Practical Non-monotonic Reasoning

Guido Governatori

Data and Knowledge Engineering Research Division School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, QLD 4072, Australia
guido@itee.uq.edu.au
http://www.itee.uq.edu.au/~guido

20 December 2004
Institute of Computer Science
Academy of Sciences of the Czech Republic
Outline

1 Motivation
   - The Semantic Web
   - Logic for the Semantic Web

2 Basic Defeasible Logic
   - Basics of Defeasible Logic
   - Proofs in Defeasible Logic
   - Defeasible Logic at Work

3 Ontologies and Defeasible Logic
   - Description Logic
   - Defeasible Description Logic

Copyright ©2004 by Guido Governatori
Practical Non-monotonic Reasoning
The Semantic Web

- RDF + rdfschema
- XML + NS + xmlschema
- Unicode
- URI
- Self-desc. doc.
- Data
- Logic
- Ontology vocabulary
- Proof
- Rules
- Trust
- Digital Signature

Copyright ©2004 by Guido Governatori
Data vs Information
Data vs Information

Information = Data + Processing
Data vs Information

Information = Data + Processing

- Huge amount of data (the whole Internet as a database), and very often irrelevant data
- Same (or similar) data from different sources
- Combine data from different sources
Ontologies

- What is an ontology?
- What are ontologies good for?
Ontologies

- What is an ontology?
  - Formal description of a phenomenon
- What are ontologies good for?
Ontologies

- What is an ontology?
  - Formal description of a phenomenon
- What are ontologies good for?
  - they allow us to understand the phenomenon they describe
Ontologies

- What is an ontology?
  - Formal description of a phenomenon

- What are ontologies good for?
  - they allow us to understand the phenomenon they describe
  - they allow us to reason about the phenomenon they describe
Ontologies: The Role of Reasoning

Class membership
- $x$ instance of $C$, $C$ subclass of $D$, therefore $x$ instance of $D$

Equivalence of classes
- $A$ equivalent to $B$, $B$ equivalent to $C$, therefore $A$ equivalent to $C$

Consistency
- Uncovers errors in the ontology and its instantiation

Classification
- $P$ a sufficient condition for $C$, $x$ satisfies $P$, therefore $x$ is an instance of $C$
Strength of Ontologies

- Motivation
  - Basic Defeasible Logic
  - Ontologies and Defeasible Logic
- The Semantic Web
  - Logic for the Semantic Web

**Strength of Ontologies**

- **Taxonomy**
  - Is sub-classification of
  - Thesaurus
    - Has narrower meaning than
  - Conceptual Model
    - Is subclass of
      - Local Domain Theory
        - Is disjoint subclass of
          - with transitive property
        - Description Logic
          - DAML+OIL, OWL
            - UML
              - Modal Logic
                - First Order Logic
                  - Weak semantics
                    - Strong semantics

- Relational Model
  - Schema
    - Extended ER
      - ER
        - Has narrower meaning than
          - Thesaurus
            - Is sub-classification of
              - Taxonomy

**Copyright ©2004 by Guido Governatori**

**Practical Non-monotonic Reasoning**
Requirements for Reasoning in the Semantic Web

- Well-defined syntax
- Well-defined and intuitively clear semantics
- Efficient reasoning support
- Sufficient expressive power
- Convenience of expression

All are important, but there is trade-off between:

- Efficient reasoning support
- Sufficient expressive power
Requirements for Reasoning in the Semantic Web

- Well-defined syntax
- Well-defined and intuitively clear semantics
- Efficient reasoning support
- Sufficient expressive power
- Convenience of expression

All are important, but there is trade-off between:

- Efficient reasoning support
- Sufficient expressive power

First-order logic? Logic programming? Description Logic?
Benefit of Reasoning: An Example

Knowledge

- herbivore $\iff$ animal eats (plant or (part_of plant))
- tree $\Rightarrow$ plant
- branch $\Rightarrow$ part_of tree
- leaf $\Rightarrow$ part_of branch
- giraffe $\Rightarrow$ animal eats leaf
- part_of = transitive

We can derive

- giraffe $\Rightarrow$ herbivore
but...

- Partial
- Incomplete
- Inconsistent
but...

- Partial
- Incomplete
- Inconsistent

Non-monotonic reasoning!
but...

