Reasoning with Inconsistent and Uncertain Ontologies

Guilin Qi
Southeast University
China
gqi@seu.edu.cn

Reasoning Web 2012  September 05, 2012
Outline

- Probabilistic logic vs possibilistic logic
- Probabilistic description logics
- Possibilistic description logics and its extension
- Revising ontologies in description logics
  - Belief revision
  - Revision of ontologies in DLs
- Mapping repair in description logics
Outline

- Probabilistic logic vs possibilistic logic
- Probabilistic description logics
- Possibilistic description logics and its extension
- Revising ontologies in description logics
  - Belief revision
  - Revision of ontologies in DLs
- Mapping repair in description logics
Example

- We encounter a strange new animal and it appears to be a bird
- As it comes closer, we see clearly it is red
  - Belief: the animal is a red bird
  - Formally: $\text{Bird}(a) \land \text{Red}(a)$

- We ask a bird expert who says the animal is not a bird but a sort of mammal
  - Conflict!

What do we believe now?
Example

Knowledge
- Old knowledge: $K = \{ \text{Bird}(a) \land \text{Red}(a) \}$
- New knowledge: $\phi = \neg \text{Bird}(a)$

Problem: $K$ and $\phi$ are in conflict
- $K \cup \{ \phi \}$ is inconsistent
Introduction of Belief Revision

- Earlier was proposed in database update
  - New tuples are added to a database
  - Cause the violation of integrity constraints

- Has been discussed from a philosophical view
  - Pioneer work by Carlos E. Alchourrón, Peter Gärdenfors, David Makinson (AGM)

- Has application in many areas
  - Databases
  - Artificial intelligence
  - Multi-agent systems
  - Planning
  - Semantics Web
Definition of a Revision Operator

- According to wikipedia
  
  “Belief revision is the process of changing beliefs to take into account a new piece of information.”

- A revision operator is a mapping from a theory and a formula to a theory
  
  - A theory is a set of deductively closed formulas (also called belief set)

- Questions
  
  - Is it reasonable to consider “theory”?
  - What is a rational revision operator?
  - How do we iterate the revision?
  - ...

...
Belief Base

- Arguments against belief set
  - No distinction is made between pieces of knowledge that are known by themselves and pieces of knowledge that are merely consequences of them
  - It fulfils the principle of irrelevance of syntax, which is debatable
    - \{p, q\} and \{p \land q\} should be treated differently when revised by \(\neg p\)

- Use of Belief base
  - A set of formulas that are not deductively closed
  - Revision operators applied to belief bases typically selects some subset of the original knowledge base that are consistent with the new knowledge
Principle of Belief Revision

- **Adequacy of representation:** The revised knowledge should have the same representation as the old knowledge.

- **Irrelevance of syntax:** The revised knowledge base should not depend on the syntactical form of either original knowledge base or the new formula.

- **Maintenance of consistency:** The revised knowledge base should be consistent.

- **Primacy of new information:** New information should always be accepted.

- **Minimal change:** As much information in original knowledge base should be kept after revision.
Example (Cont.)

- **Knowledge**
  - Old knowledge: $K = \{\text{Bird}(a) \land \text{Red}(a)\}$
  - New knowledge: $\phi = \neg \text{Bird}(a)$

- **Problem:** $K$ and $\phi$ are in conflict
  - $K \cup \{\phi\}$ is inconsistent

- $K \ast \phi = \{\neg \text{Bird}(a) \land \text{Red}(a)\}$
  - Minimal change
  - Primacy of new information
  - ...
AGM Postulates

(K₁) $K * \phi$ is a belief set (adequacy of representation)

(K₂) $\phi \in K * \phi$ (primacy of new information)

(K₃) $K * \phi \subseteq K + \phi$

(K₄) If $\neg \phi \notin K$ then $K + \phi \subseteq K * \phi$

(K₅) If $\phi$ is consistent then $K * \phi$ is also consistent (maintenance of consistency)

