# Combining Apples and Oranges: A Flexible Representation for Defeasible Logics and Repair Semantics

Abdelraouf Hecham[1]

Supervised by: Madalina Croitoru[1], Pierre Bisquert[2]

[1] GraphIK INRIA, LIRMM, University of Montpellier
[2] GraphIK INRIA, French Institute of Research in Agronomy (INRA)

26 June 2018, Montpellier, France.

# Outline

# Research Context: Knowledge Representation and Reasoning

1. Query answering from different sources of information (Data Exchange).

**SOURCE #1**

TREATMENTS

| ID | Patient | Hospital | NPI |
|---|---|---|---|
| P1 | W.Smith | PPTH | 1112 |

PHYSICIANS

| NPI | Name | Spec |
|---|---|---|
| 1112 | R. Chase | UROL |

**SOURCE #2**

MEDPRESCRIPTIONS

| ID | Patient | NPI | Doctor | Spec |
|---|---|---|---|---|
| P1 | W.Smith | 1112 | R. Chase | CARD |

**TARGET**

DOCTORS

| NPI | Name | Hospital | Spec |
|---|---|---|---|
| 1112 | R. Chase | PPTH | UROL |

PRESCRIPTIONS

| ID | Patient | NPI |
|---|---|---|
| P1 | W.Smith | 1112 |

Figure: DOCTORS Data Exchange Ontology (Geerts et al. 2004)

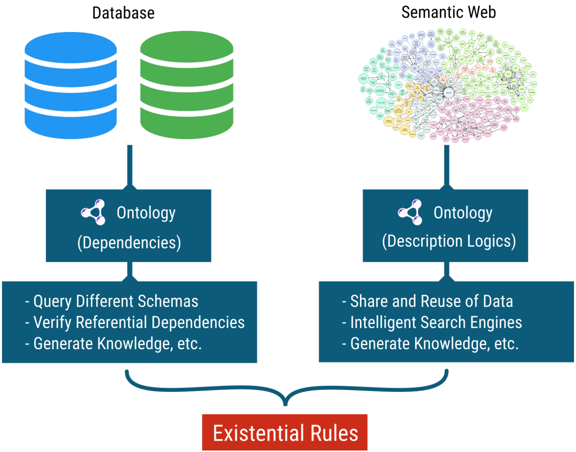
2. Bringing together different point of views for Decision Making.


Figure: DUR-DUR Knowledge Base

Figure: EcoBioCap Knowledge Base

**An expressive logical language is needed: Existential Rules**

# What Can Existential Rules Do?




- Existential Rules account for unknown individuals (value invention). e.g. Any prescription had to be made by a doctor X (might be unknown but must exist).
- They generalize certain fragments Description Logic (n-arity predicates, etc.).

# Existential Rules Language Datalog$^{\pm}$

- First order language composed of formulas built with $(\exists, \forall, \wedge, \rightarrow)$.
- Atom: of the form $p(t_1 ... t_k)$ where $p$ is a predicate and $t_i$ are variables $(X, Y, \ldots)$, fresh variables (a.k.a nulls, unknown constants. $Null_1, Null_1, \ldots$) or constants $(a, b, \ldots)$. Example: $human(raouf)$
- Fact: is an existentially closed atom. Example : $\exists X\ hasParent(raouf, X)$
- Rule: A rule $r$ is a formula of the form $\forall \vec{X}, \vec{Y} (\mathcal{B}[\vec{X}, \vec{Y}] \rightarrow \exists \vec{Z}\ \mathcal{H}[\vec{X}, \vec{Z}])$ where $\vec{X}$, $\vec{Y}$ are tuple of variables, $\vec{Z}$ is a tuple of existential variables and $\mathcal{B}, \mathcal{H}$ are finite non empty conjunctions of atoms respectively called Body and Head.
  $\forall PAT, NPI\ prescription(PAT, NPI) \rightarrow \exists NAME, SPEC\ doctor(NPI, NAME, SPEC)$