- Partial
- Incomplete
- Inconsistent

Non-monotonic reasoning!
- Plethora of non-monotonic systems
- Lack of intuitive semantics
- High complexity
Defeasible Logic

- Directly Skeptical Semantics
Defeasible Logic

- Directly Skeptical Semantics
- Argumentation Semantics
Defeasible Logic

- Directly Skeptical Semantics
- Argumentation Semantics
- Positive and Negative Constructive Conclusions
Defeasible Logic

- Directly Skeptical Semantics
- Argumentation Semantics
- Positive and Negative Constructive Conclusions
- Flexible (e.g., Ambiguity Blocking vs Ambiguity Propagation)
Defeasible Logic

- Directly Skeptical Semantics
- Argumentation Semantics
- Positive and Negative Constructive Conclusions
- Flexible (e.g., Ambiguity Blocking vs Ambiguity Propagation)
- Computationally Efficient
Defeasible Logic

- Directly Skeptical Semantics
- Argumentation Semantics
- Positive and Negative Constructive Conclusions
- Flexible (e.g., Ambiguity Blocking vs Ambiguity Propagation)
- Computationally Efficient
- Many extensions and applications
  - policy based intention
  - BDI and BOID agents
  - automated negotiation
  - e-contracts analysis and monitoring
  - web service composition
Defeasible Logic

- Directly Skeptical Semantics
- Argumentation Semantics
- Positive and Negative Constructive Conclusions
- Flexible (e.g., Ambiguity Blocking vs Ambiguity Propagation)
- Computationally Efficient
- Many extensions and applications
  - policy based intention
  - BDI and BOID agents
  - automated negotiation
  - e-contracts analysis and monitoring
  - web service composition

For a free demonstration of Defeasible Logic call

1800 Def Log
Defeasible Logic

- Directly Skeptical Semantics
- Argumentation Semantics
- Positive and Negative Constructive Conclusions
- Flexible (e.g., Ambiguity Blocking vs Ambiguity Propagation)
- Computationally Efficient
- Many extensions and applications
  - policy based intention
  - BDI and BOID agents
  - automated negotiation
  - e-contracts analysis and monitoring
  - web service composition

For a free demonstration of Defeasible Logic

www.cit.gu.edu.au/~arock/ defeasible/Defeasible.cgi
Description Logics and Non-monotonic Reasoning

- add a layer of (non-monotonic) rules on top of description logic
- consider the intersection of description logic and the non-monotonic logic
A Defeasible Theory $D = (F, R, <)$ where

- $F$ is a set of Facts: $penguin(Tweety)$;
- $R$ is a set of rules
  - Strict Rules: $penguin(X) \rightarrow bird(X)$
  - Defeasible Rules: $bird(X) \Rightarrow flies(X)$
  - Defeater: $geneticallyModifiedPenguin(X) \leadsto flies(X)$
- $<$ is a superiority relation on $R$

$$
\begin{align*}
    r &: \quad bird(X) \Rightarrow flies(X) \\
    r' &: \quad penguin(X) \Rightarrow \neg flies(X)
\end{align*}
$$

Copyright ©2004 by Guido Governatori

Practical Non-monotonic Reasoning
A conclusion in $D$ is a tagged literal and can have one of the following four forms:

- $+\Delta q$, which is intended to mean that $q$ is definitely provable (i.e., using only facts and strict rules);
- $-\Delta q$, which is intended to mean that we have proved that $q$ is not definitely provable in $D$;
- $+\partial q$, which is intended to mean that $q$ is defeasibly provable in $D$;
- $-\partial q$ which is intended to mean that we have proved that $q$ is not defeasibly provable in $D$;
Monotonic Proofs

\(+\Delta:
\)

If $P(i + 1) = +\Delta q$ then

$\exists r \in R_s[q]$

$\forall a \in A(r) : +\Delta a \in P(1..i)$
Monotonic Proofs

\[+\Delta: \]
\[\text{If } P(i+1) = +\Delta q \text{ then} \]
\[\exists r \in R_s[q] \]
\[\forall a \in A(r) : +\Delta a \in P(1..i)\]

\[\neg\Delta: \]
\[\text{If } P(i+1) = -\Delta q \text{ then} \]
\[\forall r \in R_s[q] \]
\[\exists a \in A(r) : -\Delta a \in P(1..i)\]
A conclusion $p$ is derivable when:

- $p$ is a fact; or
- there is an applicable strict of defeasible rule for $p$, and either
- all the rules for $\neg p$ are discarded or
- every rule for $\neg p$ is weaker than an applicable strict or defeasible rule for $p$. 
Formal Definition.

\[ +\partial : \text{If } P(i + 1) = +\partial q \text{ then either} \]
\[ (1) +\Delta q \in P(1..i) \text{ or} \]
\[ (2) (2.1) \exists r \in R_{sd}[q] \forall a \in A(r) : +\partial a \in P(1..i) \text{ and} \]
\[ (2.2) -\Delta \sim q \in P(1..i) \text{ and} \]
\[ (2.3) \forall s \in R[\sim q] \text{ either} \]
\[ (2.3.1) \exists a \in A(s) : -\partial a \in P(1..i) \text{ or} \]
\[ (2.3.2) \exists t \in R_{sd}[q] \text{ such that} \]
\[ \forall a \in A(t) : +\partial a \in P(1..i) \text{ and } t > s. \]
Formal Definition. Sorry!

\[ +\partial: \text{ If } P(i+1) = +\partial q \text{ then either} \]
\[(1) +\Delta q \in P\langle1..i\rangle \text{ or} \]
\[(2) \quad (2.1) \exists r \in R_{sd}[q] \forall a \in A(r): +\partial a \in P\langle1..i\rangle \text{ and} \]
\[(2.2) \quad -\Delta \sim q \in P\langle1..i\rangle \text{ and} \]
\[(2.3) \forall s \in R[\sim q] \text{ either} \]
\[(2.3.1) \exists a \in A(s): -\partial a \in P\langle1..i\rangle \text{ or} \]
\[(2.3.2) \exists t \in R_{sd}[q] \text{ such that} \]
\[ \forall a \in A(t): +\partial a \in P\langle1..i\rangle \text{ and } t > s. \]
When two aircraft are on converging headings at approximately the same height, the aircraft that has the other on its right shall give way, except that (a) power-driven heavier-than-air aircraft shall give way to airships, gliders and balloons; ...
Case 1

\[ r_1 : \neg \text{rightOfWay}(Y, X) \Rightarrow \text{rightOfWay}(X, Y) \]
\[ r_2 : \text{onTheRightOf}(X, Y) \Rightarrow \text{rightOfWay}(X, Y) \]
\[ r_3 : \text{powerDriven}(X), \neg \text{powerDriven}(Y) \Rightarrow \neg \text{rightOfWay}(X, Y) \]
\[ r_4 : \text{balloon}(X) \rightarrow \neg \text{powerDriven}(X) \]
\[ r_5 : \text{glider}(X) \rightarrow \neg \text{powerDriven}(X) \]
\[ r_6 : \Rightarrow \text{powerDriven}(X) \]

\[ r_2 < r_3, r_6 < r_4, \text{ and } r_6 < r_5. \]
Case 2

\[ r_1 : \neg \text{rightOfWay}(Y, X) \Rightarrow \text{rightOfWay}(X, Y) \]
\[ r_2 : \text{onTheRightOf}(X, Y) \Rightarrow \text{rightOfWay}(X, Y) \]
\[ r_3 : \text{powerDriven}(X), \neg \text{powerDriven}(Y) \Rightarrow \neg \text{rightOfWay}(X, Y) \]
\[ r_4 : \text{balloon}(X) \rightarrow \neg \text{powerDriven}(X) \]
\[ r_5 : \text{glider}(X) \rightarrow \neg \text{powerDriven}(X) \]
\[ r_6 : \Rightarrow \text{powerDriven}(X) \]

\[ r_2 < r_3, \ r_6 < r_4, \text{ and } r_6 < r_5. \]
Theorem

The complexity of (propositional) Defeasible Logic wrt to a defeasible theory $D$ is $O(n)$, where $n$ is the number of symbols in $D$. 
### Basics of Description Logic ($\mathcal{ALC}^-$)

- **Concepts** (unary predicates)
- **Roles** (binary predicates)

\[
\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})
\]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$A^\mathcal{I} \subseteq \Delta^\mathcal{I}$</td>
</tr>
<tr>
<td>$R$</td>
<td>$R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$</td>
</tr>
<tr>
<td>$\neg A$</td>
<td>$\Delta^\mathcal{I}/A^\mathcal{I}$</td>
</tr>
<tr>
<td>$C \sqcap D$</td>
<td>$C^\mathcal{I} \cap D^\mathcal{I}$</td>
</tr>
<tr>
<td>$\forall R.C$</td>
<td>$\forall R.C^\mathcal{I} = {a \in \Delta</td>
</tr>
</tbody>
</table>
A Knowledge Base (KB) in Description Logic consists of

