Practical Non-monotonic Reasoning

Guido Governatori

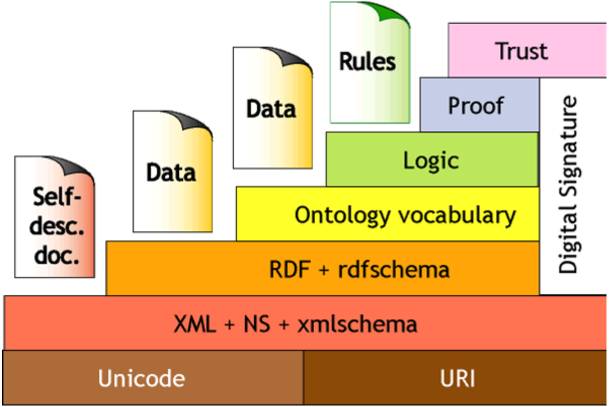
Data and Knowledge Engineering Reasearch Division School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, QLD 4072, Australia
guido@itee.uq.edu.au
http://www.itee.uq.edu.au/~guido

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

The Semantic Web



Semantic Web Issues

Data vs Information

Semantic Web Issues

Data vs Information

$$\text{Information} = \text{Data} + \text{Processing}$$

Semantic Web Issues

Data vs Information

$$\text{Information} = \text{Data} + \text{Processing}$$

- 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

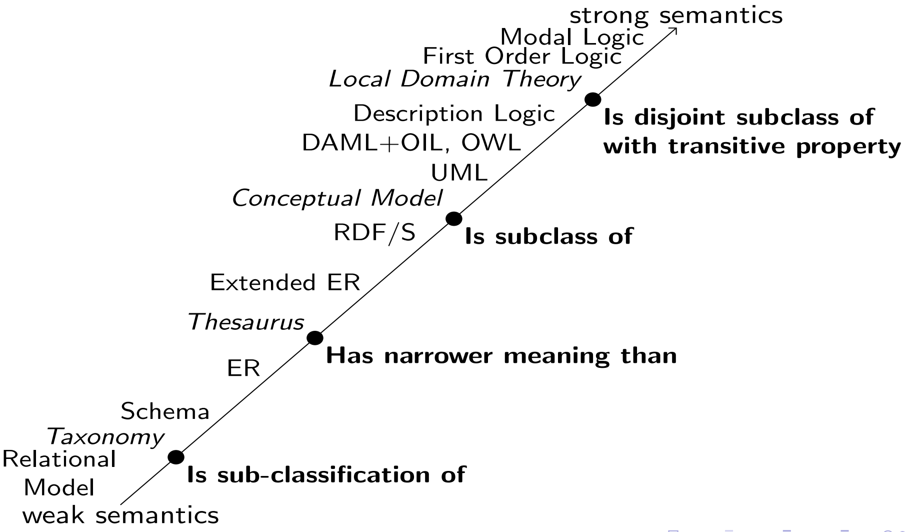
Consistency

- 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



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 ⇔ animal eats (plant or (part_of plant))
- tree ⇒ plant
- branch ⇒ part_of tree
- leaf ⇒ part_of branch
- giraffe ⇒ animal eats leaf
- part_of = transitive

We can derive

- giraffe ⇒ 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

Basics of Defeasible 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: $geneticallyModifiedPenguin(X) \rightsquigarrow flies(X)$
- $<$ is a superiority relation on R

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

Conclusions in Defeasible Logic

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

$\quad\exists r \in R_s[q]$

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

Monotonic Proofs

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

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

Non-monotonic derivations

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.

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

Formal Definition. Sorry!