# Existential Rules Language Datalog$^{\pm}$

- First order language composed of formulas built with $(\exists, \forall, \wedge, \rightarrow)$.
- Atom: of the form $p(t_1 ... t_k)$ where $p$ is a predicate and $t_i$ are variables $(X, Y, \dots)$, fresh variables (a.k.a nulls, unknown constants. $Null_1, Null_1, \dots$) or constants $(a, b, \dots)$. Example: $human(raouf)$
- Fact: is an existentially closed atom. Example : $\exists X\ hasParent(raouf, X)$
- Rule: A rule $r$ is a formula of the form $\forall \vec{X}, \vec{Y} (\mathcal{B}[\vec{X}, \vec{Y}] \rightarrow \exists \vec{Z}\ \mathcal{H}[\vec{X}, \vec{Z}])$ where $\vec{X}$, $\vec{Y}$ are tuple of variables, $\vec{Z}$ is a tuple of existential variables and $\mathcal{B}, \mathcal{H}$ are finite non empty conjunctions of atoms respectively called Body and Head.
  $\forall PAT, NPI\ prescription(PAT, NPI) \rightarrow \exists NAME, SPEC\ doctor(NPI, NAME, SPEC)$
- Negative Constraint: (binary) of the form $\forall \vec{X}, \vec{Y} (p[\vec{X}, \vec{Y}] \wedge q[\vec{X}, \vec{Z}] \rightarrow \bot)$
  $\forall NAME, NPI\ doctor(NAME, NPI, card) \wedge doctor(NAME, NPI, urol) \rightarrow \bot$
- Knowledge Base: $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ where $\mathcal{F}$ is a set of facts, $\mathcal{R}$ is a set of rules, and $\mathcal{N}$ is a set of negative constraints.

# What Can Defeasible Reasoning Do?

The problem of Inconsistency and Incoherence

- Merging and integrating different Databases might produce inconsistent knowledge. (Inconsistency: conflicts within factual information).
- Ontologies might describe different point of view of the same domain, putting them together might create incoherence. (Incoherence: contradictions between inference rules)

Inconsistency and Incoherence are problematic for query answering. Classical entailment would yield the whole language in such case (principle of explosion).

# Incoherence

- A $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is incoherent iff $\mathcal{R} \cup \mathcal{N}$ are unsatisfiable.
- Unsatisfiable means that there does not exist any set of facts $S$ (even outside of the facts of the knowledge base) where all rules in $\mathcal{R}$ are applicable such that no negative constraint is applicable [Flouris et al., 2006].

Penguin Example: **(Incoherent)**

- $\mathcal{F} = \{penguin(kowalski)\}$
- $\mathcal{R} = \{$
  $r_1 : \forall X\ penguin(X) \rightarrow bird(X),$
  $r_2 : \forall X\ bird(X) \rightarrow fly(X),$
  $r_2 : \forall X\ penguin(X) \rightarrow notFly(X),$
  $\}$
- $\mathcal{N} = \{\forall X\ fly(X) \wedge notFly(X) \rightarrow \bot\}$

$\mathcal{R} \cup \mathcal{N}$ are unsatisfiable: there does not exists a set of facts such that all rules in $\mathcal{R}$ are applicable and the negative constraint is not applicable.

# Inconsistence

- A $\mathcal{KB} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is inconsistent iff a negative constraint is applicable on the knowledge derived from it.

Legal Example: **(Inconsistent but Coherent)**

- $\mathcal{F} = \{incrim(e1, alice), absolv(e2, alice), alibi(alice)\}$
- $\mathcal{R} = \{$
  $r_1 : \forall X, Y\ incrim(X, Y) \rightarrow resp(Y),$
  $r_2 : \forall X, Y\ absolv(X, Y) \rightarrow notResp(X),$
  $r_3 : \forall X\ resp(X) \rightarrow guilty(X),$
  $r_4 : \forall X\ alibi(X) \rightarrow innocent(X)\}$
- $\mathcal{N} = \{\forall X\ resp(X) \wedge notResp(X) \rightarrow \bot,$
  $\forall X\ guilty(X) \wedge innocent(X) \rightarrow \bot\}$

- $\mathcal{KB}$ is inconsistent because a negative constraint is applicable.
- $\mathcal{KB}$ is coherent because $\mathcal{R} \cup \mathcal{N}$ is satisfiable: there exists a set of facts (e.g. $S = \{incrim(e1, bob), absolv(e2, alice), alibi(alice)\}$) s.t. all rules are applicable and no negative constraint is applicable.

