …exit…
  - ✗ N2ExpTim… …mplete co… …ined c… …y
  - ✗ NP-hard data… …plexity (-v- AC^0… …atabases)





How can we provide (empirically) scalable query answering?

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

OWL Profiles

OWL 2 defines language subsets, aka **profiles** that can be “more simply and/or efficiently implemented”

- **OWL 2 QL**
  - Based on DL-Lite
  - Efficiently implementable via rewriting into relational queries (OBDA)


Pipelines from oil facilities?

$$
\begin{aligned}
Q(?x) \leftarrow\ & (?x, \text{rdf:type}, \text{:Pipeline}) \wedge \\
& (?x, \text{:fromFacility}, ?y) \wedge \\
& (?y, \text{rdf:type}, \text{:OilFacility})
\end{aligned}
$$

$$
\begin{aligned}
Q(?x) \leftarrow\ & (?x, \text{rdf:type}, \text{:Pipeline}) \wedge \\
& (?x, \text{:fromFacility}, ?y) \wedge \\
& (?y, \text{rdf:type}, \text{:OilFacility})
\end{aligned}
$$

(:p1, rdf:type, :Pipeline)
(:p1, :fromFacility, :f1)
(:f1, rdf:type, :OilFacility)
(:p2, rdf:type, :OilPipeline)
(:p2, :fromFacility, :f2)
(:f2, rdf:type, :OilFacility)
(:p3, rdf:type, :OilPipeline)

$$
\begin{aligned}
Q(?x) \leftarrow\ & (?x, \text{rdf:type}, \text{:Pipeline}) \wedge \\
& (?x, \text{:fromFacility}, ?y) \wedge \\
& (?y, \text{rdf:type}, \text{:OilFacility})
\end{aligned}
$$

(:p1, rdf:type, :Pipeline)
(:p1, :fromFacility, :f1)
(:f1, rdf:type, :OilFacility)
(:p2, rdf:type, :OilPipeline)
(:p2, :fromFacility, :f2)
(:f2, rdf:type, :OilFacility)
(:p3, rdf:type, :OilPipeline)

$$
\begin{aligned}
Q(?x) \leftarrow\ & (?x, \text{rdf:type}, \text{:Pipeline}) \land \\
& (?x, \text{:fromFacility}, ?y) \land \\
& (?y, \text{rdf:type}, \text{:OilFacility})
\end{aligned}
$$

(:p1, rdf:type, :Pipeline)
(:p1, :fromFacility, :f1)
(:f1, rdf:type, :OilFacility)
(:p2, rdf:type, :OilPipeline)
(:p2, :fromFacility, :f2)
(:f2, rdf:type, :OilFacility)
(:p3, rdf:type, :OilPipeline)

:p1

$$
\begin{aligned}
Q(?x) \leftarrow\ & (?x, \text{rdf:type}, \text{:Pipeline}) \land \\
& (?x, \text{:fromFacility}, ?y) \land \\
& (?y, \text{rdf:type}, \text{:OilFacility})
\end{aligned}
$$

Pipelines from oil facilities?

```
SubClassOf(:OilPipeline
  ObjectIntersectionOf(:Pipeline
    ObjectSomeValuesFrom(:fromFacility :OilFacility)))
```

(:p1, rdf:type, :Pipeline)
(:p1, :fromFacility, :f1)
(:f1, rdf:type, :OilFacility)
(:p2, rdf:type, :OilPipeline)
(:p2, :fromFacility, :f2)
(:f2, rdf:type, :OilFacility)
(:p3, rdf:type, :OilPipeline)

:p1

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

$$Q(?x) \leftarrow (?x, \text{rdf:type}, \text{:Pipeline}) \wedge (?x, \text{:fromFacility}, ?y) \wedge (?y, \text{rdf:type}, \text{:OilFacility})$$

Pipelines from oil facilities?

SubClassOf(:OilPipeline
  ObjectIntersectionOf(:Pipeline
    ObjectSomeValuesFrom(:fromFacility :OilFacility)))

(:p1, rdf:type, :Pipeline)
(:p1, :fromFacility, :f1)
(:f1, rdf:type, :OilFacility)
(:p2, rdf:type, :OilPipeline)
(:p2, :fromFacility, :f2)
(:f2, rdf:type, :OilFacility)
(:p3, rdf:type, :OilPipeline)

:p1, :p2, :p3

$$Q(?x) \leftarrow (?x, \text{rdf:type}, \text{:Pipeline}) \wedge$$
$$(?x, \text{:fromFacility}, ?y) \wedge$$
$$(?y, \text{rdf:type}, \text{:OilFacility})$$

Pipelines from oil facilities?

SubClassOf(:OilPipeline
  ObjectIntersectionOf(:Pipeline
    ObjectSomeValuesFrom(:fromFacility :OilFacility)))

OWL 2 QL ontology

(:p1, rdf:type, :Pipeline)
(:p1, :fromFacility, :f1)
(:f1, rdf:type, :OilFacility)
(:p2, rdf:type, :OilPipeline)
(:p2, :fromFacility, :f2)
(:f2, rdf:type, :OilFacility)
(:p3, rdf:type, :OilPipeline)

:p1, :p2, :p3

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

Pipelines from oil facilities?

$$Q(?x) \leftarrow (?x, \text{rdf:type}, \text{:Pipeline}) \wedge (?x, \text{:fromFacility}, ?y) \wedge (?y, \text{rdf:type}, \text{:OilFacility})$$

rewrite

SubClassOf(:OilPipeline
  ObjectIntersectionOf(:Pipeline
    ObjectSomeValuesFrom(:fromFacility :OilFacility)))

OWL 2 QL ontology

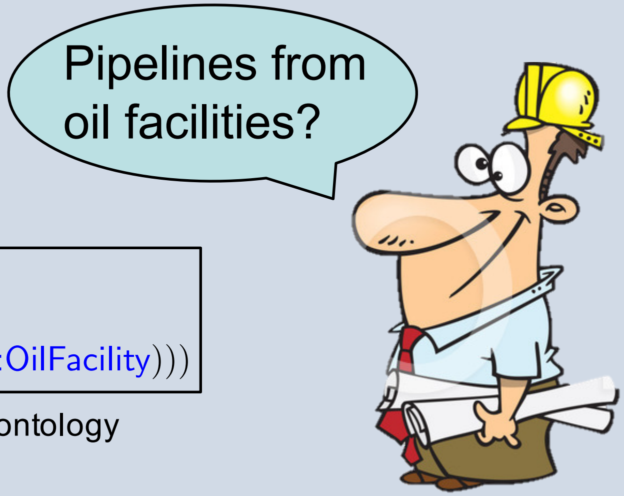
$$Q'(?x) \leftarrow (?x, \text{rdf:type}, \text{:Pipeline}) \wedge (?x, \text{:fromFacility}, ?y) \wedge (?y, \text{rdf:type}, \text{:OilFacility}) \vee (?x, \text{rdf:type}, \text{:OilPipeline})$$

(:p1, rdf:type, :Pipeline)
(:p1, :fromFacility, :f1)
(:f1, rdf:type, :OilFacility)
(:p2, rdf:type, :OilPipeline)
(:p2, :fromFacility, :f2)
(:f2, rdf:type, :OilFacility)
(:p3, rdf:type, :OilPipeline)

