Linked Data and Efficient OWL 2 Reasoning

Tutorial at CSWS2010

Jeff Z. Pan

Department of Computing Science
University of Aberdeen, UK

Tutorial web site: http://www.abdn.ac.uk/~csc280/tutorial/csws2010/

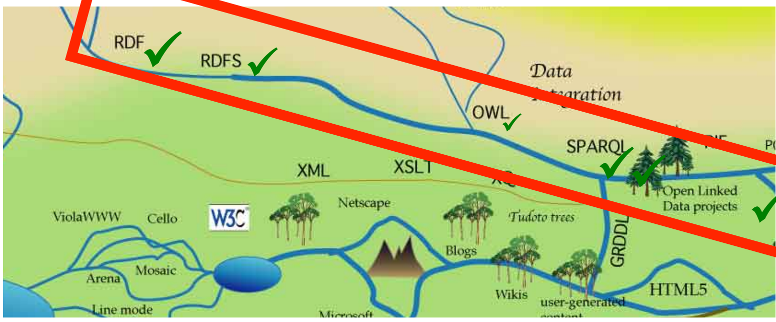
UNIVERSITY OF ABERDEEN

The Web map 2008 © Tim Berners-Lee

http://www.w3.org/2007/09/map/main.jpg

The Web map 2008 © Tim Berners-Lee

CSWS2010



- more and more ontologies and ontological data available thanks to …
- … related standards (RDF, OWL) becoming established
- … the community (across many application domains) is growing fast
- … In this tutorial: How to use OWL 2 for linked dada, as well as efficient and scalable ontology reasoning for OWL 2

3 UNIVERSITY OF ABERDEEN

Outline

CSWS2010

- **Session 1: Introduction to Linked Data and OWL 2**
  - Now: Lined Data and Ontology
  - Overview: from OWL 1 to OWL 2
  - Reasoning services in OWL 2
- Session 2: Quality Guaranteed Approximations
  - Challenges
  - Soundness preserving and Completeness preserving approximations
  - A hands-on session

4 UNIVERSITY OF ABERDEEN

What is RDF

CSWS2010

- Resource Description Framework
  - Makes content machine readable by introducing descriptions/annotations
- RDF statements
  - Subject-property-value triple
- Examples (in N3 and XML syntax)

```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows
        <http://dblp.l3s.de/d2r/…/Dan_Brickley> .
```

```
<rdf:Description rdf:about="http://www.w3.org/People/Berners-Lee/card#i">
    <foaf:knows rdf:resource="http://dblp.l3s.de/d2r/…/Dan_Brickley ">
</rdf:Description>
```

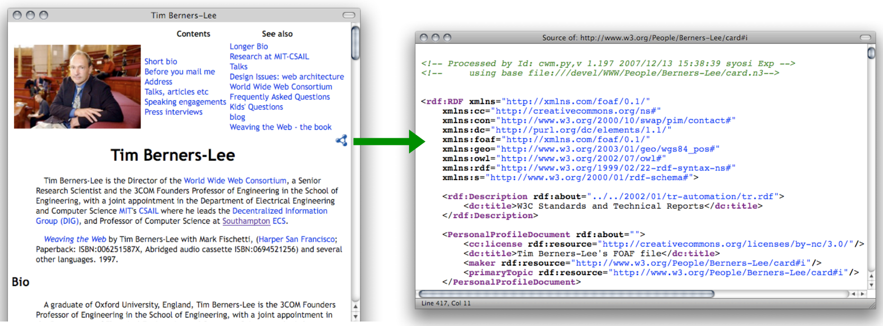
UNIVERSITY OF ABERDEEN

RDF on the Web

CSWS2010

- **(i) directly by the publishers**
- (ii) by e.g. GRDDL transformations, D2R, RDFa exporters, etc.

FOAF/RDF linked from a home page: personal data (foaf:name, foaf:phone, etc.), relationships (foaf:knows, rdfs:seeAlso)



6

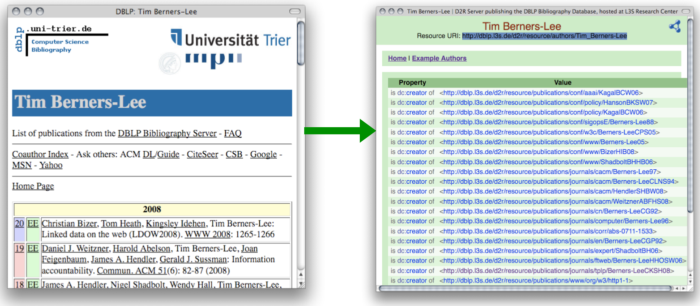
UNIVERSITY OF ABERDEEN

RDF on the Web

CSWS2010

- (i) directly by the publishers
- **(ii) by e.g. GRDDL transformations, D2R, RDFa exporters, etc.**

e.g. L3S' RDF export of the DBLP citation index, using FUB's D2R (http://dblp.l3s.de/d2r/)



Gives unique URIs to authors, documents, etc. on DBLP! E.g.,

http://dblp.l3s.de/d2r/resource/authors/Tim_Berners-Lee,
http://dblp.l3s.de/d2r/resource/publications/journals/tplp/Berners-LeeCKSH08

Provides RDF version of all DBLP data + query interface!

7

UNIVERSITY of ABERDEEN

RDF Data online: Example

CSWS2010

- Data in RDF: **Triples**
  - DBLP:

```
<http://dblp.l3s.de/…/journals/tplp/Berners-LeeCKSH08> rdf:type swrc:Article.
<http://dblp.l3s.de/…/journals/tplp/Berners-LeeCKSH08> dc:creator
     <http://dblp.l3s.de/d2r/…/Tim_Berners-Lee> .
  …
<http://dblp.l3s.de/d2r/…/Tim_Berners-Lee> foaf:homepage
       <http://www.w3.org/People/Berners-Lee/> .
…
<http://dblp.l3s.de/d2r/…/Dan_Brickley> foaf:name "Dan Brickley"^^xsd:string.
```

  - Tim Berners-Lee's FOAF file:

```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows
     <http://dblp.l3s.de/d2r/…/Dan_Brickley> .
<http://www.w3.org/People/Berners-Lee/card#i> rdf:type foaf:Person .
<http://www.w3.org/People/Berners-Lee/card#i> foaf:homepage
     <http://www.w3.org/People/Berners-Lee/> .
```

8

UNIVERSITY of ABERDEEN

How can I query that data? SPARQL

- SPARQL – W3C approved standardized query language for RDF:
  - look-and-feel of "SQL for the Web"
  - allows to ask queries like
    - "All documents by Tim Berners-Lee"
    - "Names of all persons who co-authored with authors of http://dblp.l3s.de/d2r/…/Berners-LeeCKSH08 or known by co-authors"

    …

Example:

```
SELECT ?D
FROM <http://dblp.l3s.de/…/authors/Tim_Berners-Lee>
WHERE {?D dc:creator <http://dblp.l3s.de/…/authors/Tim_Berners-Lee>}
```