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

A Regulation in Defeasible Logic

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; ...

$$r_1 : \neg rightOfWay(Y, X) \Rightarrow rightOfWay(X, Y)$$
$$r_2 : onTheRightOf(X, Y) \Rightarrow rightOfWay(X, Y)$$
$$r_3 : powerDriven(X), \neg powerDriven(Y) \Rightarrow \neg rightOfWay(X, Y)$$
$$r_4 : balloon(X) \rightarrow \neg powerDriven(X)$$
$$r_5 : glider(X) \rightarrow \neg powerDriven(X)$$
$$r_6 : \quad \Rightarrow powerDriven(X)$$

$r_2 < r_3$, $r_6 < r_4$, and $r_6 < r_5$.

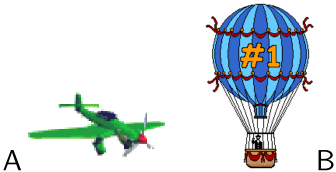
Case 1

A   B

$$
\begin{aligned}
r_1 &: \neg rightOfWay(Y,X) \Rightarrow rightOfWay(X,Y) \\
r_2 &: onTheRightOf(X,Y) \Rightarrow rightOfWay(X,Y) \\
r_3 &: powerDriven(X), \neg powerDriven(Y) \Rightarrow \neg rightOfWay(X,Y) \\
r_4 &: balloon(X) \rightarrow \neg powerDriven(X) \\
r_5 &: glider(X) \rightarrow \neg powerDriven(X) \\
r_6 &: \quad \Rightarrow powerDriven(X)
\end{aligned}
$$

$r_2 < r_3$, $r_6 < r_4$, and $r_6 < r_5$.

Case 2



$r_1 : \neg rightOfWay(Y, X) \Rightarrow rightOfWay(X, Y)$

$r_2 : onTheRightOf(X, Y) \Rightarrow rightOfWay(X, Y)$

$r_3 : powerDriven(X), \neg powerDriven(Y) \Rightarrow \neg rightOfWay(X, Y)$

$r_4 : balloon(X) \rightarrow \neg powerDriven(X)$

$r_5 : glider(X) \rightarrow \neg powerDriven(X)$

$r_6 : \quad \Rightarrow powerDriven(X)$

$r_2 < r_3$, $r_6 < r_4$, and $r_6 < r_5$.

Complexity of Defeasible Logic

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)

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

$$\begin{array}{l|l}
A & A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \\
R & R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\
\neg A & \Delta^{\mathcal{I}} / A^{\mathcal{I}} \\
C \sqcap D & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
\forall R.C & \forall R.C^{\mathcal{I}} = \{a \in \Delta | \forall b.(a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}
\end{array}$$

Representing knowledge in Description Logics

A Knowledge Base (KB) in Description Logic consists of

TBox: Concepts definitions

- equivalence axioms $C \equiv D$ ($C^{\mathcal{I}} = D^{\mathcal{I}}$)

$$Course \equiv ITcourse \sqcap EEcourse$$

- inclusion axioms $C \sqsubseteq D$ ($C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$)

$$Lecturer \sqsubseteq \exists teaches.Course$$

- for each term/concept there is at most one definition

ABox: individual assertions

$$Lecturer(GUIDO)$$
$$takes(S123, INFS4201)$$
$$\forall teaches.ITcourse(BOB)$$
$$Course(COMP6801)$$

DL + DL = DDL

Embedding DL in DL

Description Logic Theory

$(\mathcal{A}, \mathcal{T})$

Defeasible Logic Theory

$(F, R, <)$

DL + DL = DDL

Embedding DL in DL

Description Logic Theory | Defeasible Logic Theory

$$(\mathcal{A}, \mathcal{T}) \hookrightarrow (\mathcal{A} \cup F, \Delta_{\mathcal{T}}, \mathcal{T} \cup R, <) \hookleftarrow (F, R, <)$$

DL + DL = DDL

Embedding DL in DL

Description Logic Theory Defeasible Logic Theory

$$(\mathcal{A}, \mathcal{T}) \hookrightarrow (\mathcal{A} \cup F, \Delta_T, \mathcal{T} \cup R, <) \hookleftarrow (F, R, <)$$