:p1, :p2, :p3

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

$$Q(?x) \leftarrow (?x, \text{rdf:type}, \text{:Pipeline}) \wedge (?x, \text{:fromFacility}, ?y) \wedge (?y, \text{rdf:type}, \text{:OilFacility})$$



rewrite

SubClassOf(:OilPipeline
  ObjectIntersectionOf(:Pipeline
    ObjectSomeValuesFrom(:fromFacility :OilFacility)))

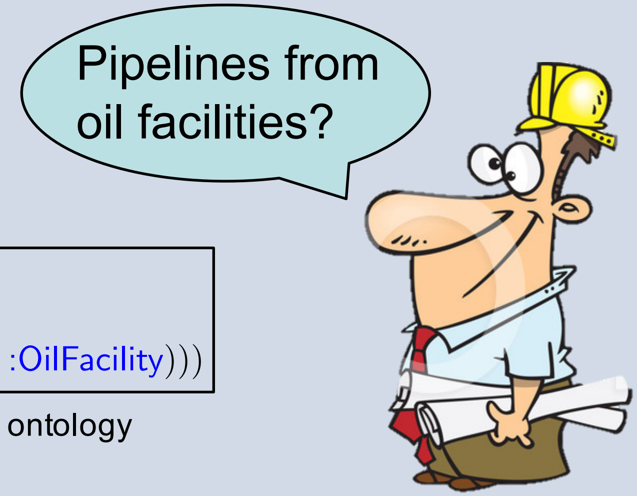
OWL 2 QL ontology

$$Q'(?x) \leftarrow (?x, \text{rdf:type}, \text{:Pipeline}) \wedge (?x, \text{:fromFacility}, ?y) \wedge (?y, \text{rdf:type}, \text{:OilFacility}) \vee (?x, \text{rdf:type}, \text{:OilPipeline})$$

Pipeline

ID	Oil	From
p1	N	f1
p2	Y	f2
p3	Y	Null

$Q(?x) \leftarrow (?x, \text{rdf:type}, \text{:Pipeline}) \wedge$
$(?x, \text{:fromFacility}, ?y) \wedge$
$(?y, \text{rdf:type}, \text{:OilFacility})$



rewrite

SubClassOf(:OilPipeline
 ObjectIntersectionOf(:Pipeline
  ObjectSomeValuesFrom(:fromFacility :OilFacility)))

OWL 2 QL ontology

$Q'(?x) \leftarrow (?x, \text{rdf:type}, \text{:Pipeline}) \wedge$
$(?x, \text{:fromFacility}, ?y) \wedge$
$(?y, \text{rdf:type}, \text{:OilFacility})$
$\vee\ (?x, \text{rdf:type}, \text{:OilPipeline})$

(R2RML) mappings

:OilPipeline = select ID from Pipeline
where Oil = "Y"
⋮

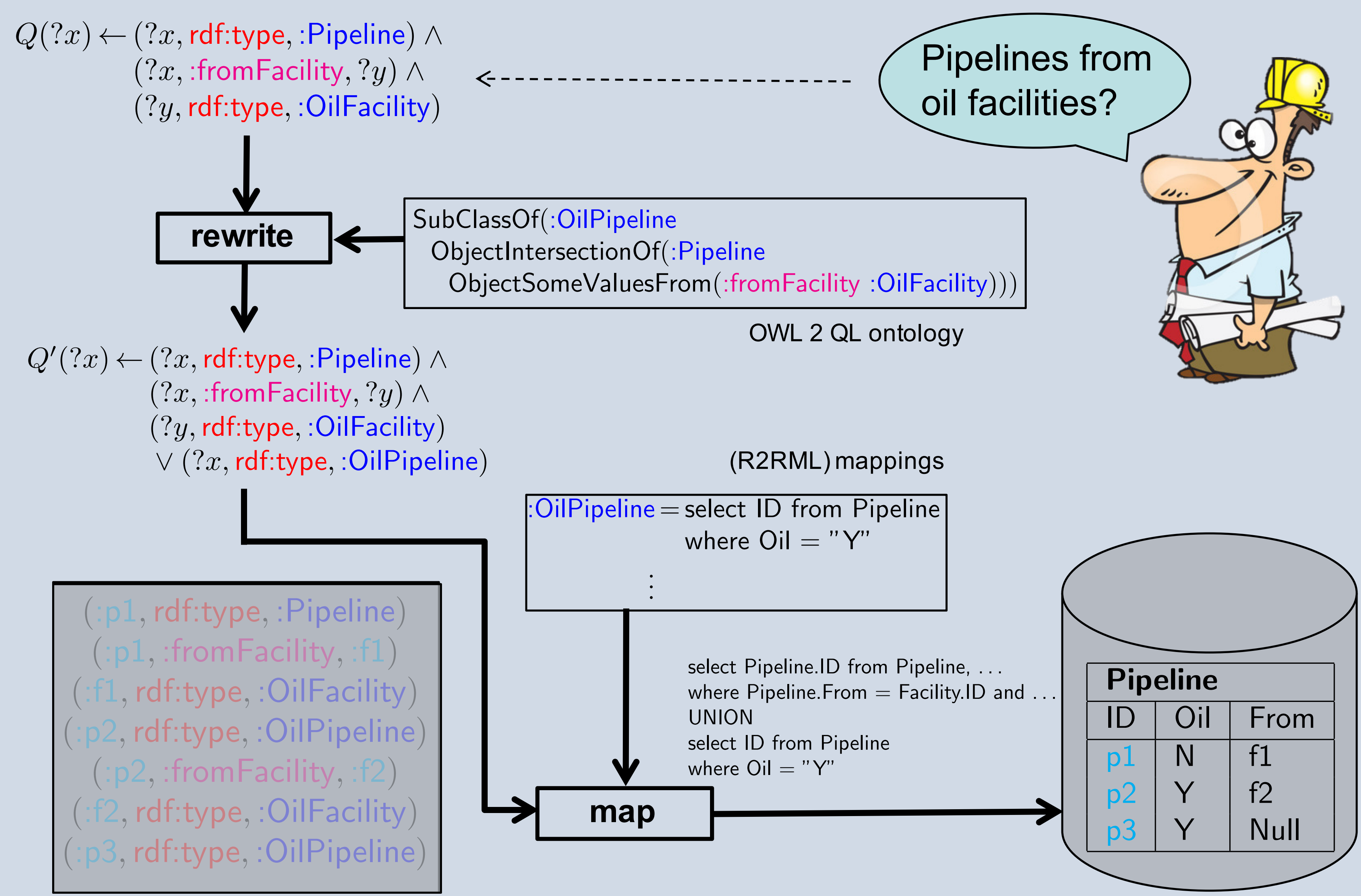
map

(:p1, rdf:type, :Pipeline)
(:p1, :fromFacility, :f1)
(:f1, rdf:type, :OilFacility)
(:p2, rdf:type, :OilPipeline)
(:p2, :fromFacility, :f2)
(:f2, rdf:type, :OilFacility)
(:p3, rdf:type, :OilPipeline)