**TBox:** Concepts definitions
- equivalence axioms $C \equiv D$ ($C^I = D^I$)
  \[\text{Course} \equiv \text{IT} \cap \text{EE} \]  
- inclusion axioms $C \subseteq D$ ($C^I \subseteq D^I$)
  \[\text{Lecturer} \subseteq \exists \text{teaches.} \text{Course} \]
- for each term/concept there is at most one definition

**ABox:** individual assertions

\[
\begin{align*}
\text{Lecturer}(\text{GUIDO}) \\
\text{takes}(S123, \text{INFS4201}) \\
\forall \text{teaches.} \text{IT} \text{course}(\text{BOB}) \\
\text{Course} (\text{COMP6801})
\end{align*}
\]
Motivation
Basic Defeasible Logic
Ontologies and Defeasible Logic

DL + DL = DDL
Embedding DL in DL

Description Logic Theory

\[(A, T)\]

Defeasible Logic Theory

\[(F, R, <)\]
DL + DL = DDL
Embedding DL in DL

Description Logic Theory  
Defeasible Logic Theory

\((A, T) \hookrightarrow (A \cup F, \Delta_T, T \cup R, <) \hookrightarrow (F, R, <)\)
Motivation

Basic Defeasible Logic

Ontologies and Defeasible Logic

Description Logic

Defeasible Description Logic

\[ \text{DL} + \text{DL} = \text{DDL} \]

Embedding DL in DL

\[ (\mathcal{A}, \mathcal{T}) \hookrightarrow (\mathcal{A} \cup F, \Delta_T, \mathcal{T} \cup R, <) \hookrightarrow (F, R, <) \]

**ABox**  \( \mathcal{A} \): set of assertions

**TBox**  \( \mathcal{T} \): set of inclusion axioms (concepts definitions) \( \cap_{i=1}^{n} C_i \subseteq \cap_{j=1}^{m} D_j \)

which are transformed to strict rules

\[ C_1, \ldots, C_n \rightarrow D_1 \]

\[ \vdots \]

\[ C_1, \ldots, C_n \rightarrow D_m \]

and then if the axiom has the form \( \cap_{i=1}^{n} C_i \subseteq \forall R.D \) to

\[ C_1, \ldots, C_n, R(x, y) \rightarrow D(y) \]

\( \Delta_T \) is the Herbrand universe of the theory
Reasoning in DDL

\[ +\Delta \forall R.C: \]
\[
\text{If } P(i + 1) = +\Delta \forall R.C(a) \text{ then } \\
\forall b \in \Delta_T \text{ either } \\
(1) -\Delta R(a, b) \text{ or } \\
(2) +\Delta C(b)
\]

\[ +\partial \forall R.C: \]
\[
\text{If } P(i + 1) = +\partial \forall R.C(a) \text{ then } \\
\forall b \in \Delta_T \text{ either } \\
(1) -\partial R(a, b) \text{ or } \\
(2) +\partial C(b)
\]
The complexity of Defeasible Description Logic wrt a defeasible description theory $D$ is $O(n^4)$ where $n$ is the number of symbols in $D$. 
Example

TBox

\[ \text{TBox} \]

\[ \text{IteeStudent}(x) \sqsubseteq \text{Student}(x) \]

\[ \text{DualDegree}(x) \sqsubseteq \text{IteeStudent}(x) \]

Rules

\[ \forall \text{supervises}. \text{IteeStudent}(x) \Rightarrow \text{facultyMember}(x, \text{ITEE}) \]

\[ \text{Student}(x), \forall \text{takes}. \text{IteeCourse}(x) \Rightarrow \text{IteeStudent}(x) \]

\[ \text{Student}(x), \forall \text{takes}. \text{ArtsCourse}(x) \Rightarrow \neg \text{IteeStudent}(x) \]

