Semantic reasoning for dynamic knowledge bases

Lionel Médini
M2IA – Knowledge Dynamics
2021
Outline

• Summary
  – Logics
  – Semantic Web Languages
  – Reasoning
• Web-based reasoning techniques
• Reasoning using SemWeb languages expressivity
• General aspects of reasoning and calculability
• Reasoning on dynamic KBs
• Conclusion
Summary

• Formal logics

La logique est à l'origine la recherche de règles générales et formelles permettant de distinguer un raisonnement concluant de celui qui ne l'est pas. [Wikipédia-Fr]

– First Order Logic
– Propositional Logic
– Predicate logic
– Modal Logics
– Description logics
– Fuzzy logic...
Summary

• Description Logics (DL)

DL is a family of formal knowledge representation languages. Many DLs are more expressive than propositional logic but less expressive than first-order predicate logic. In contrast to the latter, the core reasoning problems for DLs are (usually) decidable, and efficient decision procedures have been designed and implemented for these problems. Wikipédia-En

— DLs model concepts, roles and individuals, and their relationships

— The fundamental modeling concept of a DL is the axiom - a logical statement relating roles and/or concepts.
Summary

• Description Logics (DL)
  – DLs provide various characteristics, corresponding to different types of *expressivity*
  – Each feature indicates “what you can say” with it
  – 3 basic logics:
    • Attributive Language (used in ontologies)
    • Frame-based description language
    • Existential language
  – A set of extensions that provide specific constructs
Summary

• The Semantic Web
  – RDF
    • Triple: subject, predicate, object
    • Nodes
      – Subject, object
      – Possible types: IRI, literal, blank node
    • Predicate: link

Dogs eat Meat
Summary

• The Semantic Web
  – RDF
    • Graph: set of (possibly interrelated) triples
Summary

• The Semantic Web
  – RDF
    • Blank node: resource without a definite IRI

<table>
<thead>
<tr>
<th>?</th>
<th>eat</th>
<th>Meat</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasSpecies</td>
<td></td>
<td>hasSpecies</td>
</tr>
<tr>
<td>Cows</td>
<td></td>
<td>Dogs</td>
</tr>
</tbody>
</table>
Summary

• The Semantic Web
  – RDF
    • Containers (Bag, Seq, Alt): provide access to a set of similar resources

```
let rdf:Bag
  rdf:_1
  rdf:_2
Dogs
  Cows
  Meat
  eat
```
Summary

• The Semantic Web
  – RDF
    • Resource: basically anything in RDF
Summary

- The Semantic Web
  - RDF
    - Statement: talking about a triple

Dogs ➔ Meat ➔ 90%

Dogs eat Meat

hasLikelihood
Summary

• The Semantic Web
  – Structured Graph: RDF-S
  • Schema: Class, Property, hierarchy, data
  • Constraints: domain, range
Summary

• The Semantic Web
  – Ontologies: OWL
    • Web Ontology Language
    • 2 versions
      – OWL (1), February 2004: Lite, DL, Full
      – OWL2, December 2012: 3 profiles \( \rightarrow \) EL, RL, QL
    • Adds expressivity to the couple RDF-S + RDF
    • Allows defining vocabularies
    • Based on Description Logics
Summary

• The Semantic Web
  – Ontologies: OWL
    • OWL ontology components
      – Terminology (TBox): domain vocabulary
      – Assertions (ABox): to store actual data
      – Roles (RBox): term articulation
    • Vocabulary components
      Classes          Roles
      Individuals     Object properties
      Literals        Data properties
Summary

• The Semantic Web
  – Ontologies: OWL
    • Supplementary semantic constructs
      – Expressions
        » Object Property
        » Data ranges
        » Class expressions
      – Axioms
        » Class expression axioms
        » Object properties
        » Data properties
        » Datatypes
        » Keys
        » Data assertions

An OWL 2 ontology $O$ is satisfied in an interpretation $I$ if all axioms in the axiom closure of $O$ are satisfied in $I$. 
Summary

• The Semantic Web
  – Ontologies: OWL
    • Cheatsheet reference for OWL 2 DL
      – Summary of OWL constructs: https://www.w3.org/TR/owl-quick-reference/
      – Detailed definitions: https://www.w3.org/TR/owl2-syntax/
      – Formal definitions: https://www.w3.org/TR/owl-direct-semantics/
      – DLs names vs. expressivity levels: https://en.wikipedia.org/wiki/Description_logic#Nomenclature
Summary