Pipeline

ID	Oil	From
p1	N	f1
p2	Y	f2
p3	Y	Null

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME



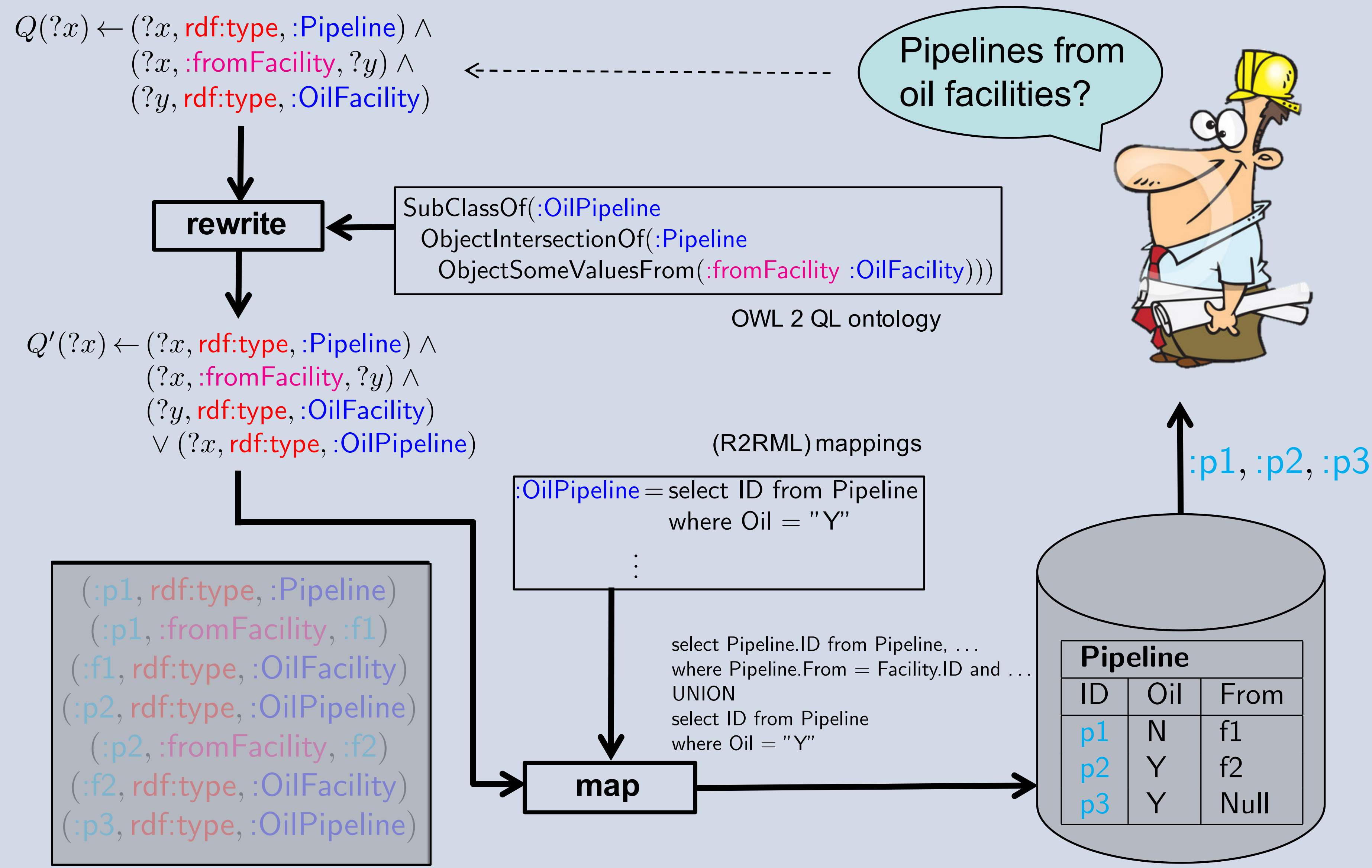
UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE

Information Systems Group

EPSRC Engineering and Physical Sciences Research Council

Optique

SEVENTH FRAMEWORK PROGRAMME



UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE

Information Systems Group

EPSRC Engineering and Physical Sciences Research Council

Optique

SEVENTH FRAMEWORK PROGRAMME

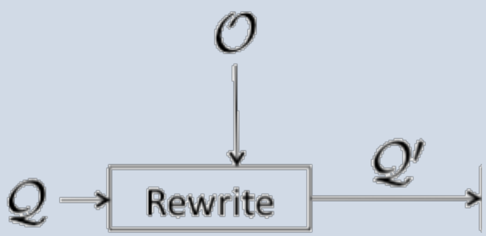
OWL 2 QL and Query Rewriting

Given QL ontology $\mathcal{O}$ query $\mathcal{Q}$ and mappings $\mathcal{M}$:

OWL 2 QL and Query Rewriting

Given QL ontology $\mathcal{O}$ query $\mathcal{Q}$ and mappings $\mathcal{M}$:

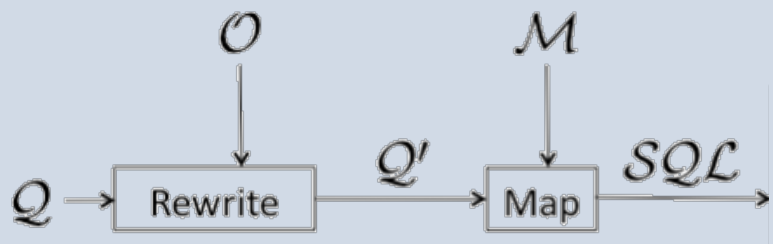
- Use $\mathcal{O}$ to **rewrite** $\mathcal{Q} \rightarrow \mathcal{Q}'$ s.t. answering $\mathcal{Q}'$ without $\mathcal{O}$ is equivalent to answering $\mathcal{Q}$ w.r.t. $\mathcal{O}$ *for any dataset*



OWL 2 QL and Query Rewriting

Given QL ontology $\mathcal{O}$ query $\mathcal{Q}$ and mappings $\mathcal{M}$:

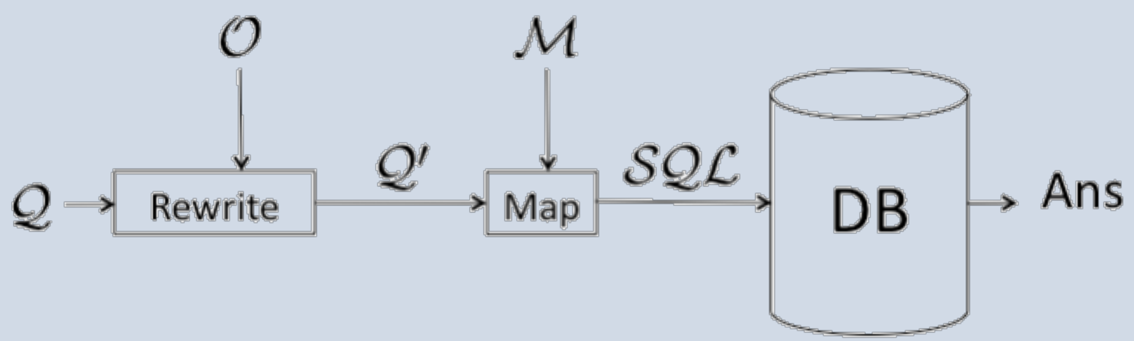
- Use $\mathcal{O}$ to **rewrite** $\mathcal{Q} \rightarrow \mathcal{Q}'$ s.t. answering $\mathcal{Q}'$ without $\mathcal{O}$ is equivalent to answering $\mathcal{Q}$ w.r.t. $\mathcal{O}$ *for any dataset*
- **Map** ontology queries $\rightarrow$ DB queries (typically SQL) using mappings $\mathcal{M}$ to rewrite $\mathcal{Q}'$ into a DB query