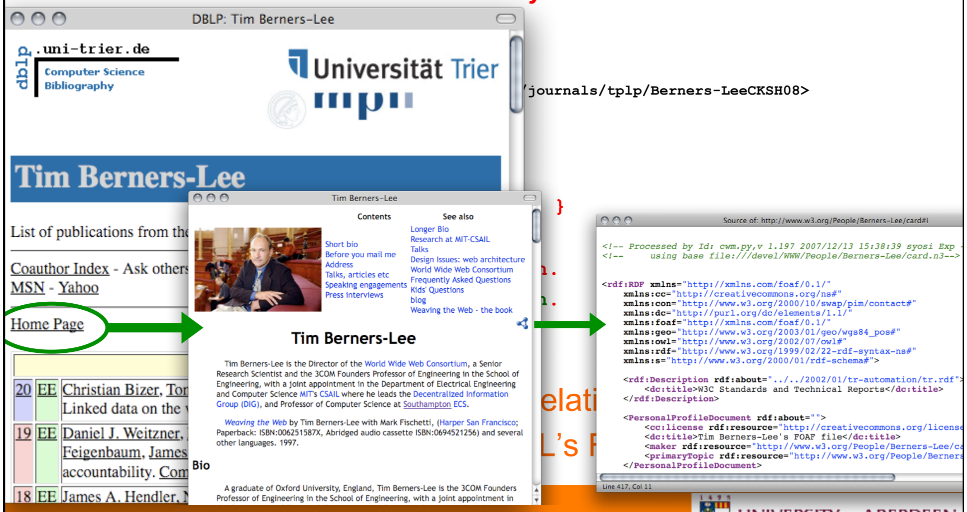

SPARQL more complex patters: e.g. UNIONs

- "**Names** ***of all persons who co-authored with authors of http://dblp.l3s.de/d2r/…/Berners-LeeCKSH08***"

```
SELECT ?Name WHERE
   { <http://dblp.l3s.de/d2r/resource/publication/journals/tplp/Berners-LeeCKSH08>
        dc:creator ?Author.
        ?D dc:creator ?Author.
        ?D dc:creator ?CoAuthor.
        ?CoAuthor foaf:name ?Name
   }
```

SPARQL more complex patters: e.g. UNIONs

- "Names *of all persons who* ***co-authored*** *with authors of* *http://dblp.l3s.de/d2r/…/Berners-LeeCKSH08* ***or known by co-authors***"



UNIVERSITY OF ABERDEEN

Back to the Data:

- DBLP:

```
<http://dblp.l3s.de/…/journals/tplp/Berners-LeeCKSH08> rdf:type swrc:Article.
<http://dblp.l3s.de/…/journals/tplp/Berners-LeeCKSH08> dc:creator
    <http://dblp.l3s.de/d2r/…/Tim_Berners-Lee> .
 …
<http://dblp.l3s.de/d2r/…/Tim_Berners-Lee> foaf:homepage
      <http://www.w3.org/People/Berners-Lee/> .
```

- Tim Berners-Lee's FOAF file:

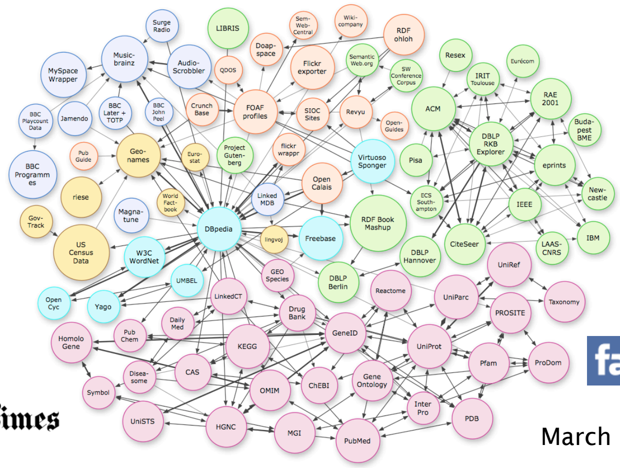
```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows
    <http://dblp.l3s.de/d2r/…/Dan_Brickley> .
<http://www.w3.org/People/Berners-Lee/card#i> foaf:homepage
    <http://www.w3.org/People/Berners-Lee/> .
```

- Even if I have the FOAF data, I cannot answer the query:
  - Different identifiers used for Tim Berners-Lee
  - Who tells me that Dan Brickley is a foaf:Person?
- Need OWL and **Reasoning**!

UNIVERSITY OF ABERDEEN

Linked Open Data

CSWS2010



...

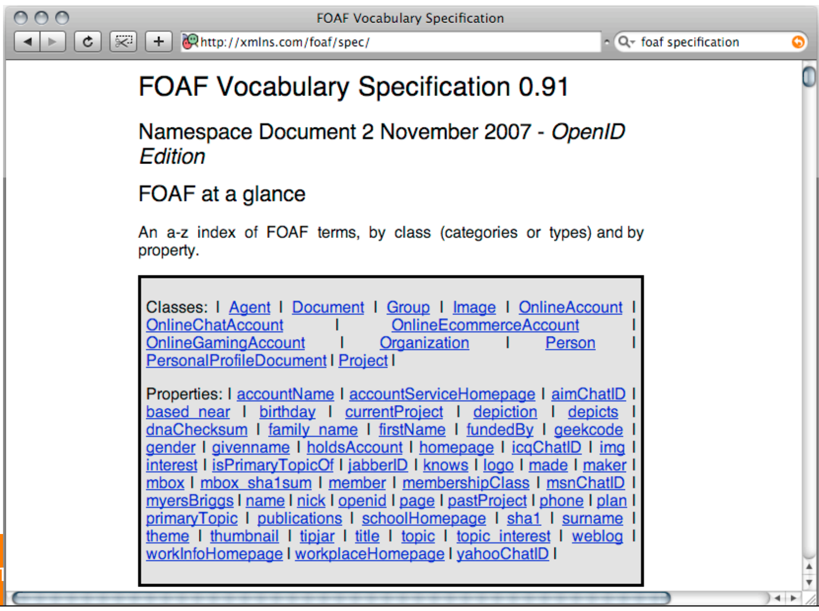
March 2009

- Excellent tutorial here: http://www4.wiwiss.fu- berlin.de/bizer/pub/LinkedDataTutorial/

13

Reasoning on Semantic Web Data

CSWS2010



FOAF Vocabulary Specification

http://xmlns.com/foaf/spec/

foaf specification

FOAF Vocabulary Specification 0.91

Namespace Document 2 November 2007 - *OpenID Edition*

FOAF at a glance

An a-z index of FOAF terms, by class (categories or types) and by property.

Classes: | Agent | Document | Group | Image | OnlineAccount | OnlineChatAccount | OnlineEcommerceAccount | OnlineGamingAccount | Organization | Person | PersonalProfileDocument | Project |

Properties: | accountName | accountServiceHomepage | aimChatID | based_near | birthday | currentProject | depiction | depicts | dnaChecksum | family_name | firstName | fundedBy | geekcode | gender | givenname | holdsAccount | homepage | icqChatID | img | interest | isPrimaryTopicOf | jabberID | knows | logo | made | maker | mbox | mbox_sha1sum | member | membershipClass | msnChatID | myersBriggs | name | nick | openid | page | pastProject | phone | plan | primaryTopic | publications | schoolHomepage | sha1 | surname | theme | thumbnail | tipjar | title | topic | topic_interest | weblog | workInfoHomepage | workplaceHomepage | yahooChatID |

Ontologies: Example FOAF

CSWS2010

`foaf:knows rdfs:domain foaf:Person`

$$\exists knows.\top \sqsubseteq Person$$

`foaf:knows rdfs:range foaf:Person`

$$\exists knows^{-}.\top \sqsubseteq Person$$

`foaf:Person rdfs:subclassOf foaf:Agent`

$$Person \sqsubseteq Agent$$

`foaf:homepage rdf:type owl:inverseFunctionalProperty .`

$$\top \sqsubseteq \leq 1 homepage^{-}$$

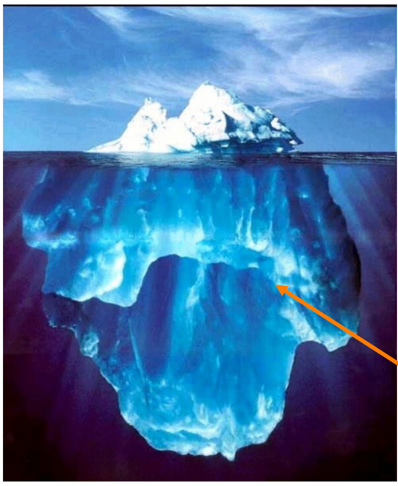
…

15

UNIVERSITY OF ABERDEEN

Why Reasoning

CSWS2010



Linked Data

explicit data

implicit data

Because, e.g., *we want to exploit implicit data for "query answering"*

UNIVERSITY OF ABERDEEN

Example: Querying Jeff’s Talks

CSWS2010

Query:
Give me all talks presented by Jeff
`eswc:Jeff ex:presented ?talk .`

EXPLICIT

IMPLICIT

`eswc:Jeff ex:presented eswc:LDReasoningTut .`

UNIVERSITY OF ABERDEEN

Problem: Incomplete Answers

CSWS2010

Query:
Give me all talks presented by Jeff
eswc:Jeff ex:presented *?talk* .