• The Semantic Web
  – OWL 2 overview

• 5 serialization syntaxes

• 2 semantics
  – Direct: « OWL 2 DL » (see restrictions)
  – RDF: « OWL 2 Full » (no restriction)

Source: https://www.w3.org/TR/owl2-overview/
Summary

• The Semantic Web
  – OWL2 profiles
    • Expression Language (EL)
      – For classification tasks
      – Different expressivity levels
      – Query rewriting (SPARQL -> SPARQL)
    • Query Language (QL)
      – For querying large datasets
      – Relies on an internal RDB
      – Query rewriting (SPARQL -> SQL)
    • Rule Language (RL)
      – For entailing facts using rules
      – More to come...
Summary

• Ontology limitations
  – Made by non-experts
    • inconsistent
  – Automatically generated
    • Not very useful without domain knowledge
  – Heterogeneous
  – Too high complexity
Summary

• The Semantic Web
  – Finally, what is it good for?
    • Describing domain vocabularies
    • Querying structured data sources (aka knowledge bases)
    • Agreeing on a common meaning of types of information
    • Interlinking data sources

  – But can’t we do better?...
Web-based reasoning

• Reasoning
  – The process of logically determining information that has not been explicitly been stated

  – Examples
    • Calculus
    • Syntactic analysis
    • Proofs
    • Verification
    • Inferences
    • Science?...
Web-based reasoning

• Inference
  – The process of entailing a fact based on rules
  – Basis of a reasoning process

Condition (head)                        conclusion (body)
(I finished my homework) AND (the weather is nice) → I can go for a walk
Web-based reasoning

• Examples of inductive (logical) reasoning
  – Satisfiability (SAT)
  – Backward chaining (Prolog)
  – Forward chaining (rules)
  – Saturation (tableaux)
Outline

• Summary

• Web-based reasoning techniques
  – Rule-based reasoning
  – Tableaux-based reasoning
  – Comparison

• Reasoning using SemWeb languages expressivity

• General aspects of reasoning and calculability

• Reasoning on dynamic KBs

• Conclusion
Rule-based reasoning

• Reasoning type
  – Deductive reasoning
  – Forward chaining

• Principles
  – Rules are applied to a set of facts
    • At the beginning: *explicit facts*
  – May infer new facts
    • *Implicit facts*
  – On which the rules are applied
  – Loop until nothing more happens
Rule-based reasoning

• Functioning with RDF
  – Facts = triples
  – Order of rule application must not be relevant

• Usages
  – Datalog: query language for deductive DBs
  – View materialization: data duplication for creating views in DBs
Rule-based reasoning

• Use conjunctive rules
  – Only conjunctions in head clauses
  – To ensure commutativity

• Separate non-commutative operations in different queries
  – Deletions
  – Insertions
Tableaux-based algorithms

- Explore all possible Aboxes (« tableaux »)
- Apply rules to search for unsatisfiabilities
  - Start: tableau is open
  - Apply rules (recursively)
  - If a clash is found: tableau is closed
  - If no clash is found and no more rule can be applied: tableau is complete
- If there is a complete tableau, the Tbox is satisfiable
Reasoning algorithms: Rule-based vs. Tableaux

- Rule-based instantiate variables (if they can)
- Tableaux make assumptions on all possibilities
- Example:

Do I know a woman who knows a man?
Outline

• Summary
• Web-based reasoning techniques
• **Reasoning using SemWeb languages expressivity**
  – RDF
  – RDF-S
  – OWL
• General aspects of reasoning and calculability
• Reasoning on dynamic KBs
• Conclusion
RDF reasoning

• Only about facts, triple structure and datatypes

• RDF entailments
  – Simple entailments: structural matching by renaming blank nodes
  – RDF entailments: interpret RDF vocabulary
RDF reasoning

- Simple entailment rules

<table>
<thead>
<tr>
<th>Rule name</th>
<th>If E contains</th>
<th>then add</th>
</tr>
</thead>
<tbody>
<tr>
<td>sel</td>
<td>uuu aaa xxx .</td>
<td>uuu aaa _:nnn . where _:nnn identifies a blank node allocated to xxx by rule sel or se2.</td>
</tr>
<tr>
<td>se2</td>
<td>uuu aaa xxx .</td>
<td>_:nnn aaa xxx . where _:nnn identifies a blank node allocated to uuu by rule sel or se2.</td>
</tr>
<tr>
<td>lg</td>
<td>uuu aaa lll .</td>
<td>uuu aaa _:nnn . where _:nnn identifies a blank node allocated to the literal lll by this rule.</td>
</tr>
<tr>
<td>gl</td>
<td>uuu aaa _:nnn . where _:nnn identifies a blank node allocated to the literal lll by rule lg.</td>
<td>uuu aaa lll .</td>
</tr>
</tbody>
</table>