(K₆) If $Cn(\phi) = Cn(\psi)$ then $K * \phi = K * \psi$ (independency of syntax)

(K₇) $K * (\phi \land \psi) \subseteq (K * \phi) + \psi$

(K₈) If $\neg \psi \notin K * \phi$ then $(K * \phi) + \psi \subseteq K * (\phi \land \psi)$
Constructive Models for AGM Postulates

- Selection function
- Epistemic entrenchments
- System of spheres
Partial Meet Belief Revision

- Selection function $\gamma$: maps a non-empty collection $X$ of subsets of $K$ to a non-empty subset $\gamma(X)$ of $X$
- $\phi$-remainder of $K$: a maximal subsets of $K$ that fail to entail $\phi$
- $K \perp \phi$: set of all $\phi$-remainders of $K$
- Partial meet belief revision for $K$ and $\phi$
  - We first find all the $\neg \phi$-remainders of $K$ (subsets of $K$ that are consistent with $\phi$)
  - We apply the selection function to $K \perp \neg \phi$, get $\gamma(K \perp \neg \phi)$
  - Take conjunction of elements in $\gamma(K \perp \neg \phi)$ and $\phi$ as the result of revision

- Theorem: partial meet belief revision operators correspond to the postulates $(K_1)$ to $(K_8)$
Reformulation of AGM Postulates in Propositional Logic

(R_1) \phi*\mu \vdash \mu

(R_2) If \phi \land \mu is satisfiable then \phi*\mu \equiv \phi \land \mu

(R_3) If \mu is satisfiable then \phi*\mu is also satisfiable

(R_4) If \phi_1 \equiv \phi_2 and \mu_1 \equiv \mu_2 then \phi_1*\mu_1 \equiv \phi_2* \mu_2

(R_5) (\phi*\mu) \land \psi implies \phi*(\mu \land \psi)

(R_6) If (\phi*\mu) \land \psi is satisfiable then \phi*(\mu \land \psi) implies (\phi*\mu) \land \psi

☐ Theorem: Given a belief set \( K \), if \( \phi \) is a formula that satisfies \( K = \text{Cn}(\phi) \) and \( K*\mu = \text{Cn}(\phi \circ \mu) \), then \* satisfies \( (K_1) \rightarrow (K_8) \) iff \( \circ \) satisfies \( (R_1) \rightarrow (R_6) \)
Dalal’ s Revision Operator

- Distance function: Hamming distance between two interpretations

Example: atoms are p, q, r

\[ \omega: \ 1 \ 1 \ 0 \]
\[ \omega': \ 0 \ 1 \ 0 \]

\[ d(\omega, \omega') = 1 \]

- Idea: to revise formula \( \phi \) by formula \( \psi \)
  - Compute the distance \( d(\phi, \psi) \) between \( \phi \) and \( \psi \)
  - Take models of \( \psi \) whose distance with \( \phi \) is equal to \( d(\phi, \psi) \)

- Theorem: Dalal’ s operator satisfies \( (R_1)-(R_6) \)
Base Revision Operators

- Assumption: $K$ is not closed under logical consequence, i.e. $K \neq \text{Cn}(K)$

- Operators: related to foundationalism in philosophy
  - **WIDTIO (When in Doubt, Throw it Out)**
    - Idea: the maximal subsets of $K \cup \{\phi\}$ that are consistent and contain $\phi$ are combined by intersection
  - **Ginsberg–Fagin–Ullman–Vardi**
    - Idea: the maximal subsets of $K \cup \{\phi\}$ that are consistent and contain $\phi$ are combined by disjunction
  - **Nebel’s revision operators**
    - Similar to WIDTIO and Ginsberg–Fagin–Ullman–Vardi but priority among formulas are given
  - **Hansson’s revision operators**: defined by selection function
Example

- Tweety is a bird: \( \text{Bird}(\text{Tweety}) \)
- Any bird can fly: \( \forall x (\text{Bird}(x) \rightarrow \text{Fly}(x)) \)
  - We can infer that \( \text{Fly}(\text{Tweety}) \)
- Later on, we learn that \( \neg \text{Fly}(\text{Tweety}) \) (Inconsistency!)