EXPLICIT

IMPLICIT

```
eswc:Jeff ex:presented eswc:LDReasoningTut .
aaai:OntoReasoningTut ex:presentedBy aaai:Jeff .
```

UNIVERSITY OF ABERDEEN

Problem: Incomplete Answers

CSWS2010

Query:
Give me all talks presented by Jeff
eswc:Jeff ex:presented *?talk* .

EXPLICIT

```
eswc:Jeff ex:presented eswc:LDReasoningTut .
aaai:OntoReasoningTut ex:presentedBy aaai:Jeff .
eswc:Jeff owl:sameAs aaai:Jeff .
```

IMPLICIT

UNIVERSITY OF ABERDEEN

Problem: Incomplete Answers

CSWS2010

Query:
Give me all talks presented by Jeff
eswc:Jeff ex:presented *?talk* .

EXPLICIT

```
eswc:Jeff ex:presented eswc:LDReasoningTut .
aaai:OntoReasoningTut ex:presentedBy aaai:Jeff .
eswc:Jeff owl:sameAs aaai:Jeff .
ex:presentedBy owl:inverseOf ex:presented .
```

IMPLICIT

```
eswc:Jeff ex:presented aaai:OntoReasoningTut .
```

UNIVERSITY OF ABERDEEN

OWL2

HOT! NEW!

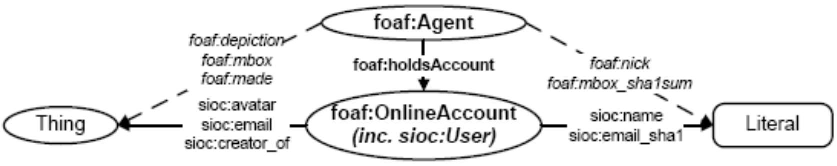
CSWS2010

Common ontologies on the Web don't use it a lot as of yet…

… but adds interesting functionality, potentially useful for Web ontologies, e.g.

- PropertyChains
  - E.g. could be useful to tie sioc:name and foaf:nick via foaf:holdsAccount:

```
foaf:nick owl:propertyChainAxiom (foaf:holdsAccount sioc:name)
```



OWL2

HOT! NEW!

CSWS2010

Common ontologies on the Web don't use it a lot as of yet…

… but adds interesting functionality, potentially useful for Web ontologies, e.g.

- hasKey:
  - *Multi-attribute Keys now possible in OWL, e.g.*
    - `foaf:OnlineAccount/sioc:User` members are uniquely identified by a combination of `foaf:accountName` and `foaf:accountServiceHomepage`:

```
foaf:OnlineAccount owl:hasKey
    (foaf:accountName foaf:accountServiceHomepage) .
```

  - *OWL DL does not allow Datatype Properties to be IFPs*
    - *E.g.* foaf:mbox_sha1sum
    - IFPs could partly be simulated using OWL2's owl:hasKey, as follows:
      `owl:Thing owl:hasKey ( foaf:mbox_sha1sum ) .`
    - means that if *two named individuals* have the same mbox_sha1sum, then they are the same.

Why OWL1 is Not Enough

CSWS2010

- Too expensive to reason with
  - High complexity: NEXPTIME-complete
  - The most lightweight sublanguage OWL-Lite is **NOT** lightweight
  - Some ontologies only use some limited expressive power; e.g. The SNOMED (Systematised Nomenclature of Medicine) ontology
- Not expressive enough; e.g.
  - No user defined datatypes [Pan 2004; Pan and Horrocks 2006; Motik and Horrocks 2008]
  - No metamodeling support [Pan 2004; Pan, Horrocks, Schreiber, 2005; Motik 2007]
  - Limited property support [Horrocks et al., 2006]

23 UNIVERSITY OF ABERDEEN

From OWL1 to OWL2

CSWS2010

- OWL2: A new version of OWL
- Main goals:

  1. To define "profiles" of OWL that are:
     - smaller, easier to implement and deploy
     - cover important application areas and are easily understandable to non-expert users

  2. To add a few extensions to current OWL that are useful, and is known to be implementable
     - many things happened in research since 2004

24 UNIVERSITY OF ABERDEEN

New Expressiveness in OWL 2

CSWS2010

- New expressive power
  - user defined datatypes, e.g.:

```
ex:personAge owl:equivalentClass _:x
_:x rdf:type rdfs:Datatype
_:x owl:onDatatype xsd:integer
_:x owl:withRestrictions (_:y1 _:y2)
_:y1 xsd:minInclusive "0"^^xsd:integer
_:y2 xsd:maxInclusive "150"^^xsd:integer
```

  - punning (metamodeling), e.g.:

```
ex:John rdf:type ex:Father
ex:Father rdf:type ex:SocialRole
```

25 UNIVERSITY OF ABERDEEN

New Expressiveness in OWL 2

CSWS2010

- New expressive power on properties
  - qualified cardinality restrictions, e.g.:

```
_:x rdf:type owl:Restriction
_:x owl:onProperty foaf:know
_:x owl:minQualifiedCardinality "2"^^xsd:nonNegativeInteger
_:x owl:onClass Scottish
```

  - property chain inclusion axioms, e.g.:

```
foaf:nick owl:propertyChainAxiom (foaf:holdsAccount sioc:name)
```

  - local reflexivity restrictions, e.g.:

```
_:x rdf:type owl:Restriction
_:x owl:onProperty like
_:x owl:hasSelf "true"^^xsd:boolean
```
[for narcissists]

  - reflexive, irreflexive, symmetric, and antisymmetric properties, e.g.:

```
foaf:know rdf:type owl:ReflexiveProperty
sioc:has_parent rdf:type owl:IrreflexiveProperty
```

  - disjoint properties, e.g.:

```
sioc:parent_of owl:propertyDisjointWith sioc:has_parent
```

  - keys, e.g.:

```
foaf:OnlineAccount owl:hasKey
(foaf:accountName foaf:accountServiceHomepage)
```

New Expressiveness in OWL 2

CSWS2010

- **Syntactic sugar** (make things easier to say)
  - Disjoint classes, e.g.:

```
_:x rdf:type owl:AllDisjointClasses
_:x owl:members (boy girl)
```

  - Disjoint unions, e.g.:

```
child owl:disjointUnionOf (boy girl)
```

  - Negative assertions, e.g.:

```
_:x rdf:type owl:NegativePropertyAssertion
_:x owl:sourceIndividual John
_:x owl:assertionProperty foaf:know
_:x owl:targetIndividual Mary
```

27

UNIVERSITY OF ABERDEEN

OWL 2 DL

CSWS2010

- R often used for ALC extended with property chain inclusion axioms
  - following the notion introduced in RIQ [Horrocks and Sattler, 2003]
  - including transitive property axioms
- Additional letters indicate other extensions, e.g.:
  - S for property characteristics (e.g., reflexive and symmetric)
  - O for **nominals**/singleton classes
  - I for inverse roles
  - Q for qualified number restrictions
- property characteristics (S) + R + nominals (O) + inverse (I) + qualified number restrictions(Q) = SROIQ
- SROIQ [Horrocks et al., 2006] is the basis for OWL 2 DL