Source: Corno & Farinetti: Logic and Reasoning in the Semantic Web
RDF reasoning

• RDF entailment rules

<table>
<thead>
<tr>
<th>Rule name</th>
<th>If E contains</th>
<th>then add</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdf1</td>
<td>uuu aaa yyy .</td>
<td>aaa rdf:type rdf:Property .</td>
</tr>
</tbody>
</table>
| rdf2      | uuu aaa lll .  
where lll is a well-typed XML literal . | _:nnn rdf:type  
rdf:XMLLiteral .  
where _:nnn identifies a blank node allocated to lll by rule lg. |

Source: Corno & Farinetti: Logic and Reasoning in the Semantic Web
RDF reasoning

• RDF axiomatic triples

```
rdf:type        rdf:type        rdf:Property .
rdf:subject     rdf:type        rdf:Property .
rdf:predicate   rdf:type        rdf:Property .
rdf:object      rdf:type        rdf:Property .
rdf:first       rdf:type        rdf:Property .
rdf:rest        rdf:type        rdf:Property .
rdf:value       rdf:type        rdf:Property .
rdf:_1           rdf:type        rdf:Property .
rdf:_2           rdf:type        rdf:Property .
...              rdf:type        rdf:Property .
rdf:nil          rdf:type        rdf:List .
```

Source: Corno & Farinetti: Logic and Reasoning in the Semantic Web
RDF reasoning

• Interpolation Lemma

A set of graphs S entails a graph E if and only if every subgraph of S is an instance of E

– Syntactic description of entailment
– Satisfied by simple entailment rules
RDF reasoning

- RDF entailment lemma

\( S \text{ rdf-entails } E \) if and only if there exists a graph

- that can be derived
  - from \( S \) and the RDF axiomatic triples
  - by the application of rule \( lg \) and the RDF entailment rules

- that entails \( E \)
  - Describes reasoning completeness
  - Satisfied by RDF entailment rules
RDF reasoning

- **Usage**
  - Structural graph mapping
  - Assertions about assertions (reification)

- **Examples**
  - Trust $\rightarrow$ recommendation
  - Comparison $\rightarrow$ adaptation
RDF-S reasoning

• Deduce new information on
  – Hierarchy (subclassOf, subPropertyOf)
  – Individuals (rdf-type)
RDF-S reasoning

• RDF-S entailment rules (1/2)

<table>
<thead>
<tr>
<th>Rule name</th>
<th>If E contains</th>
<th>then add</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs1</td>
<td>uuu aaa lll.</td>
<td>_:nnn rdf:type rdfs:Literal</td>
</tr>
<tr>
<td></td>
<td>where lll is a plain literal (with or without a language tag)</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where _:nnn identifies a blank node allocated to lll by rule rule lg.</td>
</tr>
<tr>
<td>rdfs2</td>
<td>aaa rdfs:domain xxx .</td>
<td>uuu rdf:type xxx .</td>
</tr>
<tr>
<td></td>
<td>uuu aaa yyy .</td>
<td></td>
</tr>
<tr>
<td>rdfs3</td>
<td>aaa rdfs:range xxx .</td>
<td>vvv rdf:type xxx .</td>
</tr>
<tr>
<td></td>
<td>uuu aaa vvv .</td>
<td></td>
</tr>
<tr>
<td>rdfs4a</td>
<td>uuu aaa xxx .</td>
<td>uuu rdf:type rdfs:Resource .</td>
</tr>
<tr>
<td>rdfs4b</td>
<td>uuu aaa vvv .</td>
<td>vvv rdf:type rdfs:Resource .</td>
</tr>
<tr>
<td>rdfs5</td>
<td>uuu rdfs:subPropertyOf vvv .</td>
<td>uuu rdfs:subPropertyOf xxx .</td>
</tr>
<tr>
<td></td>
<td>vvv rdfs:subPropertyOf xxx .</td>
<td></td>
</tr>
</tbody>
</table>

Source: Corno & Farinetti: Logic and Reasoning in the Semantic Web
## RDF-S reasoning