ABox

\[ \text{Faculty(ITEE)} \]
\[ \text{Faculty(ARTS)} \]
\[ \text{Faculty(LAW)} \]
\[ \text{IteeCourse(INFS421)} \]
\[ \text{IteeCourse(COMP460)} \]
\[ \text{ArtsCourse(PSCY120)} \]
\[ \text{LawCourse(LAWS310)} \]
\[ \text{Student(DANI)} \]
\[ \text{Student(ROBIN)} \]
\[ \text{Supervisor(GUIDO)} \]
\[ \text{Supervisor(PENNY)} \]
\[ \text{takes(DANI, INFS421)} \]
\[ \text{takes(DANI, COMP460)} \]
\[ \text{takes(ROBIN, PSCY120)} \]
\[ \text{takes(ADRIAN, COMP460)} \]
\[ \text{takes(ROBIN, COMP460)} \]
\[ \text{takes(ANNE, LAWS310)} \]
\[ \text{supervises(GUIDO, DANI)} \]
\[ \text{supervises(GUIDO, ANNE)} \]
\[ \text{supervises(PENNY, ANNE)} \]
\[ \text{supervises(PENNY, ROBIN)} \]
### Example

**TBox**

\[
\begin{align*}
\text{IteeStudent}(x) & \sqsubseteq \text{Student}(x) \\
\text{DualDegree}(x) & \sqsubseteq \text{IteeStudent}(x)
\end{align*}
\]

**Rules**

\[
\begin{align*}
\forall \text{supervises.} \text{IteeStudent}(x) & \Rightarrow \text{facultyMember}(x, \text{ITEE}) \\
\text{Student}(x), \forall \text{takes.} \text{IteeCourse}(x) & \Rightarrow \text{IteeStudent}(x) \\
\text{Student}(x), \forall \text{takes.} \text{ArtsCourse}(x) & \Rightarrow \neg \text{IteeStudent}(x)
\end{align*}
\]

**ABox**

<table>
<thead>
<tr>
<th>Faculty(ITEE)</th>
<th>Faculty(ARTS)</th>
<th>Faculty(LAW)</th>
<th>IteeCourse(INFS421)</th>
<th>Student(DANI)</th>
<th>Supervisor(GUIDO)</th>
<th>Supervisor(PENNY)</th>
<th>LawCourse(LAWS310)</th>
<th>ArtsCourse(PSCY120)</th>
<th>ArtsCourse(PENNY, ANNE)</th>
<th>Faculty(ITEE)</th>
<th>Faculty(ARTS)</th>
<th>Faculty(LAW)</th>
<th>IteeCourse(INFS421)</th>
<th>Student(DANI)</th>
<th>Supervisor(GUIDO)</th>
<th>Supervisor(PENNY)</th>
<th>LawCourse(LAWS310)</th>
<th>ArtsCourse(PSCY120)</th>
<th>ArtsCourse(PENNY, ANNE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IteeCourse(COMP460)</td>
<td>ArtsCourse(PSCY120)</td>
<td>LawCourse(LAWS310)</td>
<td>IteeCourse(INFS421)</td>
<td>Student(DANI)</td>
<td>Supervisor(GUIDO)</td>
<td>Supervisor(PENNY)</td>
<td>LawCourse(LAWS310)</td>
<td>ArtsCourse(PSCY120)</td>
<td>ArtsCourse(PENNY, ANNE)</td>
<td>Faculty(ITEE)</td>
<td>Faculty(ARTS)</td>
<td>Faculty(LAW)</td>
<td>IteeCourse(INFS421)</td>
<td>Student(DANI)</td>
<td>Supervisor(GUIDO)</td>
<td>Supervisor(PENNY)</td>
<td>LawCourse(LAWS310)</td>
<td>ArtsCourse(PSCY120)</td>
<td>ArtsCourse(PENNY, ANNE)</td>
</tr>
<tr>
<td>DualDegree(ANNE)</td>
<td>Student(ROBIN)</td>
<td>takes(ROBIN, PSCY120)</td>
<td>takes(ROBIN, PSCY120)</td>
<td>takes(ROBIN, PSCY120)</td>
<td>takes(ROBIN, PSCY120)</td>
<td>takes(ROBIN, PSCY120)</td>
<td>takes(ROBIN, PSCY120)</td>
<td>takes(ROBIN, PSCY120)</td>
<td>takes(ROBIN, PSCY120)</td>
<td>supervises(GUIDO, DANI)</td>
<td>supervises(GUIDO, DANI)</td>
<td>supervises(GUIDO, DANI)</td>
<td>supervises(GUIDO, DANI)</td>
<td>supervises(GUIDO, DANI)</td>
<td>supervises(GUIDO, DANI)</td>
<td>supervises(GUIDO, DANI)</td>
<td>supervises(GUIDO, DANI)</td>
<td>supervises(GUIDO, DANI)</td>
<td></td>
</tr>
<tr>
<td>takes(DANI, INFS421)</td>
<td>takes(DANI, COMP460)</td>
<td>takes(ANNE, LAWS310)</td>
<td>takes(ANNE, LAWS310)</td>
<td>takes(ANNE, LAWS310)</td>
<td>takes(ANNE, LAWS310)</td>
<td>takes(ANNE, LAWS310)</td>
<td>takes(ANNE, LAWS310)</td>
<td>takes(ANNE, LAWS310)</td>
<td>takes(ANNE, LAWS310)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(GUIDO, ANNE)</td>
<td></td>
</tr>
<tr>
<td>takes(ROBIN, COMP460)</td>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td></td>
</tr>
<tr>
<td>supervises(GUIDO, ANNE)</td>
<td>supervises(PENNY, ANNE)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td>supervises(PENNY, ROBIN)</td>
<td></td>
</tr>
</tbody>
</table>