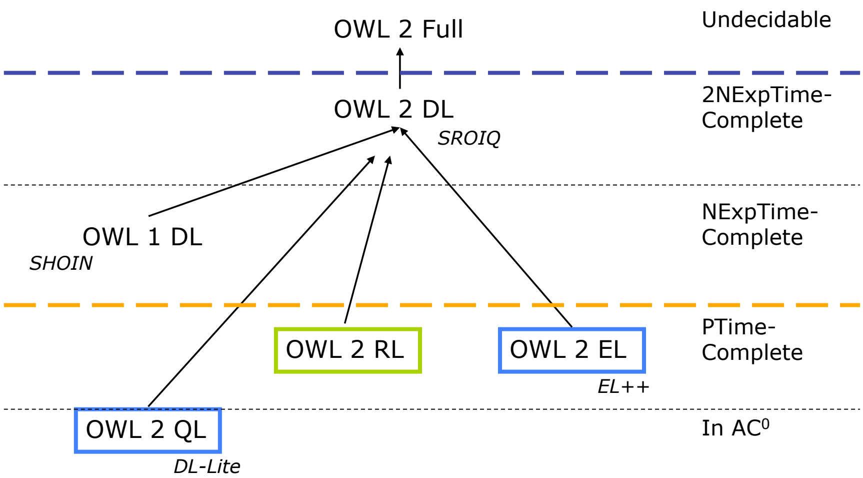
28

UNIVERSITY OF ABERDEEN

OWL 2 Profiles and Reasoning Services

- Rationale:
  - Tractable
  - Tailored to specific reasoning services
- Popular reasoning services
  - TBox reasoning: OWL 2 EL
  - ABox reasoning with rule engines: OWL 2 RL
  - Query answering: OWL 2 QL
- Specification: http://www.w3.org/TR/2009/CR-owl2-profiles-20090611/

The family tree



OWL 2 EL

CSWS2010

- A (near maximal) fragment of OWL 2 such that
  - Satisfiability checking is in PTime (**PTime-Complete**)
  - Data complexity of query answering also PTime-Complete
- Based on **EL** family of description logics [Baader et al. 2005]
- Can exploit **saturation** based reasoning techniques
  - Computes complete classification in "one pass"
  - Computationally optimal (PTime for EL)
  - Can be extended to Horn fragment of OWL DL [Kazakov 2009]

31 UNIVERSITY OF ABERDEEN

Saturation-based Technique (basics)

CSWS2010

- Normalise ontology axioms to standard form:

$$A \sqsubseteq B \quad A \sqcap B \sqsubseteq C \quad A \sqsubseteq \exists R.B \quad \exists R.B \sqsubseteq C$$

- Saturate using inference rules:

$$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C} \qquad \frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

- Extension to Horn fragment requires (many) more rules

32 UNIVERSITY OF ABERDEEN

Saturation-based Technique (basics)

CSWS2010

Example:

ex:Performer ≡ foaf:Person ⊓ ∃mo:perform_in.mo:Performance
mo:MusicArtist ≡ foaf:Person ⊓ ∃mo:perform_in. ex:MusicPerformance
ex:MusicPerformance ⊑ mo:Performance

ex:Performer ⊑ foaf:Person
ex:Performer ⊑ ∃mo:perform_in.mo:Performance
∃mo:perform_in.mo:Performance ⊑ PP
foaf:Person ⊓ PP ⊑ ex:Performer

mo:MusicArtist ⊑ foaf:Person
mo:MusicArtist ⊑ ∃mo:perform_in.ex:MusicPerformance
∃mo:perform_in.ex:MusicPerformance ⊑ PMP
foaf:Person ⊓ PMP ⊑ mo:MusicArtist

ex:MusicPerformance ⊑ mo:Performance

33 UNIVERSITY OF ABERDEEN

Saturation-based Technique (basics)

CSWS2010

Example:

ex:Performer ⊑ foaf:Person
ex:Performer ⊑ ∃mo:perform_in.mo:Performance
∃mo:perform_in.mo:Performance ⊑ PP
foaf:Person ⊓ PP ⊑ ex:Performer

mo:MusicArtist ⊑ foaf:Person
mo:MusicArtist ⊑ ∃mo:perform_in.ex:MusicPerformance
∃mo:perform_in.ex:MusicPerformance ⊑ PMP
foaf:Person ⊓ PMP ⊑ mo:MusicArtist

ex:MusicPerformance ⊑ mo:Performance

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

mo:MusicArtist ⊑ PP

34 UNIVERSITY OF ABERDEEN

Saturation-based Technique (basics)

Example:

ex:Performer ⊑ foaf:Person
ex:Performer ⊑ ∃mo:porform_in.mo:Performance
∃mo:porform_in.mo:Performance ⊑ PP
foaf:Person ⊓ PP ⊑ ex:Performer

mo:MusicArtist ⊑ foaf:Person
mo:MusicArtist ⊑ ∃mo:porform_in.ex:MusicPerformance
∃mo:porform_in.ex:MusicPerformance ⊑ PMP
foaf:Person ⊓ PMP ⊑ mo:MusicArtist

ex:MusicPerformance ⊑ mo:Performance

$$\frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

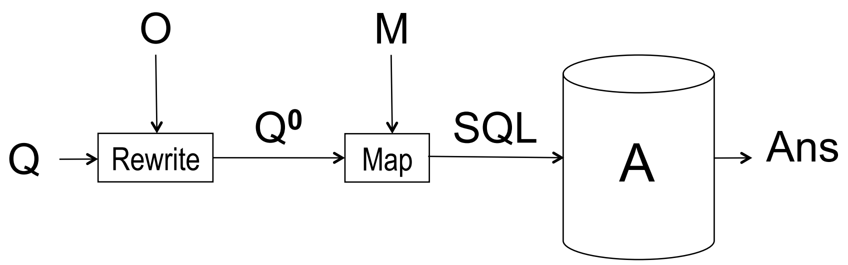
mo:MusicArtist ⊑ PP
mo:MusicArtist ⊑ ex:Performer

OWL 2 QL

- A (near maximal) fragment of OWL 2 such that
  - Data complexity of conjunctive query answering in AC^0
- Based on **DL-Lite** family of description logics [Calvanese et al. 2005; 2006; 2008]
- Can exploit **query rewriting** based reasoning technique
  - Computationally optimal
  - Data storage and query evaluation can be delegated to standard RDBMS
  - Novel technique to prevent exponential blowup produced by rewritings [Kontchakov et al. 2010, Rosati and Almatelli 2010]
  - Can be extended to more expressive languages (beyond AC^0) by delegating query answering to a Datalog engine [Perez-Urbina et al. 2009]

Query Rewriting Technique (basics)

- Given ontology O and query Q, use O to rewrite Q as Q^0 s.t., for any set of ground facts A:
  - ans(Q, O, A) = ans(Q^0, ;, A)
- Use (GAV) mapping M to map Q^0 to SQL query



Query Rewriting Technique (basics)

- Given ontology O and query Q, use O to rewrite Q as Q^0 s.t., for any set of ground facts A:
  - ans(Q, O, A) = ans(Q^0, ;, A)
- Use (GAV) mapping M to map Q^0 to SQL query
- Resolution based query rewriting
  - **Clausify** ontology axioms
  - **Saturate** (clausified) ontology and query using resolution
  - **Prune** redundant query clauses

Query Rewriting Technique (basics)

CSWS2010

- Example:

ex:Performer $\sqsubseteq$ ∃mo:porform_in.mo:Performance

mo:MusicArtist $\sqsubseteq$ ex:Performer

mo:porform_in(x,f(x)) ← ex:Performer(x)
mo:Performance(f(x)) ← ex:Performer(x)
ex:Performer(x) ← ex:MusicArtist (x)

Q(x) ← mo:porform_in(x,y) ∧ mo:Performance(y)

39

UNIVERSITY OF ABERDEEN

Query Rewriting Technique (basics)

CSWS2010

- Example:

mo:porform_in(x,f(x)) ← ex:Performer(x)
mo:Performance(f(x)) ← ex:Performer(x)
ex:Performer(x) ← ex:MusicArtist (x)

Q(x) ← mo:porform_in(x,y) ∧ mo:Performance(y)
Q(x) ← ex:Performer (x) ∧ mo:Performance(f(x))
Q(x) ← mo:porform_in(x, f(x)) ∧ ex:Performer(x)

40

UNIVERSITY OF ABERDEEN

Query Rewriting Technique (basics)

CSWS2010

- Example:

mo:perform_in($x,f(x)$) ← ex:Performer(x)
mo:Performance($f(x)$) ← ex:Performer(x)
ex:Performer(x) ← ex:MusicArtist (x)