### RDF-S entailment rules (2/2)

<table>
<thead>
<tr>
<th>Rule name</th>
<th>If E contains</th>
<th>then add</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs6</td>
<td>uuu rdf:type rdf:Property .</td>
<td>uuu rdfs:subPropertyOf uuu .</td>
</tr>
<tr>
<td>rdfs7</td>
<td>aaa rdfs:subPropertyOf bbb uuu aaa yyy .</td>
<td>uuu bbb yyy .</td>
</tr>
<tr>
<td>rdfs8</td>
<td>uuu rdf:type rdfs:Class .</td>
<td>uuu rdfs:subClassOf rdfs:Resource .</td>
</tr>
<tr>
<td>rdfs9</td>
<td>uuu rdfs:subClassOf xxx . vvv rdf:type uuu .</td>
<td>vvv rdf:type xxx .</td>
</tr>
<tr>
<td>rdfs10</td>
<td>uuu rdf:type rdfs:Class .</td>
<td>uuu rdfs:subClassOf uuu .</td>
</tr>
<tr>
<td>rdfs11</td>
<td>uuu rdfs:subClassOf vvv . vvv rdfs:subClassOf xxx .</td>
<td>uuu rdfs:subClassOf xxx .</td>
</tr>
<tr>
<td>rdfs12</td>
<td>uuu rdf:type rdfs:ContainerMembershipProperty .</td>
<td>uuu rdfs:subPropertyOf rdfs:member .</td>
</tr>
<tr>
<td>rdfs13</td>
<td>uuu rdf:type rdfs:Datatype</td>
<td>uuu rdfs:subClassOf rdfs:Literal .</td>
</tr>
</tbody>
</table>

Source: Corno & Farinetti: Logic and Reasoning in the Semantic Web
RDF-S reasoning

- RDF-S axiomatic triples (1/3)

<table>
<thead>
<tr>
<th>rdf:type</th>
<th>rdfs:domain</th>
<th>rdfs:Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:domain</td>
<td>rdfs:domain</td>
<td>rdfs:Property</td>
</tr>
<tr>
<td>rdfs:range</td>
<td>rdfs:domain</td>
<td>rdfs:Property</td>
</tr>
<tr>
<td>rdfs:subPropertyOf</td>
<td>rdfs:domain</td>
<td>rdfs:Property</td>
</tr>
<tr>
<td>rdfs:subClassOf</td>
<td>rdfs:domain</td>
<td>rdfs:Class</td>
</tr>
<tr>
<td>rdf:subject</td>
<td>rdfs:domain</td>
<td>rdf:Statement</td>
</tr>
<tr>
<td>rdf:predicate</td>
<td>rdfs:domain</td>
<td>rdf:Statement</td>
</tr>
<tr>
<td>rdf:object</td>
<td>rdfs:domain</td>
<td>rdf:Statement</td>
</tr>
<tr>
<td>rdfs:member</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdf:first</td>
<td>rdfs:domain</td>
<td>rdf:List</td>
</tr>
<tr>
<td>rdf:rest</td>
<td>rdfs:domain</td>
<td>rdf:List</td>
</tr>
<tr>
<td>rdfs:seeAlso</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:isDefinedBy</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdfs:label</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
<tr>
<td>rdf:value</td>
<td>rdfs:domain</td>
<td>rdfs:Resource</td>
</tr>
</tbody>
</table>

Source: Corno & Farinetti: Logic and Reasoning in the Semantic Web
RDF-S reasoning

• RDF-S axiomatic triples (2/3)

```
rdf:type  rdfs:range  rdfs:Class .
rdfs:domain  rdfs:range  rdfs:Class .
rdfs:range  rdfs:range  rdfs:Class .
rdfs:subPropertyOf  rdfs:range  rdf:Property .
rdfs:subClassOf  rdfs:range  rdfs:Class .
rdf:subject  rdfs:range  rdfs:Resource .
rdf:predicate  rdfs:range  rdfs:Resource .
rdf:object  rdfs:range  rdfs:Resource .
rdfs:member  rdfs:range  rdfs:Resource .
rdf:first  rdfs:range  rdfs:Resource .
rdf:rest  rdfs:range  rdf:List .
rdfs:seeAlso  rdfs:range  rdfs:Resource .
rdfs:isDefinedBy  rdfs:range  rdfs:Resource .
rdfs:comment  rdfs:range  rdfs:Literal .
rdfs:label  rdfs:range  rdfs:Literal .
rdf:value  rdfs:range  rdfs:Resource .
```