Formally
- \( K = \{ \text{Bird}(\text{Tweety}), \forall x (\text{Bird}(x) \rightarrow \text{Fly}(x)) \} \)
- \( \phi =: \text{Fly}(\text{Tweety}) \)
Example (Cont.)

- $K \bot \phi = \{K_1, K_2\}$
  - $K_1 = \{\text{Bird}(\text{Tweety})\}$
  - $K_2 = \{\forall x (\text{Bird}(x) \rightarrow \text{Fly}(x))\}$

- Different selection functions result in different revision operators
  - $\gamma(K \bot \phi) = K_1$
    - $K \phi = \{\text{Bird}(\text{Tweety}), \neg \text{Fly}(\text{Tweety})\}$
  - $\gamma(K \bot \phi) = K_2$
    - $K \phi = \{\forall x (\text{Bird}(x) \rightarrow \text{Fly}(x)), \neg \text{Fly}(\text{Tweety})\}$
Outline

- Probabilistic logic vs possibilistic logic
- Probabilistic description logics
- Possibilistic description logics and its extension
- Revising ontologies in description logics
  - Belief revision
  - Revision of ontologies in DLs
- Mapping repair in description logics
Motivation of Revision in DLs

- Ontologies change due to the following reasons
  - New axioms are added during ontology learning
  - Axioms containing modelling errors are modified
  - Ontologies with different priorities are merged
  - ...

- Problems with ontology change
  - The old ontology and the newly added ontology are not consistent together

- Revision: dealing with logical contradictions during ontology change
Reformulation of AGM Postulates

(O+1) $X \subseteq K \ast X$

(O+2) If $K \cup X$ is consistent, then $K \ast X = K \cup X$

(O+3) If $X$ is consistent, then $K \ast X$ is also consistent.

(O+4) If $X \equiv Y$, then $K \ast X \equiv K \ast Y$

Plus the following postulate which is dened by a contraction operator:

(O+5) $(K \ast X) \cap K = K - \neg X$

- The negation of an axiom has two different definitions (consistency-negation and coherence-negation)
- Two kinds of logical contradictions
Reformulation of AGM Postulates

- Problems

- Their reformulation of AGM postulates deviate the original idea of AGM theory

- Disjunction is not used: the result of revision must be a single ontology

- There are two kinds of contradictions in DLs: inconsistency and incoherence
  - Revision operators defined by these postulates are applied to deal with inconsistency only
Incoherence

- Unsatisfiable concept $C: C = \emptyset$, for all $I \models T$

- Incoherence: there is an unsatisfiable concept in $T$

- Problem of incoherence
  - Main source of inconsistency
  - Trivial subsumption
Debugging Terminologies

- **MUPS for** $A$ **w.r.t.** $T$: a subset $T'$ of TBox $T$ such that
  - $A$ is unsatisfiable in $T'$
  - $A$ is satisfiable in any $T''$ where $T'' \subseteq T'$
- Example: $T = \{\text{Manager} \sqsubseteq \text{Employee}, \text{Employee} \sqsubseteq \text{JobPosition}, \text{JobPosition} \sqsubseteq \neg \text{Employee}, \text{Leader} \sqsubseteq \text{JobPosition}\}$
  - Manager is unsatisfiable
  - MUPS: $\{\text{Manager} \sqsubseteq \text{Employee}, \text{Employee} \sqsubseteq \text{JobPosition}, \text{JobPosition} \sqsubseteq \neg \text{Employee}\}$