$Q(x)$ ← mo:perform_in(x,y) ∧ mo:Performance(y)
$Q(x)$ ← ex:Performer (x) ∧ mo:Performance($f(x)$)
$Q(x)$ ← mo:perform_in(x, $f(x)$) ∧ ex:Performer(x)
$Q(x)$ ← ex:Performer(x)
$Q(x)$ ← ex:MusicArtist (x)

41

UNIVERSITY OF ABERDEEN

Query Rewriting Technique (basics)

CSWS2010

- Example:

mo:perform_in($x,f(x)$) ← ex:Performer(x)
mo:Performance($f(x)$) ← ex:Performer(x)
ex:Performer(x) ← ex:MusicArtist (x)

$Q(x)$ ← mo:perform_in(x,y) ∧ mo:Performance(y)
~~$Q(x)$ ← ex:Performer (x) ∧ mo:Performance($f(x)$)~~
~~$Q(x)$ ← mo:perform_in(x, $f(x)$) ∧ ex:Performer(x)~~
$Q(x)$ ← ex:Performer(x)
$Q(x)$ ← ex:MusicArtist (x)

- For OWL2 QL, result is a union of conjunctive queries

$Q(x)$ ← (mo:perform_in(x,y) ∧ mo:Performance(y)) ∨ ex:Performer(x) ∨ ex:MusicArtist (x)

42

UNIVERSITY OF ABERDEEN

Query Rewriting Technique (basics)

- Data can be stored/left in **RDBMS**
- Relationship between ontology and DB defined by **mappings**, e.g.:
  - ex:Performer ↔ SELECT NAME FROM PERFORMER
  - ex:MusicArtist ↔ SELECT NAME FROM MUSICARTIST
  - mo:porform_in ↔ SELECT P1NAME AND P2NAME FROM PERFORM
  - ex:Performance ↔ SELECT PNAME FROM PERFORMANCE
- UCQ translated into **SQL query**:

$Q(x) \leftarrow (\text{mo:perform\_in}(x,y) \wedge \text{mo:Performance}(y)) \vee \text{ex:Performer}(x) \vee \text{ex:MusicArtist}(x)$

⇣

SELECT NAME FROM PERFORMER UNION
SELECT NAME FROM MUSICARTIST UNION
SELECT PNAME FROM PERFORM, PERFORMANCE WHERE P2NAME=PNAME

Outline

- Session 1: Introduction to Linked Data and OWL 2
  - Now: Linked Data and Ontology
  - Overview: from OWL 1 to OWL 2
  - Reasoning services in OWL 2
- Session 2: Quality Guaranteed Approximations
  - Challenges
  - Soundness preserving and Completeness preserving approximations
  - A hands-on session

DL Ontology

CSWS2010

- A TBox (Terminonagy Box) is a set of “schema” axioms (sentences), e.g.:

  $\{\exists_{\geq 50}$ good_friend $\sqsubseteq$ Popular,
  good_friend $\sqsubseteq$ friend$\}$

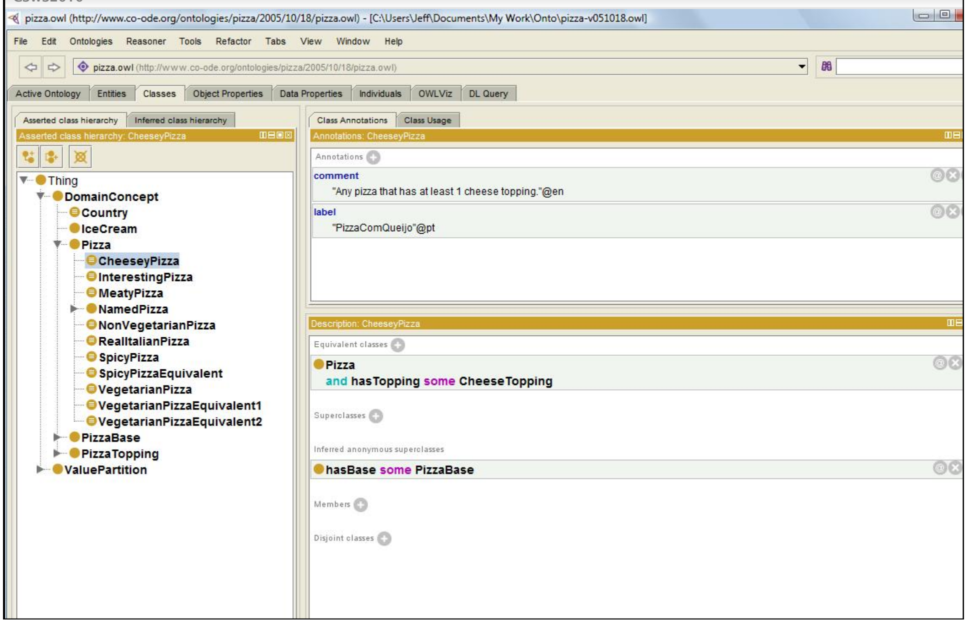
  - i.e., a **background theory for the vocabulary**
- An ABox (Assertion Box) is a set of “data” axioms (ground facts), e.g.:

  {good_friend (John,Kate)}
- An NBox (Negation As Failure Box)
  - See the presentation of “Closed World Reasoning for OWL2 with Negation As Failure”

45 UNIVERSITY OF ABERDEEN

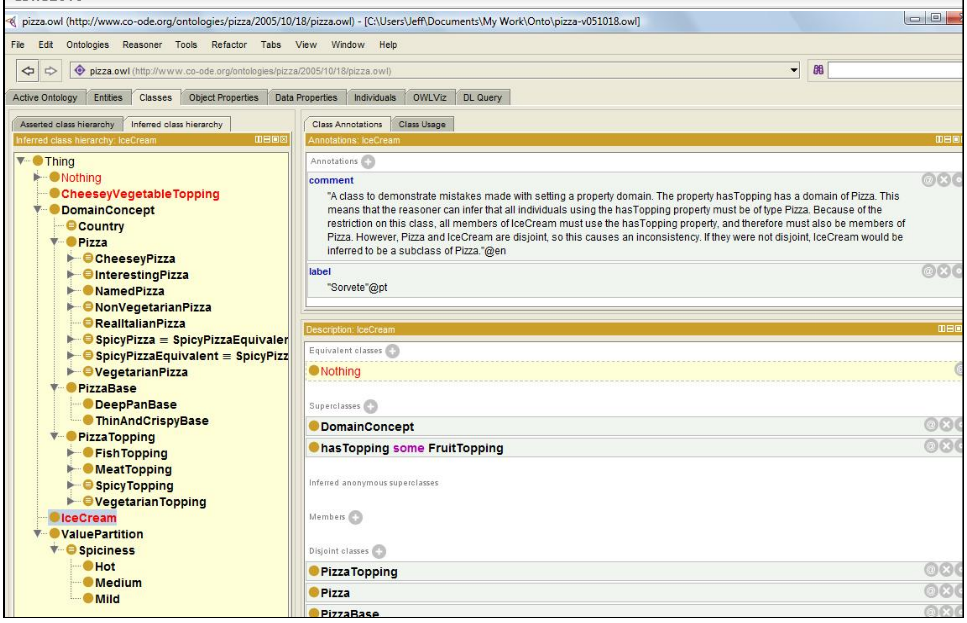
DL Ontology (II)

CSWS2010



Classification

CSWS2010



Query Answering

CSWS2010

Example:

```
SELECT ?X
FROM <http://example.org/person.owl>
WHERE {?X friend ?Y .}

SELECT ?X
FROM <http://example.org/person.owl>
WHERE {?X friend ?Y .
       ?Y rdf:type Popular .}
```


UNIVERSITY OF ABERDEEN

Non-standard Reasoning Services

- Query answering
  - X:q :- conj(X;Y)
    - Head X:q
    - Body conj(X;Y)
    - Distinguished variables X
    - Non-distinguished variables Y
  - Given a conjunctive query X:q :- conj(X;Y) and an ontology O, return all W (tuples of constants) that substitute X such that O |= conj(W;Y).
- Example: q(x) :- hasFriend(x,y) $\wedge$ Popular(y)

Why Reasoning is Challenging?