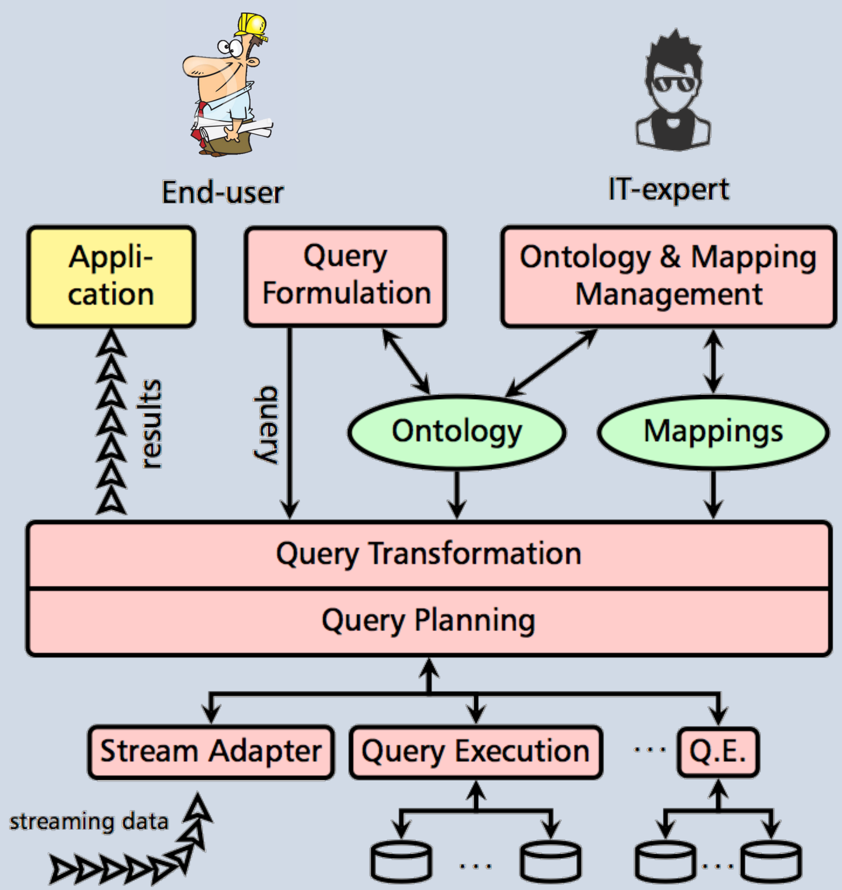
OWL 2 QL and Query Rewriting

Given QL ontology $\mathcal{O}$ query $\mathcal{Q}$ and mappings $\mathcal{M}$:

- Use $\mathcal{O}$ to **rewrite** $\mathcal{Q} \rightarrow \mathcal{Q}'$ s.t. answering $\mathcal{Q}'$ without $\mathcal{O}$ is equivalent to answering $\mathcal{Q}$ w.r.t. $\mathcal{O}$ *for any dataset*
- **Map** ontology queries $\rightarrow$ DB queries (typically SQL) using mappings $\mathcal{M}$ to rewrite $\mathcal{Q}'$ into a DB query
- **Evaluate** (SQL) query against DB



Optique Architecture



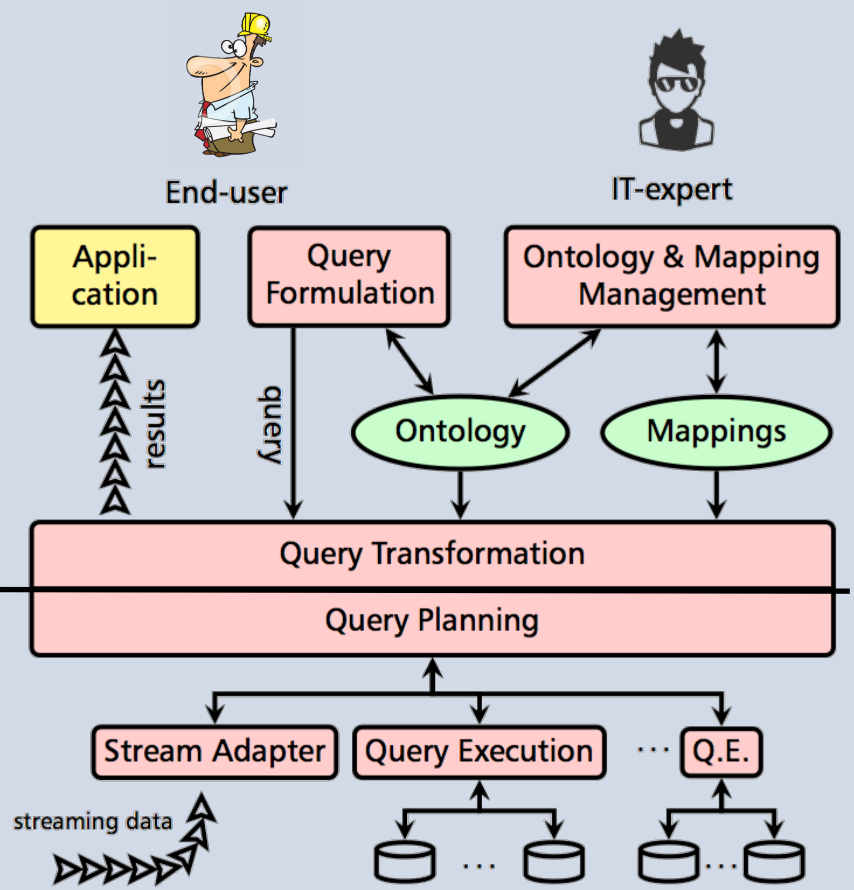
Optique Architecture

Query rewriting:

- uses ontology & mappings
- computationally hard
- ontology & mappings small

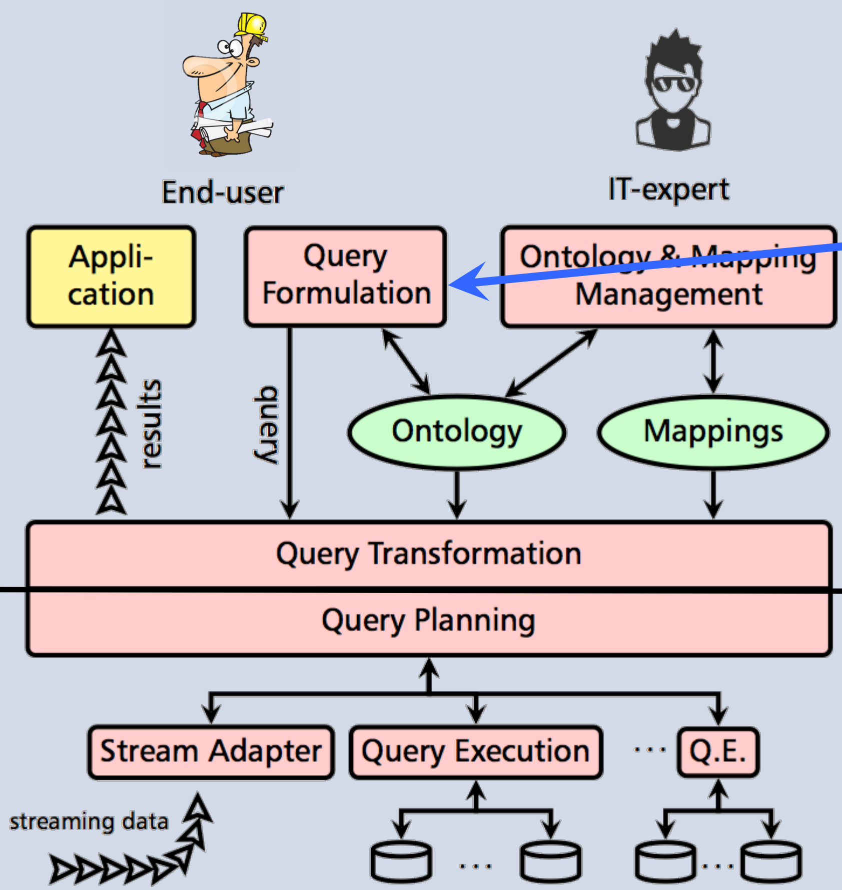
Query evaluation:

- ind. of ontology & mappings
- computationally tractable
- data sets very large



UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

Optique Architecture



Other features:

support for query formulation

Query rewriting:

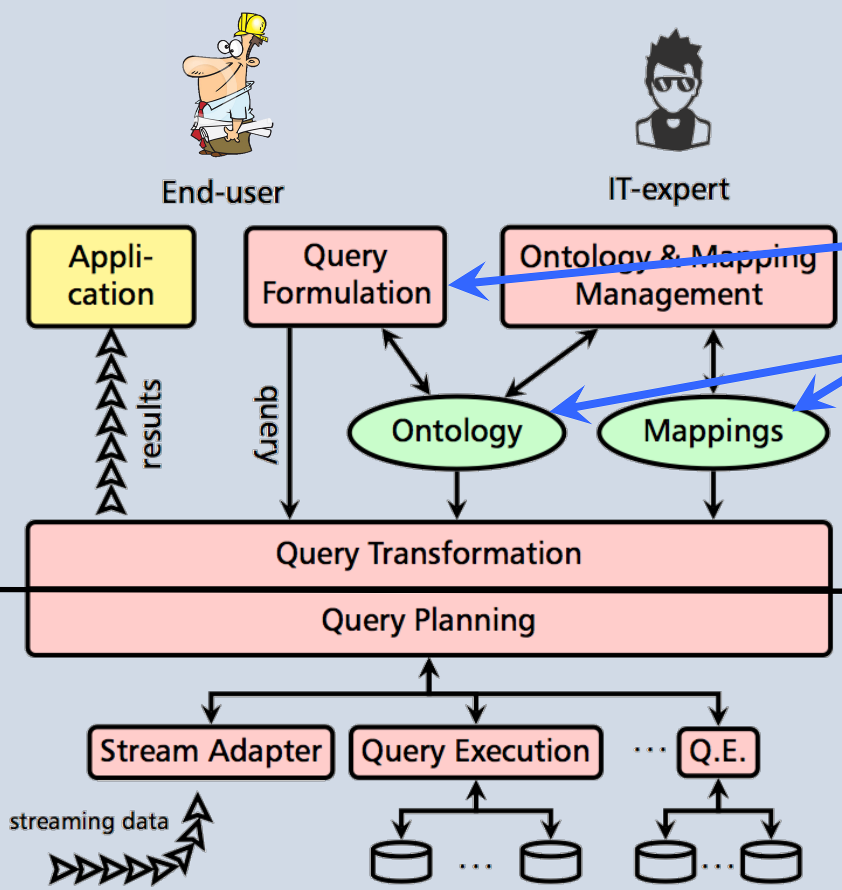
- uses ontology & mappings
- computationally hard
- ontology & mappings small

Query evaluation:

- ind. of ontology & mappings
- computationally tractable
- data sets very large

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF OXFORD · Information Systems Group · EPSRC Engineering and Physical Sciences Research Council · Optique · SEVENTH FRAMEWORK PROGRAMME

Optique Architecture



Other features:

support for query formulation

"Bootstrapping" Ontology & mappings

Query rewriting:

- uses ontology & mappings
- computationally hard
- ontology & mappings small

Query evaluation:

- ind. of ontology & mappings
- computationally tractable
- data sets very large

Query Rewriting — Issues

1 Rewriting

- May be large (worst case exponential in size of ontology)
- Queries may be hard for existing DBMSs

2 Mappings

- May be difficult to develop and maintain

3 Expressivity

- OWL 2 QL (necessarily) has (very) restricted expressive power, e.g.:
  - No functional or transitive properties
  - No universal (for-all) restrictions
  - …

OWL Profiles – Beyond QL?

OWL 2 defines language subsets, aka **profiles** that can be "more simply and/or efficiently implemented"

- **OWL 2 QL**
  - Based on DL-Lite
  - Efficiently implementable via rewriting into relational queries (OBDA)
- **OWL 2 RL**
  - Based on "**Description Logic Programs**" ($\approx$ DL $\cap$ Datalog)
  - Implementable via Datalog query answering
- **OWL 2 EL**
  - Based on $\mathcal{EL}^{++}$
  - Implementable via Datalog query answering plus "filtration"

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

RL/Datalog Query Ans. via Materialisation

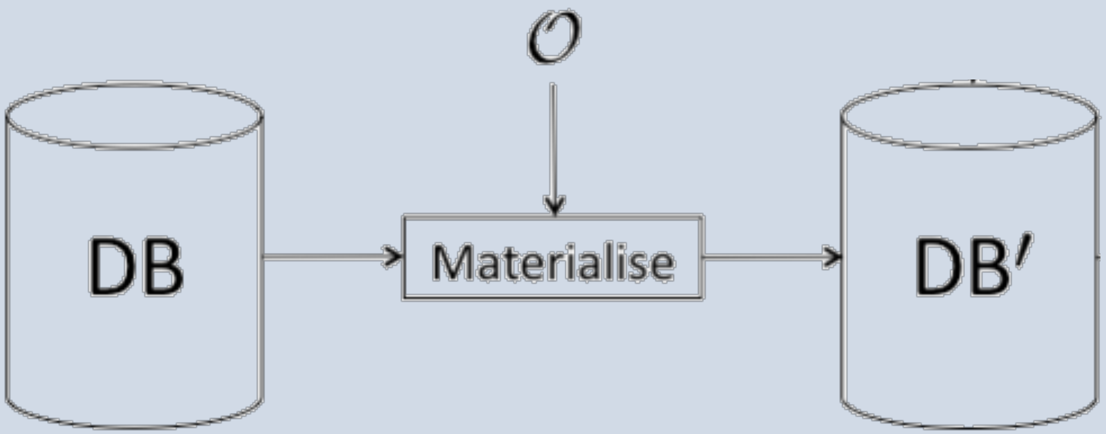
Given (RDF) data DB, RL/Datalog ontology $\mathcal{O}$ and query $\mathcal{Q}$:

RL/Datalog Query Ans. via Materialisation

Given (RDF) data DB, RL/Datalog ontology $\mathcal{O}$ and query $\mathcal{Q}$:

- **Materialise** (RDF) data DB $\rightarrow$ DB$'$ s.t. evaluating $\mathcal{Q}$ w.r.t. DB$'$ equivalent to answering $\mathcal{Q}$ w.r.t. DB and $\mathcal{O}$

  nb: Closely related to **chase** procedure used with DB dependencies



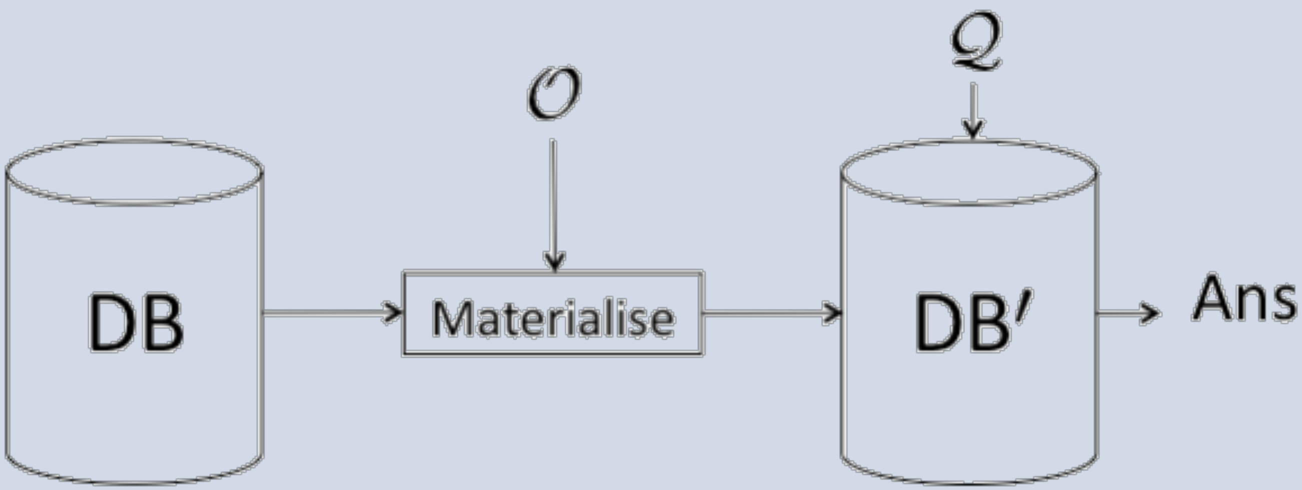
RL/Datalog Query Ans. via Materialisation

Given (RDF) data DB, RL/Datalog ontology $\mathcal{O}$ and query $\mathcal{Q}$:

- **Materialise** (RDF) data DB $\longrightarrow$ DB$'$ s.t. evaluating $\mathcal{Q}$ w.r.t. DB$'$ equivalent to answering $\mathcal{Q}$ w.r.t. DB and $\mathcal{O}$

  nb: Closely related to **chase** procedure used with DB dependencies

- **Evaluate** $\mathcal{Q}$ against DB$'$



Materialisation — Example

$$\mathcal{O}\left\{\ \mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip}\right.$$

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD
Information Systems Group
EPSRC
Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

Materialisation — Example

$$\mathcal{O}\left\{\ \mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip}\right.$$

$$\mathrm{DB}\left\{\begin{array}{l}\mathsf{OilEquip}(\mathsf{a})\\ \mathsf{hasPart}(\mathsf{a},\mathsf{b})\\ \mathsf{hasPart}(\mathsf{b},\mathsf{c})\end{array}\right.$$

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

Materialisation — Example

$\mathcal{O}$ $\left\{ \mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip} \right.$

$$\mathrm{DB} \begin{cases} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \end{cases}$$

$\mathcal{Q}_1 \quad Q(x) \leftarrow \mathsf{OilEquip}(x)$

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

Materialisation — Example

$$\mathcal{O}\ \left\{\ \mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip}\right.$$

$$\mathsf{DB} \left\{ \begin{array}{l} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \end{array} \right.$$

$$\mathcal{Q}_1 \quad Q(x) \leftarrow \mathsf{OilEquip}(x) \qquad \rightsquigarrow \qquad \{a\}$$

Materialisation — Example

$\mathcal{O}$ $\{$ $\mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip}$ $\quad$ $\mathsf{OilEquip}(x) \wedge \mathsf{hasPart}(x, y) \rightarrow \mathsf{OilEquip}(y)$

$$\mathsf{DB} \begin{cases} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \end{cases}$$

$\mathcal{Q}_1$ $\quad$ $Q(x) \leftarrow \mathsf{OilEquip}(x)$

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

Materialisation — Example

$\mathcal{O}$ $\Big\{$ $\mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip}$ $\quad$ $\mathsf{OilEquip}(x) \land \mathsf{hasPart}(x, y) \rightarrow \mathsf{OilEquip}(y)$

$$\mathsf{DB} \begin{cases} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \end{cases} \qquad \mathsf{DB}' \begin{cases} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \end{cases}$$

$\mathcal{Q}_1$ $\quad Q(x) \leftarrow \mathsf{OilEquip}(x)$

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

Materialisation — Example

$\mathcal{O}$ $\left\{ \mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip} \quad \mathsf{OilEquip}(x) \wedge \mathsf{hasPart}(x,y) \rightarrow \mathsf{OilEquip}(y) \right.$

$$\mathsf{DB} \left\{ \begin{array}{l} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \end{array} \right. \qquad \mathsf{DB}' \left\{ \begin{array}{l} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \\ \mathsf{OilEquip}(\mathsf{b}) \end{array} \right.$$