- **MIPS for** $T$: a subset $T'$ of TBox $T$ such that
  - $T'$ is incoherent
  - any $T''$ with $T'' \subseteq T'$ is coherent
- Example (cont.): One MIPS
  - $\{\text{Employee} \sqsubseteq \text{JobPosition}, \text{JobPosition} \sqsubseteq \neg \text{Employee}\}$
A Kernel Revision Operator

- **Idea**: based on MIPS
  - Step 1: find MIPS of $T$ w.r.t. $T_0$
  - Step 2: remove some axioms in these MIPS

- **MIPS of $T$ w.r.t. $T_0$**: a subset $T'$ of TBox $T$
  - $T' \cup T_0$ is incoherent (incoherence)
  - Any $T''$ with $T'' \subseteq T'$ is coherent with $T_0$ (minimalism)

- **Example**
  - $T = \{\text{Manager} \sqsubseteq \text{Employee}, \text{Employee} \sqsubseteq \text{JobPosition}\}$
  - $T_0 = \{\text{JobPosition} \sqsubseteq \neg \text{Employee}, \text{Leader} \sqsubseteq \text{JobPosition}\}$
  - A MIPS of $T$ w.r.t. $T_0$
    - $\{\text{Employee} \sqsubseteq \text{JobPosition}\}$
A Kernel Revision Operator

Which axioms should be removed from MIPS?

- **Incision function $\sigma$** for $T$: for each TBox $T_0$ and the set $\text{MIPS}_{T_0}(T)$ of all MIPS of $T$ w.r.t. $T_0$
  
  - $\sigma(\text{MIPS}_{T_0}(T)) \subseteq \bigcup_{T_i \in \text{MIPS}_{T_0}(T)} T_i$ (Axioms selected belong to some MIPS)
  
  - $T' \cap \sigma(\text{MIPS}_{T_0}(T)) \neq \emptyset$, for any $T' \in \text{MIPS}_{T_0}(T)$ (Each MIPS has at least one axiom selected)

- **Naïve incision function**: $\sigma(\text{MIPS}_{T_0}(T)) = \bigcup_{T_i \in \text{MIPS}_{T_0}(T)} T_i$

- **Principle**: minimal change, i.e., select minimal number or set of axioms
A Kernel Revision Operator

- Kernel revision operator: Given $T$ and $\sigma$, for any $T_0$

\[ T *_{\sigma} T_0 = (T \setminus \sigma(MIPS_{T_0}(T))) \cup T_0 \]

- The result of revision is always a coherent TBox

- Logical properties

  - (R_1) $T_0 \subseteq T *_{\sigma} T_0$ (success)
  - (R_2) If $T \cup T_0$ is coherent, then $T *_{\sigma} T_0 = T \cup T_0$
  - (R_3) If $T_0$ is coherent then $T *_{\sigma} T_0$ is coherent (coherence preserve)
  - (R_4) If $T_1 \equiv T_2$, then $T *_{\sigma} T_1 \equiv T *_{\sigma} T_2$ (weak syntax independence)
  - (R_5) If $\phi \in T$ and $\phi \notin T *_{\sigma} T_0$, then there is a subset $S$ of $T$ and a subset $S_0$ of $T_0$ such that $SU S_0$ is coherent, but $SU S_0 \cup \{\phi\}$ is not (relevance)
Different incision functions will result in different specific kernel revision operators

- Incision functions can be computed by Reiter’s hitting set tree (HST) algorithm

However, there are potentially exponential number of hitting sets computed by the algorithm

- We reduce the search space by using scoring function or confidence values
Main steps: Given $T$ and $T_0$

- Step 1: compute MIPS of $T$ w.r.t. $T_0$
- Step 2: For each MIPS, we take its subset consisting of axioms whose priority is the lowest
- Step 3 Remove minimal number of axioms in these subsets from the ontology
Example

\[T = \{\text{Example} \sqsubseteq \text{Knowledge}, \text{Document} \sqsubseteq \neg \text{Knowledge}, \text{Form} \sqsubseteq \text{Knowledge}, \text{Firm} \sqsubseteq \text{Organization}\}\]

\[T_0 = \{\text{Document} \sqsubseteq \text{Example}, \text{Knowhow_document} \sqsubseteq \text{Document}, \text{Form} \sqsubseteq \text{Document}\}\]