Challenge: Ontology Classification

CSWS2010

- Classification is a complex reasoning service
  - Compute the class hierarchy
- Classification can be reduce to subsumption checking
  - between every pair of classes
  - which can be further reduced to class unsatisfiability checking
- For example, if an ontology consists of 100,000 classes
  - then a naïve implementation needs
  - 100,000 * 100,000 class unsatisfiability checking

UNIVERSITY of ABERDEEN

Challenge: Ontology Classification (II)

CSWS2010

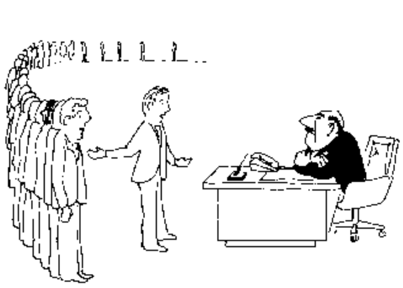
- General class axioms $C \sqsubseteq D$ are widely used
  - such as the domain constraint
    - ObjectProperty(hasF domain (Person))
    - $\exists hasF \sqsubseteq Person$
- Expansion rules for general class axioms $C \sqsubseteq D$
  - If a new node y is added, then add a general class description $D \sqcup \neg C$ in L(y) for each general class axiom $C \sqsubseteq D$
- Case study: if an ontology consists of 100 general class axioms
  - then theoretically there are 2^{100} possible expansions for each node

UNIVERSITY of ABERDEEN

Challenge: Query Answering with DL is Hard

CSWS2010

- **Open problem**: whether query answering in OWL 2 DL is decidable
- And it is known to be hard …



I can't find any algorithms, but neither can all these famous people.

[Garey & Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, 1979.]

53 UNIVERSITY OF ABERDEEN

Challenge: Query Answering with DL is Hard

CSWS2010

- **Open problem**: whether query answering in OWL 2 DL is decidable
- And it is known to be hard
  - SHIQ with queries allowing only simple properties in [Ortiz et al., 2006]
    - Data complexity: co-NP-complete
  - SHIQ [Glimm et al., 2007]:
    - Combined complexity: 2EXPTIME
    - Data complexity: co-NP-complete
  - In fact, co-NP-complete even for **very small fragments** of SHIQ [Calvnese et al., 2006], e.g. allowing
    - Full negations in the left hand side of class inclusions, or
    - Disjunctions or universal restrictions in the right hand side of class inclusions

54 UNIVERSITY OF ABERDEEN

CSWS2010

Approach: Approximation

UNIVERSITY OF ABERDEEN

Why approximations?

CSWS2010

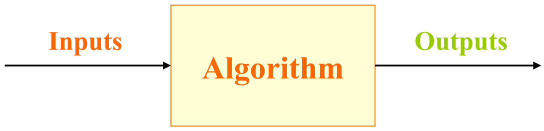
- **In short, because**
  - there exist no algorithms for query answering in OWL 2 DL

56

UNIVERSITY OF ABERDEEN

What to approximate?

- **Inputs**
  - query
  - ontology
- **Algorithm**



Quality of approximations

- **Outputs**
  - soundness preserving
  - completeness preserving

Syntactic vs. Semantic Approximations

CSWS2010

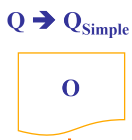
- Syntactic approximations might not guarantee either soundness or completeness
- Example (syntactic approximation)
  - An OWL DL source ontology O = {$\exists_{\geq 50}$ good_friend ⊑ Popular, good_friend ⊑ friend, good_friend(John,Kate)}
  - A syntactic approximation in ALC : O' = {$\exists_{\geq 1}$ good_friendv Popular, good_friend(John,Kate)}
  - O' entails Popular(John) but O does not
  - O entails friend(John,Kate) but O' does not

59 UNIVERSITY of ABERDEEN

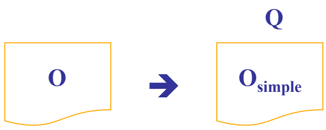
Existing Work

CSWS2010

- Approximation of queries [Struckenschmidt and van Harmelen, 2002; Wache et al., 2005]



- Approximations of ontologies [DLDB, 2004; OWLJessKB, 2003]



60 UNIVERSITY of ABERDEEN

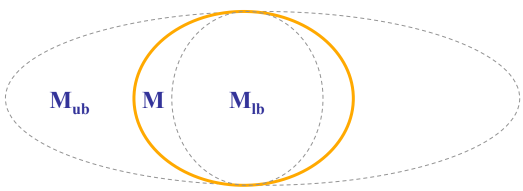
Summary: Existing Work

- Most based on syntactic approximations
- Few providing proofs on soundness or completeness

Knowledge Compilations

CSWS2010

- Knowledge Compilation [Selman and Kautz, 1996]
  - Lower-bound KB_{lb}: $M(KB_{lb}) \subseteq M(KB)$ (hence $KB_{lb} \models KB$)
    - Greatest lower-bound: no KB' such that $M(KB_{lb}) \subseteq M(KB') \subseteq M(KB)$
  - Upper-bound KB_{ub}: $M(KB) \subseteq M(KB_{ub})$ (hence $KB \models KB_{ub}$)
    - Least upper-bound: no KB' such that $M(KB) \subseteq M(KB') \subseteq M(KB_{ub})$

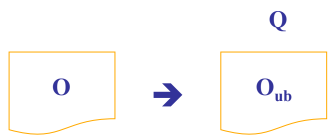


UNIVERSITY OF ABERDEEN

Semantic Approximation [Pan and Thomas, 2007]

CSWS2010

- Idea: to approximate a source ontology O (in more expressive L_S) with its (least) upper bound O_{ub} (in less expressive L_T)
  - O |= O_{ub} (soundness guaranteed)



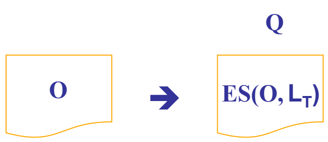
63

UNIVERSITY of ABERDEEN

Entailment Set

CSWS2010

- We use N_C, N_P and N_I to denote the sets of class names, property names and individual names used in O
  - N_C, N_P and N_I are assumed to be finite
- Entailment set ES(O, L_T) of O in L_T
  - The set of **all** L_T axioms that are entailed by O under N_C, N_P and N_I
- **Lemma1**: ES(O, L_T) is the least upper bound (LUB) of O w.r.t. L_T



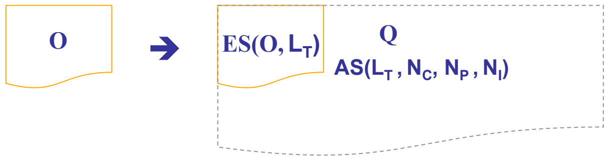
64

UNIVERSITY of ABERDEEN

Axiom Set

CSWS2010

- Axiom set $AS(L_T, N_C, N_P, N_I)$ of L_T w.r.t. N_C, N_P and N_I
  - The set of **all** L_T axioms that can be constructed by using only vocabulary in N_C, N_P and N_I
- We can calculate $ES(O, L_T)$ from $AS(L_T, N_C, N_P, N_I)$



65 UNIVERSITY of ABERDEEN

Choosing L_T for Query Answering

CSWS2010

- **Why** OWL 2 QL [Calvanese et al., 2005, 2006, 2008]

1. Data complexity of query answering: AC^0
2. Lemma2: The size of $AS(\text{OWL2-QL}, N_C, N_P, N_I)$ is polynomial w.r.t. the sizes of N_C, N_P and N_I.
3. Existing OWL 2 QL reasoning and query engines, such as
   - QuOnto [Acciarri et al., 2005]
   - Quill (QL reasoner in TrOWL) [Pan and Thomas, 2007; Thomas et. al., 2010]

66 UNIVERSITY of ABERDEEN

Evaluation: Constructing ES (2007)

- Wine ontology (79 named classes, 12 individual properties and 206 individuals):
  - Without optimisation: did not finish in an hour
  - With optimisation: 4m and 14s
- Lehigh University Benchmark: LUBM (50 universities) with more than 6,800,000 individuals
  - Without optimisation: ran out of memory
  - With optimisation: about 2h