**New conclusions**

\[
\begin{align*}
\text{IteeStudent}(\text{DANI}) & \\
\text{facultyMember}(\text{GUIDO, ITEE})
\end{align*}
\]
Example

**TBox**

\[
\text{IteeStudent}(x) \sqsubseteq \text{Student}(x) \\
\text{DualDegree}(x) \sqsubseteq \text{IteeStudent}(x)
\]

**Rules**

\[
\forall \text{supervises}. \text{IteeStudent}(x) \Rightarrow \text{facultyMember}(x, \text{ITEE}) \\
\text{Student}(x), \forall \text{takes}. \text{IteeCourse}(x) \Rightarrow \text{IteeStudent}(x) \\
\text{Student}(x), \forall \text{takes}. \text{ArtsCourse}(x) \Rightarrow \neg \text{IteeStudent}(x)
\]

**ABox**

- Faculty(ITEE)
- Faculty(ARTS)
- Faculty(LAW)
- IteeCourse(INFS421)
- IteeCourse(COMP460)
- ArtsCourse(PSCY120)
- LawCourse(LAWS310)
- Student(DANI)
- Supervisor(GUIDO)
- Supervisor(PENNY)
- takes(DANI, INFS421)
- takes(DANI, COMP460)
- takes(ROBIN, PSCY120)
- takes(ADRIAN, COMP460)
- takes(ROBIN, COMP460)
- takes(ANNE, LAWS310)
- takes(PENNY, ANNE)
- supervises(GUIDO, DANI)
- supervises(PENNY, ANNE)
- supervises(GUIDO, ANNE)
- supervises(PENNY, ROBIN)

**New conclusions**

\[
\neg \delta \text{IteeStudent}(ROBIN) \\
\neg \delta \text{facultyMember}(PENNY, ITEE)
\]
Conclusions and Future Work

- first step towards the integration of DL and DL
- orthogonal to other similar approaches
- extending the expressive power of Defeasible Logic
  - including other DL constructors
  - nested rules
- optimising deductions (search space reduction)
- integrating ontologies and agents in Defeasible Logic
- implementation
Conclusions and Future Work

- first step towards the integration of DL and DL
- orthogonal to other similar approaches
- extending the expressive power of Defeasible Logic
  - including other DL constructors
  - nested rules
- optimising deductions (search space reduction)
- integrating ontologies and agents in Defeasible Logic
- implementation but don’t hold your breath
Acknowledgements

This work was partially supported by the “Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realisation” project of the Program of the Information Society of the Thematic Program II of the National Research Program of the Czech Republic (Project Number 1ET100300419)
Grigoris Antoniou, David Billington, Guido Governatori, and Michael J. Maher.
Representation results for defeasible logic.


Guido Governatori.
Defeasible description logic.

Guido Governatori, Michael J. Maher, David Billington, and Grigoris Antoniou.
Argumentation semantics for defeasible logics.

Guido Governatori and Antonino Rotolo.
Defeasible logic: Agency, intention and obligation.

Guido Governatori, Antonino Rotolo, and Shazia Sadiq.
A model of dynamic resource allocation in workflow systems.