# Incoherence vs Inconsistence

- The problem of inconsistence has been resolved for existential rules using Repair Semantics. However, this techniques assume that the knowledge base is coherent.
- The problem of incoherence can be solved using Defeasible Logics

- Defeasible Logics [Pollock, 1987] originate from the need to reason with incomplete knowledge by "*filling the gaps in the available information by making some kind of plausible (or desirable) assumptions*".
  Applications: Legal reasoning, agent negotiations, etc.

- Repair Semantics [Lembo and Ruzzi, 2007] originate from the need to handle inconsistency that arises due to merging or revision of different data sources.
  Applications: Ontology Based Data Access, etc.

# Incoherence vs Inconsistence

- Defeasible Logics were made to handle incoherence.
- Repair Semantics were made to handle inconsistence.

- Inconsistence is a special case of incoherence: Incoherence will always lead to Inconsistence [Flouris et al., 2006].
- Defeasible Logics and Repair Semantics can both be applied to inconsistent but coherent knowledge bases.

- There is no universally agreed upon / appropriate way to reason with conflicts (inconsistence or incoherence). [Horty et al., 1987]. Defeasible Logics and Repair Semantics are based on different intuitions.

Objective: Compare and Combine Defeasible Logics and Repair Semantics Intuitions (in a single formalism)!

# Statement Graph

**Statement Graph** is a representation of the reasoning process happening inside a knowledge base. It is built using logical building blocks (called statements) that describe a situation (premises) and a rule that can be applied on that situation.

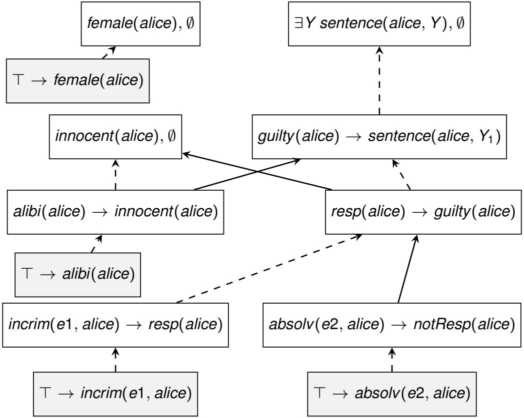



Figure: Statement Graph of Legal Example (support are dashed edges and fact statements are gray).

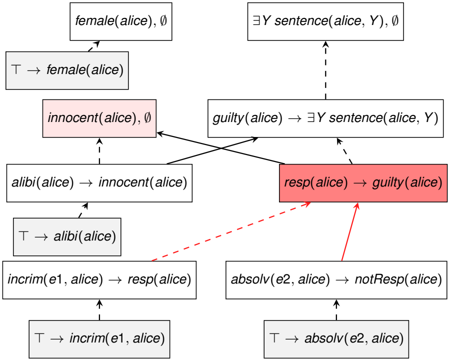
# Reasoning with Statement Graphs

- An SG provides statements and edges with a label using a labeling function.
- Labeling is used for Query answering.
- **Labeling Function** is $Lbl : \mathcal{V} \cup \mathcal{E}_A \cup \mathcal{E}_S \rightarrow Label = \{\mathrm{IN}, \mathrm{OUT}, \mathrm{AMBIG}\}$.
- The intuition behind these labels is: IN indicates that the statement is accepted and its rule can be applied, AMBIG indicates that the statement's premises are challenged by conflicting facts, and OUT indicates that the statement is rejected.
- **Complete Support** of a statement is a minimal set of support edges that support each one of its premises. IN complete support: for each premise there is a support edge labeled IN. AMBIG complete support: not IN complete support and for each premise there is a support edge labeled IN or AMBIG.

# Defeasible Reasoning Intuitions

- **Ambiguity Handling** whether an information that is derived from a contested (ambiguous) fact should be used to contest another fact.

  A fact $f$ is ambiguous if there is an accepted rule application for $f$ and another one for $f'$ such that $f$ and $f'$ are in conflict.



# Defeasible Reasoning Intuitions

- Ambiguity Blocking $\vDash_{block}$ facts based on ambiguous facts are blocked from challenging other facts.

  $innocent(alice)$ is in conflict with $guilty(alice)$ that relies on the ambiguous $resp(alice)$. Therefore $\mathcal{KB} \vDash_{block} innocent(alice)$

- We use the labeling function BDL (Blocking Defeasible Logic) to obtain entailment results equivalent to blocking defeasible logics [Billington, 1993].