Soundness and Completeness

- Theorem 1: Given an ontology O, a conjunctive query q(X) and an evaluation [X→S], if ES(O, OWL 2 QL) $\models q_{[X \rightarrow S]}$, then O $\models q_{[X \rightarrow S]}$.
- However, in general, it could be incomplete

Soundness and Completeness (II)

CSWS2010

- Theorem 2: Given an ontology O_S, a database-style conjunctive query q(X) without non-distinguished variables and an evaluation [X→S], ES(O_S, OWL 2 QL) $\models q_{[X \rightarrow S]}$ iff $O_S \models q_{[X \rightarrow S]}$.
- Remark: disallowing non-distinguished variables seems to be the price to pay, in order to achieve complete results efficiently
  - Racer does not allow them in their nRQL [Haarslev et al., 2004]
  - KAON2 [Hustadt et al.,2004] does not allow them either

69 UNIVERSITY OF ABERDEEN

Evaluation: Query Answering

CSWS2010

- Lehigh University Benchmark
- OWL 2 DL/ SHIQ reasoners [Motik and Sattler, 2006]
  - Pellet and Racer: unable to complete queries 1, 2 and 3 when more than 2 universities
  - KAON2 can answer queries for up to 4 universities

[Q1(?x) :- GraduateStudent(?x) ∧ takesCourse(?x,U0D0GraduateCourse0)

Q2(?x,?y,?z) :- GraduateStudent(?x) ∧ University(?y),Department(?z) ∧ memberOf(?x,?z) ∧ subOrganisationOf(?z,?y) ∧ undergraduateDegreeFrom(?x,?y)

Q3(?x) :- Publication(?x) ∧ publicationAuthor(?x,U0D0AssistentProfessor0)]

70 UNIVERSITY OF ABERDEEN

Evaluation: Query Answering (II)

CSWS2010

- Syntactic approximation systems [Guo et al., 2004]
  - OWLJessKB can only handle 1 university
    - All results are covered
    - Returned incorrect answers for queries 4, 6, 8 and 12
  - DLDB can handle 50 universities on all queries but query 2
    - Returned incomplete answers for queries 11 and 13

[Q11(?x) :- ResearchGroup(?x) ∧ subOrganisationOf(?x,University0)

Q12(?x,?y) :- Chair(?x) ∧ Department(?y) ∧ worksFor(?x,?y), subOrganisationOf(?y,University0)

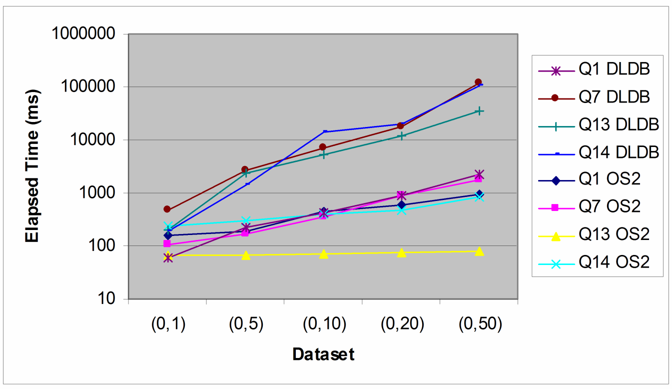
Q13(?x) :- Person(?x) ∧ hasAlumnus(University0,?x)]

71 UNIVERSITY OF ABERDEEN

Evaluation: Query Answering (III)

CSWS2010

- Quill/ONTOSEARCH2 can handle at least 50 universities
  - Sound and complete results for all queries
- DLDB vs. Quill/ONTOSEARCH2



ITY OF ABERDEEN

Where do we go from here?

Completeness Guaranteed Semantic Approximations [Pan et al. 2009]

- Main idea: Approximating queries

$Q \rightarrow Q_{Simple}$

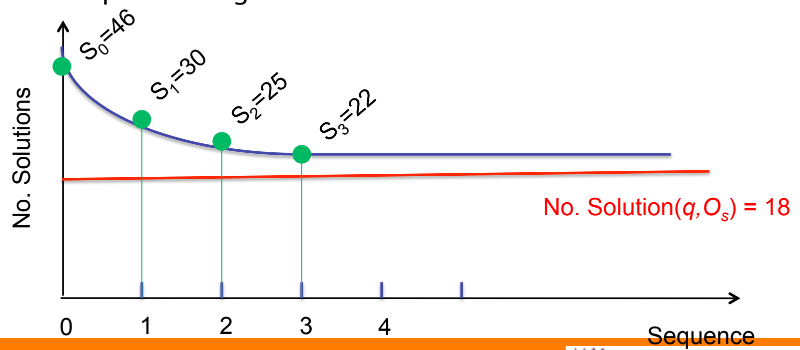
O

- Example: q(x) :- Popular(x) ∧ good_friend(x,y) ∧ has_pet(y,z)
  - can be approximated to q'(x) :- Popular(x)
  - This is a completeness guaranteed approximation, even w.r.t. ES(O, OWL 2 QL)

Anytime Reasoning

CSWS2010

- Given an ontology O and a query q, we offer two answer sets, based on ES(O, OWL2-QL) and its extentions
  - S: soundness guaranteed
  - Ci: completeness guaranteed



75

UNIVERSITY OF ABERDEEN

Motivation: Syntactic Approximation

CSWS2010

- Why SROIQ (OWL2 DL) to EL++ (OWL2 EL)?
  - Minor syntactic gap results in major complexity difference
    - From most expressive/complex to most efficient for TBox
  - Using approximation to bridge the gap

DL SROIQ	DL EL++
$\top \mid \bot \mid A \mid C \sqcap D \mid \exists R.C \mid \{a\} \mid \boxed{\neg C \mid \geq nR.C \mid \exists R.Self}$	$\top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\}$
$C \sqsubseteq D \qquad r \sqsubseteq s, r_1 \circ \ldots \circ r_n \sqsubseteq s$	
$a : C \qquad (a, b) : r$	
2NEXPTIME-complete	PTIME-complete

UNIVERSITY OF ABERDEEN

Big Picture

CSWS2010

- We provide faithful syntactic approximation for OWL2 DL TBox reasoning
  - Soundness preserving
  - Practically close to complete
- Challenge
  - removing axioms with non-OWL2 EL descriptions could lead to low recall

77 UNIVERSITY of ABERDEEN

How Does it Work

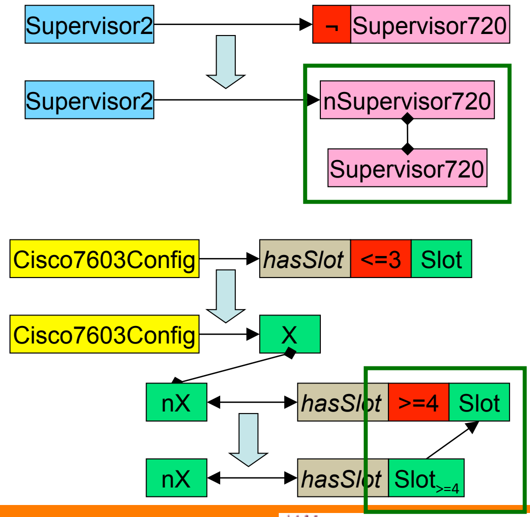
CSWS2010

- TBox approximation
  - Directly represent non-OWL2-EL concepts with fresh named concepts
    - E.g., ∀r.C subClassOf D ➔ $A_{\forall r.C}$ subClassOf D
  - Maintain semantic relations for these named concepts
    - *complementary relations*
    - *cardinality relations*
- ABox approximation
  - Internalise ABox into TBox
    - E.g., C(a) ➔ {a} subClassOf C
    - r(a,b) ➔ {a} subClassOf ∃r.{b}, …
- Reasoning
  - Using additional completion rules to recover the semantics