Source: Corno & Farinetti: Logic and Reasoning in the Semantic Web
RDF-S reasoning

• RDF-S axiomatic triples (3/3)

```
rdf:Alt rdfs:subClassOf rdfs:Container.
rdf:Bag rdfs:subClassOf rdfs:Container.
rdf:Seq rdfs:subClassOf rdfs:Container.
rdfs:ContainerMembershipProperty rdfs:subClassOf rdfs:Property.

rdfs:isDefinedBy rdfs:subPropertyOf rdfs:seeAlso.

rdf:XMLLiteral rdf:type rdfs:Datatype.
rdf:XMLLiteral rdfs:subClassOf rdfs:Literal.
rdfs:Datatype rdfs:subClassOf rdfs:Class.

rdf:_1 rdf:type rdfs:ContainerMembershipProperty.
rdf:_1 rdfs:domain rdfs:Resource.
rdf:_1 rdfs:range rdfs:Resource.
rdf:_2 rdf:type rdfs:ContainerMembershipProperty.
rdf:_2 rdfs:domain rdfs:Resource.
rdf:_2 rdfs:range rdfs:Resource.
...```

Source: Corno & Farinetti: Logic and Reasoning in the Semantic Web
RDF-S reasoning

• RDF-S entailment lemma
  – S rdfs-entails E if and only if there is a graph
    • that can be derived
      – from S plus the RDF and RDFS axiomatic triples
      – by the application of rule lg, rule gl and the RDF and RDFS entailment rules
    • that either
      – simply entails E or
      – contains a clash
OWL reasoning

- General principle
  - Deducing new information from OWL axioms
- (Most used) principle: applying rules to
  - Enforce OWL constructors
  - Respond to queries
- In OWL2 RL, rules are specified in the spec:
  
  https://www.w3.org/TR/owl2-profiles/
  #Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules
OWL reasoning

- Based on OWL language constructs
- Different levels of expressivity

However, computing all interesting logical conclusions of an OWL ontology can be a challenging problem, and reasoning is typically multi-exponential or even undecidable.

- OWL2 profiles were created to
  - Be *tractable* (polynomial for tasks they are designed for)
  - Be easier to understand for practitioners
  - Make it easier to develop reasoning algorithms
Outline

• Summary
• Web-based reasoning techniques
• Reasoning using SemWeb languages expressivity
• **General aspects of reasoning**
  – Inference problems
  – Complexity
• Reasoning on dynamic KBs
• Conclusion
Basic inference problems

• Consistency
• Subsumption
  – Equivalence
• Instance retrieval

⇒ All these problems are reducible to KB satisfiability

Source: http://www.computational-logic.org/content/events/iccl-ss-2005/lectures/horrocks/part4b-tableaux.pdf
OWL inference problems

• Ontology Consistency
• Ontology Entailment
• Ontology Equivalence
• Ontology Equisatisfiability
• Class Expression Satisfiability
• Class Expression Subsumption
• Instance Checking
• Boolean Conjunctive Query Answering

Source: http://www.w3.org/TR/owl2-direct-semantics/
Closure and Reduction

- The closure of a graph $G$ is the graph containing $G$ and all triples that can be inferred from $G$ by a given reasoning algorithm.
- The reduction of a graph $G$ is the minimal subset of triples needed to compute $G$ (and therefore, its closure).

→ What can they be used for?
Closure and Reduction

• Computing the closure of a graph
  – “pre-computes” query answers (view materialization)
    ➔ minimizes query processing time

• Computing the reduction of a graph
  – Produces a “light graph”
    ➔ minimizes storage space
Hardness of reasoning

It is very easy to compute some OWL entailments – the challenge is to compute all entailments of a certain kind. Markus Krötzsch

- **Soundness**
  - All computed inferences are really entailed
  - Necessary to be reliable
- **Completeness**
  - All computed inferences should actually be entailed
  - Not always necessary, if well documented
Trade-offs between expressivity and complexity vs. speed of data processing
A note about complexity

- **OWL 1 (OWL under RDF semantics)**
  - OWL DL was NP complete
  - OWL Full was undecidable

- **OWL 2 (OWL under direct semantics)**
  - In general, reasoning is N2ExpTime-complete
  - EL, RL and QL profiles have been designed to be *tractable* (i.e. polynomial) in their most important tasks
A note about complexity

• Practical considerations
  – A tractable reasoning process is polynomial
    • In worst case, but worst case is not always the case