1. *$BDL(\mathfrak{s}) = IN$ iff $\mathfrak{s}$ is a fact statement or $\mathfrak{s}$ has a IN complete support, and $\nexists \mathbf{e} \in \mathcal{E}^-_A(\mathfrak{s})$ s.t $BDL(\mathbf{e}) = IN$.*
2. *$BDL(\mathfrak{s}) = AMBIG$ iff either $\mathfrak{s}$ has an AMBIG complete support, or $\mathfrak{s}$ has a IN complete support and $\exists \mathbf{e} \in \mathcal{E}^-_A(\mathfrak{s})$ s.t $BDL(\mathbf{e}) = IN$.*
3. *$BDL(\mathfrak{s}) = OUT$ iff $\mathfrak{s}$ does not have a IN or AMBIG complete support.*

# Statement Graph: BDL

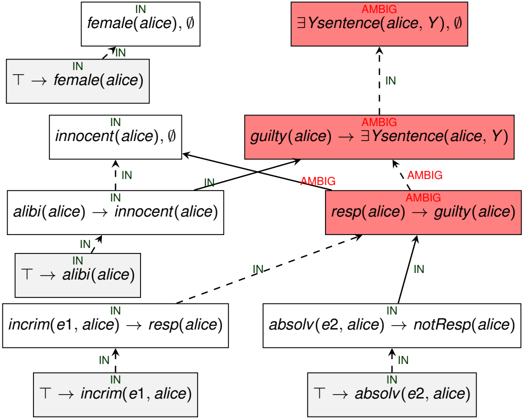



Figure: BDL applied to Legal Example's Statement Graph.

# Labeling for Ambiguity Propagating

- **Ambiguity Propagating $\models_{prop}$** The ambiguity of a fact is propagated to any fact it is in conflict with.

  $innocent(alice)$ is in conflict with $guilty(alice)$ that relies on the ambiguous $resp(alice)$. Therefore $\mathcal{KB} \models_{block} innocent(alice)$

- We use the labeling function **PDL (Propagating Defeasible Logic)** to obtain entailment results equivalent to propagating defeasible logics [Antoniou et al., 2000].

1. *$PDL(\mathbb{s}) = IN$ iff: $\mathbb{s}$ is a fact statement or $\mathbb{s}$ has a IN complete support, and $\nexists \boldsymbol{e} \in \mathcal{E}_A^-(\mathbb{s})$ s.t $PDL(\boldsymbol{e}) \in \{IN, AMBIG\}$.*
2. *$PDL(\mathbb{s}) = AMBIG$ iff:*
   1. *either $\mathbb{s}$ has an AMBIG complete support,*
   2. *or $\mathbb{s}$ has a IN complete support and $\exists \boldsymbol{e} \in \mathcal{E}_A^-(\mathbb{s})$ s.t $PDL(\boldsymbol{e}) \in \{IN, AMBIG\}$.*
3. *$PDL(\mathbb{s}) = OUT$ iff $\mathbb{s}$ does not have a IN or AMBIG complete support.*

# Statement Graph: PDL

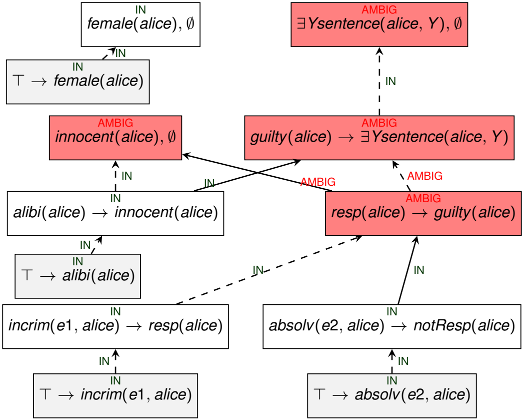



Figure: PDL applied to Legal Example's Statement Graph.

# Repair Semantics: IAR

- **IAR semantics $\models_{IAR}$** The intersection of ABox Repairs [Lembo et al., 2010]. A query Q is IAR entailed if it is classically entailed by the intersection of all repairs constructed from the starting set of facts.