78 UNIVERSITY of ABERDEEN

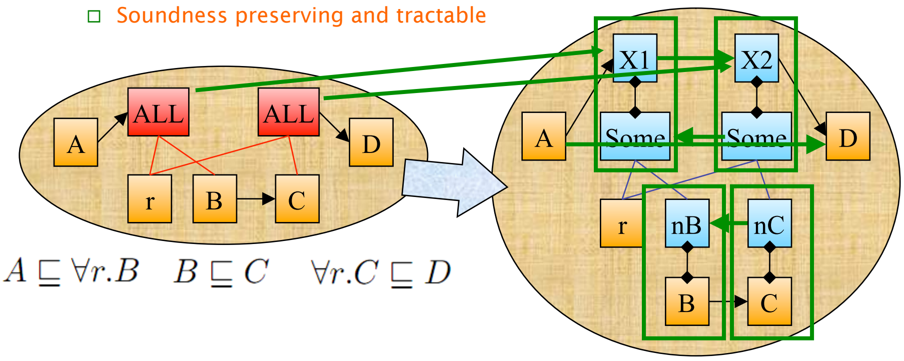
Example: How Does it Work

- Syntactic approximation
  - Directly represent non-OWL2-EL concept with fresh names
  - Maintain complementary relations in complementary table (CT)
  - Maintain cardinality relations in cardinality table (QT)
  - Internalise ABox into TBox
  - …



Example: How Does it Work (II)

- Completion Rules (16 rules), e.g.
  - Handling complement
    - E.g. A subClassOf B => ¬B subClassOf ¬A
  - Handling cardinality
    - E.g. A subClassOf >= 3 r. B => A subClassOf >= 2 r. B
  - Soundness preserving and tractable



$A \sqsubseteq \forall r.B \quad B \sqsubseteq C \quad \forall r.C \sqsubseteq D$

Demo Plan 1

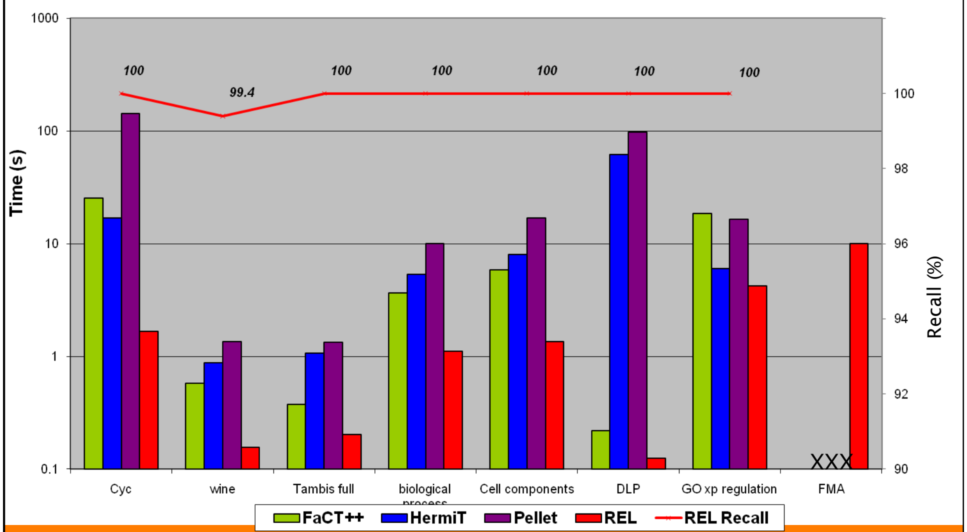
CSWS2010

- Oxford Benchmark and some related ontologies
  - 1. classification of DLP397 in Pellet: time consuming
  - 2. classification of DLP397 in REL: efficient
  - 3. classification of Trento1 in REL: efficient
  - 4. classification of Trento1 in Pellet: time consuming

UNIVERSITY OF ABERDEEN

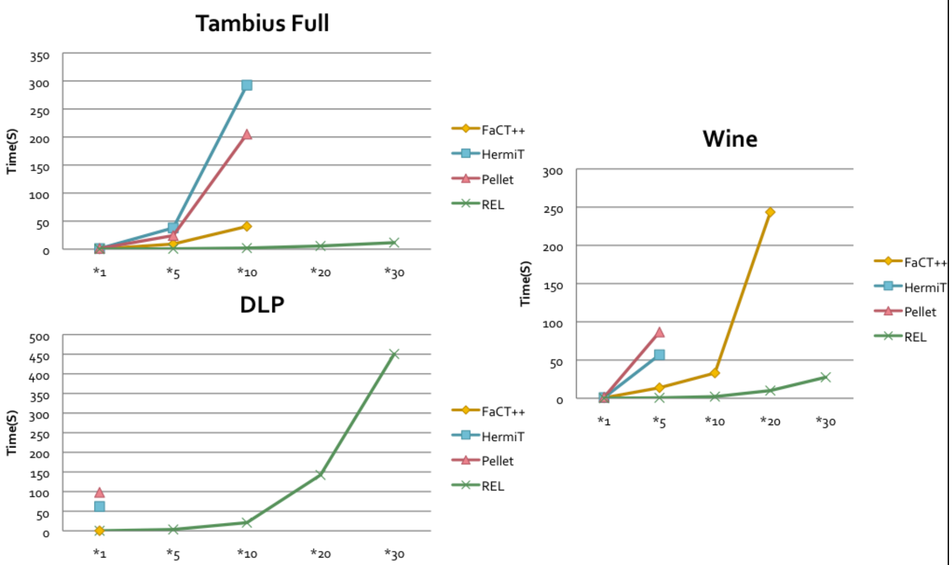
Evaluations for the Oxford Benchmarks (REL – EL reasoner in TrOWL)

CSWS2010



UNIVERSITY OF ABERDEEN

Evaluation: Scalability Comparison



Evaluation: Constructing ES II

- Wine ontology (79 named classes, 12 individual properties and 206 individuals):
  - Without optimisation: did not finish in an hour
  - With optimisation: 4m and 14s
  - With REL: 12s (completeness rate: about 95%)
- Lehigh University Benchmark: LUBM (50 universities) with more than 6,800,000 individuals
  - Without optimisation: ran out of memory
  - With optimisation: about 2h
  - With REL: about 40m (complete)

Demo Plan 2

CSWS2010

- Physical Device Configuration Ontologies
  - 1. classification of Pellet: time consuming
  - 2. classification of REL: efficient
  - 3. inconsistency justification of REL: able to detect multiple sources of inconsistency
  - 4. inconsistency justification of Pellet: may yield incorrect results

85 UNIVERSITY OF ABERDEEN

TrOWL

CSWS2010

TrOWL: Tractable reasoning infrastructure for OWL 2 **[Thomas et al. 2010]**

- Quality guaranteed transformations (such as modularisation, approximations, forgetting)
  - OWL 2 DL -> OWL 2 QL (semantic approximation)
  - OWL 2 DL -> OWL 2 EL (syntactic approximation)
- Ontology reasoners (supporting OWL API and Protégé 4)
  - TrOWL Quill (OWL 2 QL / DL)
  - TrOWL REL (OWL 2 EL / DL)
- Explanation
  - Justification
  - Explanation on transformation
- ONTOSEARCH2 serves as its front end
  - ontological search engine
  - supporting keyword plus entailment search **[PSSTT-WWW2008]**

TrOWL www.trowl.eu

86 86 UNIVERSITY OF ABERDEEN

Take Home Message

- *Q: How to provide tractable and scalable reasoning services for expressive ontology languages like OWL2-DL?*

  A: Divide and conquer approaches and **faithful approximations** can help

Questions?

- Session 1: Introduction to Linked Data and OWL 2
  - Now: Linked Data and Ontology
  - Overview: from OWL 1 to OWL 2
  - Reasoning services in OWL 2
- Session 2: Quality Guaranteed Approximations
  - Challenges
  - Soundness preserving and Completeness preserving approximations
  - A hands-on session

Acknowledgement

- Aidan Hogan
- Ian Horrocks
- Nophadol Jekjantuk
- Axel Polleres
- Yuan Ren
- Edward Thomas
- Yuting Zhao

UNIVERSITY OF ABERDEEN