  – In practice, polynomial reasoning is not necessarily achievable
    • What’s the use of a $n^{100}$-complex algorithm?
    • Even $n^4$ can be too complex on big data sets

➤ Restraining the set of used constructs to the necessary ones is a good practice
➤ Reducing the size of requests can also help...
Outline

• Summary
• Web-based reasoning techniques
• Reasoning using SemWeb languages expressivity
• General aspects of reasoning
• **Reasoning on dynamic KBs**
  – Problem
  – Example of solution
• Conclusion
Reasoning on Dynamic Knowledge

• Problem position
  – Semantic reasoning can be a hard problem
  – What if the ontology changes?

• What can happen
  – Depending on the application, the KB can vary
    • Evolution of the domain knowledge
      → Graph structure (classes, properties)
    • Evolution of the application state
      → Individuals
    • Evolution of the model behavior
      → Inference rules
Reasoning on Dynamic Knowledge

• Problem position
  – Example on rule-based reasoning
  – What if:

  - E1 is removed
  - E5 is removed
  - E5 and E6 are removed
  - E5 and E6 are removed, and I3 is inserted as an explicit fact
Reasoning on Dynamic Knowledge

- Solution 1: « *tabula rasa* »
  - Erase all inferences and recompute everything
  - ...

59
Reasoning on Dynamic Knowledge

• Solution 2: Incremental reasoning (Motik, Horrocks, Kim 2012)
  – Based on Gupta & Al. 1993, DRed
  – Only designed for conjunctive rules
  – Overdeletion
    • Evaluate the list of implicit facts inferred from explicit facts to suppress
  – Rederivation
    • Evaluate the list of implicit facts that could have been derived from other explicit facts
Reasoning on Dynamic Knowledge

- Incremental reasoning: example
  - E5 is removed
Reasoning on Dynamic Knowledge

- **Solution 3:** Keep track of previous reasoning results
  - Ontology modules (Grau & al. 2007)
    - Split the ontology in atomic parts that do not interfere during reasoning
    - Apply IR technique only on modified modules
  - Tag-based reasoning (Terdjimi & al. 2018)
    - Tag all explicit facts "valid" at insertion and "invalid" at deletion
    - Store the explicit causes of each implicit fact
    - Filter implicit facts at response time according to the validity of their causes
  - ...
Outline

• Summary
• Web-based reasoning techniques
• Reasoning using SemWeb languages expressivity
• General aspects of reasoning
• Reasoning on dynamic KBs
• Conclusion
Conclusion

• Reasoning results can depend on
  – Representation language expressivity
  – KB
    • Size
    • Class hierarchy
    • Number and complexity of (actually used) relations
    • Presence of instances
  – Reasoning engine type
Conclusion

• Problems due to reasoning on the Semantic Web
  – Scalability
  – Distribution
  – Heterogeneity
  – Data quality
  – Data velocity (streams)
Conclusion

• Reasoning algorithms are quite simple
  – Building a reasoner is easy
  – Optimizing it is (much) harder

• Complexity is always computed in worst case
  ➔ Make sure the worst case never happens
  ➔ Use different algorithms for different problems
Conclusion

• In any case, logical reasoning is a mean to retrieve information “hidden” in the KB

  – It can solve complex queries in a generic, elegant manner

  – But it cannot produce new “knowledge”

Special thanks to Mehdi Terdjimi!
References

• F. Corno, L. Farinetti - Politecnico di Torino: Logic and Reasoning in the Semantic Web (part I – RDF/RDFS)
  http://elite.polito.it/files/courses/01LHV/2012/4-RDFreasoning.pdf

• F. Corno, L. Farinetti - Politecnico di Torino: Logic and Reasoning in the Semantic Web (part II – OWL)
  http://elite.polito.it/files/courses/01LHV/2012/7-OWLreasoning.pdf

• M. Krötzsch: OWL 2 Profiles: An Introduction to Lightweight Ontology Languages

• RDF Semantics - W3C Recommendation 10 February 2004
  http://www.w3.org/TR/rdf-mt/

• RDF Schema 1.1 - W3C Recommendation 25 February 2014
  http://www.w3.org/TR/rdf-schema/

• OWL 2 Web Ontology Language Document Overview (Second Edition) - W3C Recommendation 11 December 2012
  http://www.w3.org/TR/owl2-overview/
  http://www.w3.org/TR/owl-direct-semantics
  http://www.w3.org/TR/owl-quick-reference