- $Repair_1 = \{absolv(e2, alice), alibi(alice), female(alice)\}$
- $Repair_2 = \{incrim(e2, alice), female(alice)\}$
  $\mathcal{KB} \models_{IAR} female(alice)$

- Labeling function **IAR**: First apply PDL to detect conflicts, then start from the top (query statements) and reject any statement that leads to or is generated after a conflict.

1. $IAR(\mathfrak{s}) = IN$ *iff* $IAR(\mathfrak{s}) \neq AMBIG$ *and* $PDL(\mathfrak{s}) = IN$.
2. $IAR(\mathfrak{s}) = AMBIG$ *iff either* $PDL(\mathfrak{s}) = AMBIG$ *or* $\exists e \in \mathcal{E}^+_S(\mathfrak{s}) \cup \mathcal{E}^+_A(\mathfrak{s})$ *such that* $IAR(Target(e)) = AMBIG$.
3. $IAR(\mathfrak{s}) = OUT$ *iff* $PDL(\mathfrak{s}) = OUT$.

# Repair Semantics: IAR

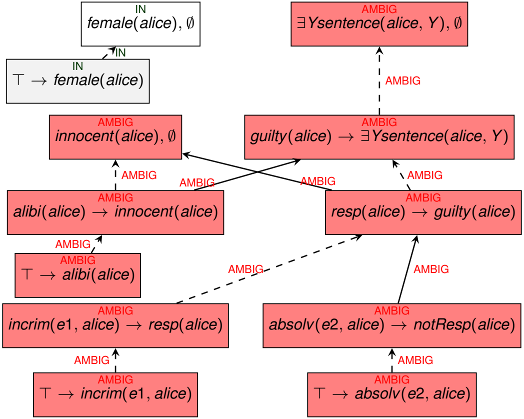



Figure: IAR applied to Legal Example's Statement Graph.

# Repair Semantics: ICAR

- **ICAR semantics $\models_{ICAR}$** The intersection of Closed ABox Repairs [Lembo et al., 2010]. A query Q is IAR entailed if it is classically entailed by the intersection of all repairs constructed from the saturated set of facts.

- $Repair_1 = \{absolv(e2, alice), alibi(alice), female(alice), notResp(alice), \exists Y\ sentence(alice, Y)\}$
- $Repair_2 = \{incrim(e2, alice), female(alice), resp(alice), guilty(alice), \exists Y\ sentence(alice, Y)\}$
  $\mathcal{KB} \models_{ICAR} female(alice) \land \exists Y\ sentence(alice, Y)$

- Labeling function **ICAR**: First apply PDL to detect conflicts, then start from the top (query statements) and reject any statement that leads to a conflict and accept those that are generated after a conflict.

1. $ICAR(\mathfrak{s}) = IN$ iff $ICAR(\mathfrak{s}) \neq AMBIG$ and $PDL(\mathfrak{s}) \in \{IN, AMBIG\}$.
2. $ICAR(\mathfrak{s}) = AMBIG$ iff
   1. either $PDL(\mathfrak{s}) = AMBIG$ and $\exists \boldsymbol{e} \in \mathcal{E}_A^-(\mathfrak{s})$ s.t. $PDL(\boldsymbol{e}) \in \{IN, AMBIG\}$,
   2. or $\exists e \in \mathcal{E}_S^+(\mathfrak{s}) \cup \mathcal{E}_A^+(\mathfrak{s})$ such that $ICAR(Target(e)) = AMBIG$.
3. $ICAR(\mathfrak{s}) = OUT$ iff $PDL(\mathfrak{s}) = OUT$.

# Repair Semantics: ICAR

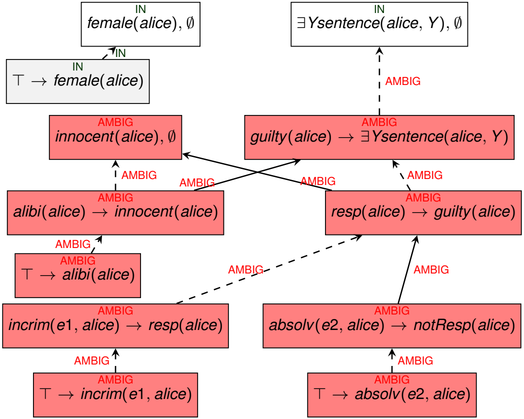



Figure: ICAR applied to Legal Example's Statement Graph.