ABox $\mathcal{A}$: set of assertions

TBox $\mathcal{T}$: set of inclusion axioms (concepts definitions) $\sqcap_{i=1}^{n} C_i \sqsubseteq \sqcap_{j=1}^{m} D_j$ which are transformed to strict rules

$$C_1, \dots, C_n \rightarrow D_1$$

$$\vdots$$

$$C_1, \dots, C_n \rightarrow D_m$$

and then if the axiom has the form $\sqcap_{i=1}^{n} C_i \sqsubseteq \forall R.D$ to

$$C_1, \dots, C_n, R(x, y) \rightarrow D(y)$$

- Δ_T is the Herbrand universe of the theory

Reasoning in DDL

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

$+\partial\forall R.C$:
If $P(i+1) = +\partial\forall R.C(a)$ then
$\forall b \in \Delta_T$ either
(1) $-\partial R(a,b)$ or
(2) $+\partial C(b)$

Complexity of Defeasible Description Logic

Theorem

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

$$\text{IteeStudent}(x) \sqsubseteq \text{Student}(x)$$
$$\text{DualDegree}(x) \sqsubseteq \text{IteeStudent}(x)$$

Rules

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

ABox

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

Example

TBox

$$\mathrm{IteeStudent}(x) \sqsubseteq \mathrm{Student}(x)$$
$$\mathrm{DualDegree}(x) \sqsubseteq \mathrm{IteeStudent}(x)$$

Rules

$$\forall \mathrm{supervises.IteeStudent}(x) \Rightarrow \mathrm{facultyMember}(x, \mathrm{ITEE})$$
$$\mathrm{Student}(x), \forall \mathrm{takes.IteeCourse}(x) \Rightarrow \mathrm{IteeStudent}(x)$$
$$\mathrm{Student}(x), \forall \mathrm{takes.ArtsCourse}(x) \Rightarrow \neg \mathrm{IteeStudent}(x)$$

ABox

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

New conclusions

IteeStudent(DANI) facultyMember(GUIDO, ITEE)

Example

TBox

$$\text{IteeStudent}(x) \sqsubseteq \text{Student}(x)$$
$$\text{DualDegree}(x) \sqsubseteq \text{IteeStudent}(x)$$

Rules

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

ABox

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

New conclusions

$-\partial\text{IteeStudent}(\text{ROBIN})$ $\qquad$ $-\partial\text{facultyMember}(\text{PENNY}, \text{ITEE})$

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)

Bibliography

Grigoris Antoniou, David Billington, Guido Governatori, and Michael J. Maher.
Representation results for defeasible logic.
ACM Transactions on Computational Logic, 2(2):255–287, April 2001.

Guido Governatori.
Defeasible description logic.
In Grigoris Antoniou and Harold Boley, editors, *Rules and Rule Markup Languages for the Semantic Web: Third International Workshop, RuleML 2004*, number 3323 in LNCS, pages 98–112, Berlin, 8 November 2004. Springer-Verlag.

Guido Governatori, Michael J. Maher, David Billington, and Grigoris Antoniou.
Argumentation semantics for defeasible logics.
Journal of Logic and Computation, 14(5):675–702, October 2004.

Guido Governatori and Antonino Rotolo.
Defeasible logic: Agency, intention and obligation.
In Alessio Lomuscio and Donald Nute, editors, *Deontic Logic in Computer Science*, number 3065 in LNAI, pages 114–128, Berlin, 2004. Springer-Verlag.

Guido Governatori, Antonino Rotolo, and Shazia Sadiq.
A model of dynamic resource allocation in workflow systems.
In Klaus-Dieter Schewe and Hugh E. Williams, editors, *Database Technology 2004*, number 27 in Conference Research and Practice of Information Technology, pages 197–206. Australian Computer Science Association, ACS, 19-21 January 2004.