$\mathcal{Q}_1 \quad Q(x) \leftarrow \mathsf{OilEquip}(x)$

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME

Materialisation — Example

$$\mathcal{O}\left\{\ \mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip} \qquad \mathsf{OilEquip}(x) \wedge \mathsf{hasPart}(x,y) \rightarrow \mathsf{OilEquip}(y)\right.$$

$$\mathsf{DB}\left\{\begin{array}{l}\mathsf{OilEquip}(\mathsf{a})\\ \mathsf{hasPart}(\mathsf{a},\mathsf{b})\\ \mathsf{hasPart}(\mathsf{b},\mathsf{c})\end{array}\right. \qquad \mathsf{DB}'\left\{\begin{array}{l}\mathsf{OilEquip}(\mathsf{a})\\ \mathsf{hasPart}(\mathsf{a},\mathsf{b})\\ \mathsf{hasPart}(\mathsf{b},\mathsf{c})\\ \mathsf{OilEquip}(\mathsf{b})\\ \mathsf{OilEquip}(\mathsf{c})\end{array}\right.$$

$$\mathcal{Q}_1 \quad Q(x) \leftarrow \mathsf{OilEquip}(x)$$

Materialisation — Example

$\mathcal{O} \left\{ \mathsf{OilEquip} \sqsubseteq \forall \mathsf{hasPart}.\mathsf{OilEquip} \qquad \mathsf{OilEquip}(x) \wedge \mathsf{hasPart}(x, y) \rightarrow \mathsf{OilEquip}(y) \right.$

$$\mathsf{DB} \left\{ \begin{array}{l} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \end{array} \right. \qquad \mathsf{DB}' \left\{ \begin{array}{l} \mathsf{OilEquip}(\mathsf{a}) \\ \mathsf{hasPart}(\mathsf{a}, \mathsf{b}) \\ \mathsf{hasPart}(\mathsf{b}, \mathsf{c}) \\ \mathsf{OilEquip}(\mathsf{b}) \\ \mathsf{OilEquip}(\mathsf{c}) \end{array} \right.$$

$\mathcal{Q}_1 \quad Q(x) \leftarrow \mathsf{OilEquip}(x) \qquad \rightsquigarrow \qquad \{a, b, c\}$

Materialisation — Issues

1 Scalability

- Ptime complete
- Efficiently implementable in practice?

2 Updates

- Additions relatively easy (continue materialisation)
- But what about retraction?

3 Migrating data to RDF

- Materialisation assumes data in “special” (RDF triple) store
- How can legacy data be migrated?

4 Expressivity

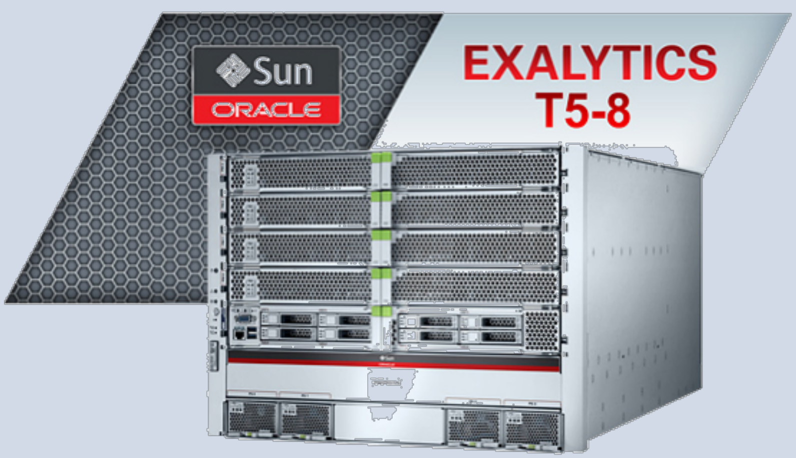
- $\mathcal{QL} \not\subseteq \mathcal{RL}$; in particular, no invention of new individuals

Materialisation: Scalability

- Efficient **Datalog/RL** engine is critical
- Existing approaches mainly target distributed “shared-nothing” architectures, often via **map reduce**
  - High communication overhead
  - Typically focus on small fragments (e.g., RDFS), so don’t really address expressivity issue
  - Even then, query answering over (distributed) materialized data is non-trivial and may require considerable communication

RDFox Datalog Engine

- Targets SOTA **main-memory, mulit-core** architecture
  - Optimized in-memory storage with ‘mostly’ lock-free parallel inserts
  - Memory efficient: commodity server with 128 GB can store $>10^9$ triples
  - Exploits multi-core architecture: 10-20 x speedup with 32/16 threads/cores
  - LUBM 120K ($>10^{10}$ triples) in 251s (20M t/s) on T5-8 (4TB/1024 threads)

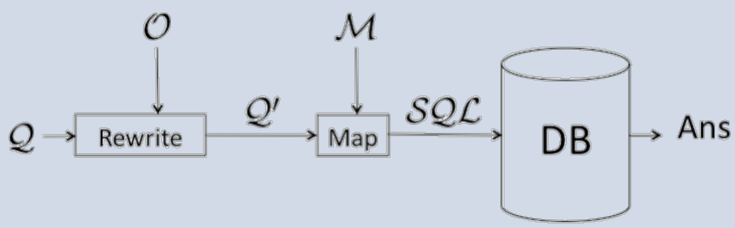


RDFox Datalog Engine

- **Incremental addition and retraction** of triples
  - Retraction via novel FBF “view maintenance” algorithm
  - Retraction of 5,000 triples from materialised LUBM 50k in less than 1s
- Many other **novel features**
  - Handles more general (than RL) Dalalog and SWRL rules
  - SPARQL features such as BIND and FILTER in rule bodies
  - Native equality handling (owl:sameAs) via rewriting
  - Stratified negation as failure (NAF)

Materialisation: Data Migration

- Need to specify a suitable **migration** process
  - Use **R2RML** mappings to extract data and transform into RDF
  - But where do these mappings come from?
- Recall query rewriting:
  - **Mappings** $\mathcal{M}$ *are* R2RML mappings
  - Run mappings **in reverse** to extract and transform data



- "**Lazy ETL**"
  - Deploy query rewriting (OBDA) system
  - Extend $\mathcal{O}$ and $\mathcal{M}$ as needed
  - Use $\mathcal{M}$ to ETL data into RDF store

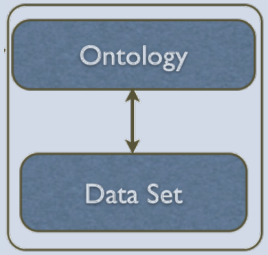
Materialisation: Expressivity

- RL is more powerful than QL, but QL $\not\subseteq$ RL
  - In particular, no "individual creation" (RHS existentials)
  - Can't express, e.g., $\mathsf{OilPipeline} \sqsubseteq \mathsf{Pipeline} \sqcap \exists \mathsf{fromFacility}.\mathsf{OilFacility}$

- Recall **OWL 2 EL**
  - Based on $\mathcal{EL}^{++}$
  - Implementable via Datalog query answering plus "filtration"

OWL 2 EL via Datalog + Filtration

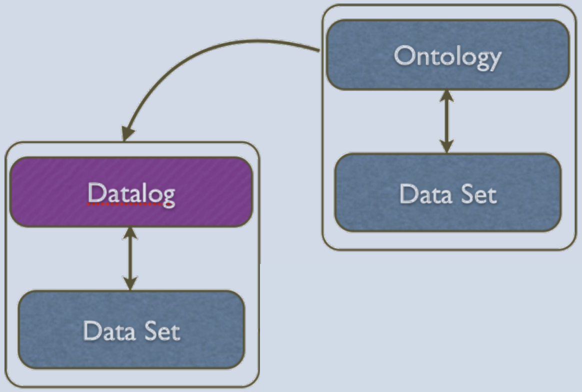
Given (RDF) Data Set, EL ontology $\mathcal{O}$ and query $\mathcal{Q}$:



OWL 2 EL via Datalog + Filtration

Given (RDF) Data Set, EL ontology $\mathcal{O}$ and query $\mathcal{Q}$:

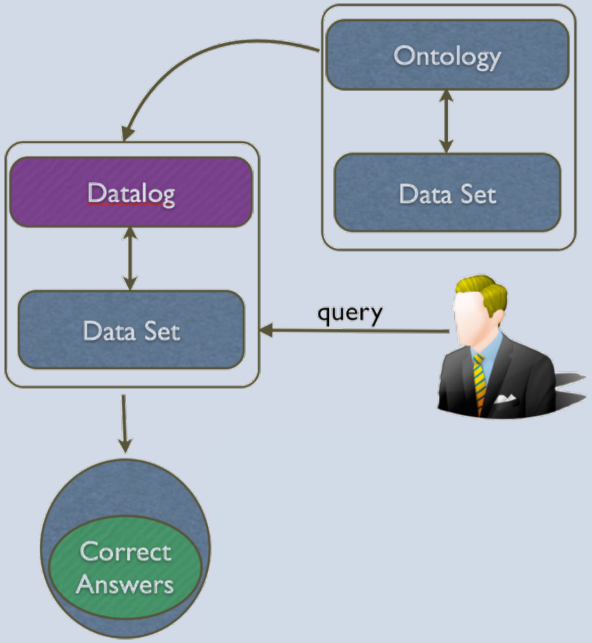
- **Over-approximate** $\mathcal{O}$ into Datalog program D



OWL 2 EL via Datalog + Filtration

Given (RDF) Data Set, EL ontology $\mathcal{O}$ and query $\mathcal{Q}$:

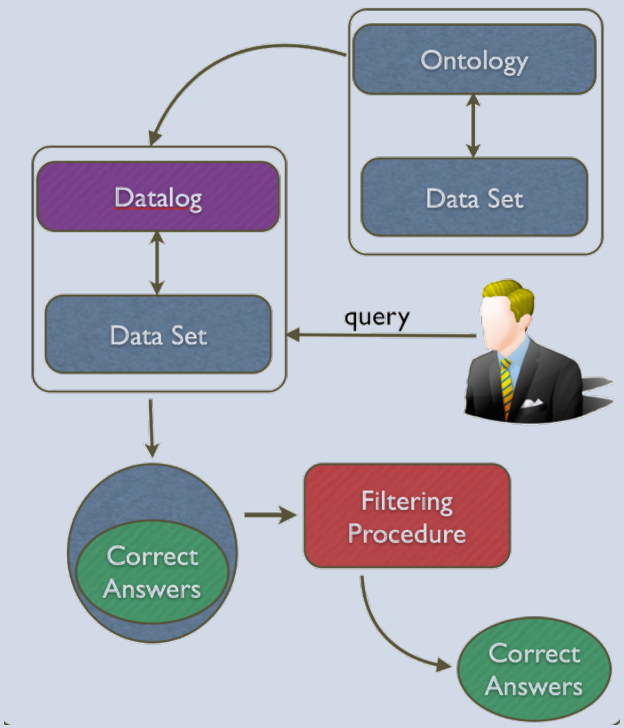
- **Over-approximate** $\mathcal{O}$ into Datalog program D
- **Evaluate** $\mathcal{Q}$ over D + Data Set (via materialisation)



OWL 2 EL via Datalog + Filtration

Given (RDF) Data Set, EL ontology $\mathcal{O}$ and query $\mathcal{Q}$:

- **Over-approximate** $\mathcal{O}$ into Datalog program D
- **Evaluate** $\mathcal{Q}$ over D + Data Set (via materialisation)
- Use (polynomial) **Filtering Procedure** to eliminate spurious answers



Discussion

- **QL-Rewriting** has many advantages
  - Data can be left untouched and in legacy storage
  - Exploits existing DB infrastructure and scalability
  - …
- But what if **more expressiveness/flexibility** is needed?
  - Query answering for EL and RL still tractable (polynomial)
  - Critically depend on Datalog scalability – RDFox to the rescue!
  - Easy migration path from QL-rewriting via “lazy ETL”

Future Work

- Piloting, evaluation and tuning
- Porting to other large-scale architectures
- Semantic (data) partitioning for distributed architectures
- (Incremental maintenance of) aggregations
- Improved query planning
- Stream reasoning
- Hybrid rewriting/materialisation (on demand) approach
- Expressiveness beyond RL/EL via PAGOdA techniques
- …

Acknowledgements

EPSRC Engineering and Physical Sciences Research Council

SIXTH FRAMEWORK PROGRAMME

SEVENTH FRAMEWORK PROGRAMME

DARPA

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE

Information Systems Group

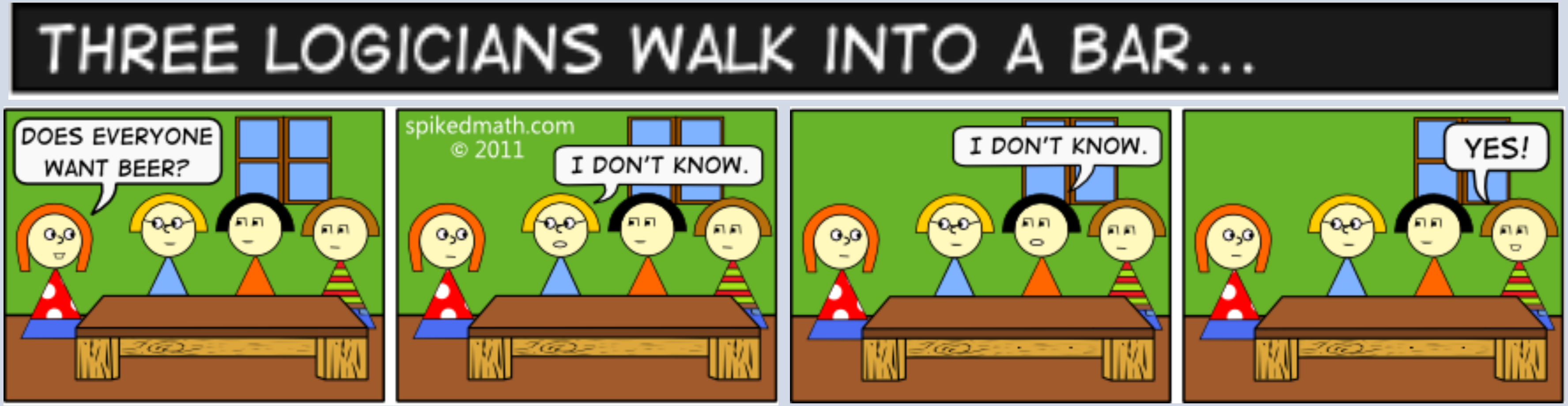
EPSRC Engineering and Physical Sciences Research Council

Optique

SEVENTH FRAMEWORK PROGRAMME

Thank you for listening

Thank you for listening



Any questions?

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD
Information Systems Group
EPSRC
Engineering and Physical Sciences Research Council
Optique
SEVENTH FRAMEWORK PROGRAMME