# Human Reasoning Experiment

What intuitions humans follow in abstract inconsistent situations?

- Experiment with 41 participants, 5 situations.
  Context: engineer trying to analyze a situation based on a set of sensors.

- Example: Three sensors are respectively indicating that “o” has the properties S, Q, and T. We know that any object that has the property S also has the property V. Moreover, an object cannot have the properties S and Q at the same time, and the properties V and T at the same time. **Question:** Can we say that the object “o” has the property T?

  Logical representation (not shown to participants):
  - $\mathcal{F} = \{s(o), q(o), t(o)\}$
  - $\mathcal{R} = \{\forall X\, s(X) \rightarrow v(X)\}$
  - $\mathcal{N} = \{\forall X\, s(X) \land q(X) \rightarrow \bot, \forall X\, v(X) \land t(X) \rightarrow \bot\}$
  - Query $Q() = t(o)$

# Human Reasoning

Table: Situations Entailment and Results.

| Situations | $\models_{block}$ | $\models_{prop}$ | $\models_{IAR}$ | $\models_{ICAR}$ | % of "Yes" | $\models^{block}_{IAR}$ |
|---|---|---|---|---|---|---|
| #1 | ✓ | - | - | - | 73.17% | ✓ |
| #2 | ✓ | ✓ | - | - | 21.95% | - |
| #3 | ✓ | ✓ | - | ✓ | 21.95% | - |
| #4 | - | - | - | ✓ | 4.87% | - |
| #5 | ✓ | ✓ | ✓ | ✓ | 78.04% | ✓ |

- We observe that blocking and IAR are correspond the most to human reasoning.
- However, blocking or IAR alone are not sufficient.

- Possible explanation: participants are using a semantics that is a mix of IAR and ambiguity blocking.

# Combining Defeasible Logics and Repair Semantics

- We define $\vDash^{IAR}_{block}$ by using a labeling function that first applies BDL from fact statements to query statements then applies IAR from query statement to fact statements.

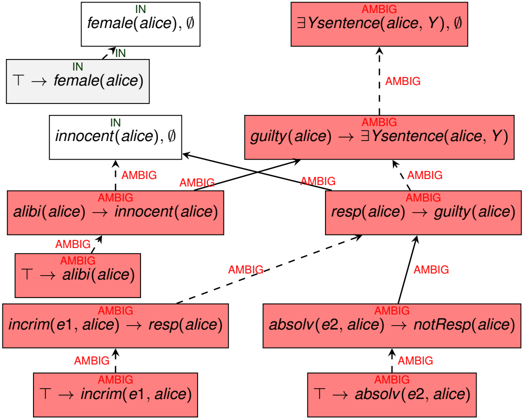



Figure: *BDL*/*IAR* applied to Legal Example's Statement Graph.

# Productivity Comparison

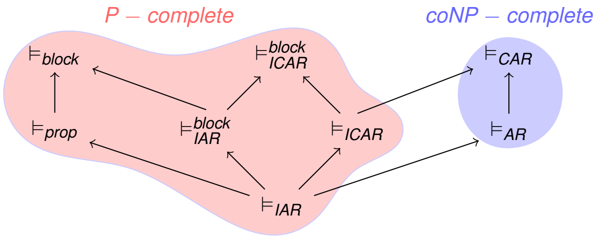



Figure: Productivity and complexity of different semantics under FES fragment of existential rules.

# Questions?

**Thank you for your attention.**


- Antoniou, G., Billington, D., Governatori, G., Maher, M. J., and Rock, A. (2000).
  A Family of Defeasible Reasoning Logics and its Implementation.
  In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 459–463.
- Billington, D. (1993).
  Defeasible Logic is Stable.
  *Journal of logic and computation*, 3(4):379–400.
- Flouris, G., Huang, Z., Pan, J. Z., Plexousakis, D., and Wache, H. (2006).
  Inconsistencies, negations and changes in ontologies.
  In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1295. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- Horty, J. F., Touretzky, D. S., and Thomason, R. H. (1987).
  A clash of intuitions: the current state of nonmonotonic multiple inheritance systems.
  In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 476–482.
- Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., and Savo, D. F. (2010).
  Inconsistency-tolerant semantics for description logics.