- \[w_{\text{Example} \sqsubseteq \text{Knowledge}} = 0.4\]
- \[w_{\text{Document} \sqsubseteq \neg \text{Knowledge}} = 0.8\]
- \[w_{\text{Form} \sqsubseteq \text{Knowledge}} = 0.6\]
- \[w_{\text{Firm} \sqsubseteq \text{Organisation}} = 0.9\]
- The axioms in \(T_0\) are assigned weight 1
Example

- \( T = \{ \text{Example} \sqsubseteq \text{Knowledge}, \text{Document} \sqsubseteq \neg \text{Knowledge}, \text{Form} \sqsubseteq \text{Knowledge}, \text{Firm} \sqsubseteq \text{Organization} \} \)

- \( T_0 = \{ \text{Document} \sqsubseteq \text{Example}, \text{Knowhow_document} \sqsubseteq \text{Document, Form} \sqsubseteq \text{Document} \} \)

- MIPS of \( T \) w.r.t. \( T_0 \)
  
  - \( T_1 = \{ \text{Document} \sqsubseteq \neg \text{Knowledge} (0.8), \text{Form} \sqsubseteq \text{Knowledge} (0.6) \} \)
  
  - \( T_1 = \{ \text{Example} \sqsubseteq \text{Knowledge} (0.4), \text{Document} \sqsubseteq \neg \text{Knowledge} (0.8) \} \)

- Result of revision

\[ T \ast \sigma T_0 = T \cup T_0 \setminus \{ \text{Example} \sqsubseteq \text{Knowledge, Form} \sqsubseteq \text{Knowledge} \} \]
Outline

- Probabilistic logic vs possibilistic logic
- Probabilistic description logics
- Possibilistic description logics and its extension
- Revising ontologies in description logics
- Mapping repair in description logics
Ontology Mapping

O1

O2

Construct Mapping

Mapping between O1 and O2

Vehicle

Boat

Car

hasSpeed

Speed

Vehicle

Automobile

speed

Speed

Vehicle ↔ Vehicle

Car ↔ Automobile

Speed ↔ Speed

hasSpeed ↔ speed
Example

Source ontology crs: $O_1$

- document
- article
- program

Target Ontology ekaw: $O_2$

- Document
- Paper
- Workshop_Paper
- Conference_Paper

Mapping:

- document $\rightarrow$ Document: 0.93
- article $\rightarrow$ Paper: 0.80
- program $\rightarrow$ Workshop_Paper: 0.65
- program $\rightarrow$ Conference_Paper: 0.65
Example

Combined Ontology (O)

Mapping (M)

article

Conference_Paper

article

Workshop_Paper

program

Document

program

Document

document

Document
Formal Definition of Mapping Revision

- Distributed system D: \(<O_1,O_2,M>\)
- Union: \(O_1 \cup_M O_2 = O_1 \cup O_2 \cup \{t(m): m \in M\}\)
  
  \(t(<\text{crs:article}, \text{ekaw:Conference_paper}, \sqsubseteq, 0.65>) = \text{crs:article} \sqsubseteq \text{ekaw:Conference_paper}\)

- Inconsistency: M is inconsistent with \(O_1\) and \(O_2\) iff there is a concept which is satisfiable in \(O_i\), but unsatisfiable in \(O_1 \cup_M O_2\)

- Mapping revision operator: \(*<O_1,O_2,M> = <O_1,O_2,M'>\) with \(M' \subseteq M\)
Example

Source ontology crs: $O_1$

- document
  - article
    - program
  - disjoint

Target Ontology ekaw: $O_2$

- Document
  - Paper
    - Workshop_Paper
  - Conference_Paper

Mapping:
- document $\rightarrow$ Document: 0.93
- document $\rightarrow$ Paper: 0.80
- document $\rightarrow$ Workshop_Paper: 0.65
- document $\rightarrow$ Conference_Paper: 0.65
- article $\rightarrow$ Paper: 0.80
- article $\rightarrow$ Workshop_Paper: 0.65
- article $\rightarrow$ Conference_Paper: 0.65
- program $\rightarrow$ Paper: 0.80
- program $\rightarrow$ Workshop_Paper: 0.65
- program $\rightarrow$ Conference_Paper: 0.65

Isa relationships:
- document $\sqsubseteq$ Paper
- article $\sqsubseteq$ Paper
- program $\sqsubseteq$ Paper
- document $\sqsubseteq$ Workshop_Paper
- article $\sqsubseteq$ Workshop_Paper
- program $\sqsubseteq$ Workshop_Paper
- document $\sqsubseteq$ Conference_Paper
- article $\sqsubseteq$ Conference_Paper
- program $\sqsubseteq$ Conference_Paper
Conflict–based Mapping Revision

- Consider a distributed system \( D: \langle O_1, O_2, M \rangle \)
- Conflict set for \( A \) in \( O_i \): \( C \subseteq M \), \( A \) is satisfiable in \( O_i \) but unsatisfiable in \( O_1 \cup C \cup O_2 \)
  - Minimal conflict set: conflict set which is minimal w.r.t. set inclusion
  - \( \text{MCS}_{O_1, O_2}(M) \): all the minimal conflict sets for all the unsatisfiable concepts
- Incision function \( \sigma \) for \( D \)
  - \( \sigma(D) \subseteq \cup (\text{MCS}_{O_1, O_2}(M)) \)
  - If \( C \neq \emptyset \) and \( C \in \text{MCS}_{O_1, O_2}(M) \), then \( C \cap \sigma(D) \neq \emptyset \);
  - If \( m=\langle C, C', r, \alpha \rangle \in \sigma(D) \), then there exists \( C \in \text{MCS}_{O_1, O_2}(M) \) such that \( m \in C \), \( \alpha=\min\{\alpha_i: \langle C_i, C'_i, r_i, \alpha_i \rangle \in C\} \)
- Conflict–based Revision operator:
  - \( * \langle O_1, O_2, M \rangle = \langle O_1, O_2, \ M \setminus \sigma(\text{MCS}_{O_1, O_2}(M)) \rangle \)
Inconsistency degree of $D$:

$$\text{Inc}(D) = \max\{\alpha : \text{there is an unsatisfiable concept in } D_{\geq \alpha}\}$$

Postulates:

- (Relevance): a correspondence is removed only if it is (1) involved in a conflict, and (2) its confidence degree is minimal.
- (Consistency): consistency must be restored after revision.

Theorem: Operator $\ast$ is a conflict-based mapping revision operator iff it satisfies (Relevance) and (Consistency).
An iterative algorithm for Mapping Revision

**Input:** A distributed system $D=\langle O_1, O_2, M \rangle$ and a revision operator

**Output:** A repaired distributed system

**Algorithm:**

- **Step 1:** Stratify the mapping $M$
- **Step 2:** Compute inconsistency degree $d$
- **Step 3:** Use $O_1 \cup O_2 \cup M_{>d}$ to revise $M_{=d}$
- **Step 4:** If revised $D$ is still inconsistent, go to Step 2
Algorithm (Step 1)
----- Stratify the mapping

Stratify the mapping

M

Mapping (M)

document 0.93 Document
program 0.80 Document
program 0.80 Document
article 0.65 Workshop_Paper
article 0.65 Conference_Paper
article 0.65 Conference_Paper
program 0.80 Document
program 0.80 Document
document 0.93 Document
An iterative algorithm for Mapping Revision

**Input:** A distributed system $D=\langle O_1, O_2, M \rangle$ and a revision operator

**Output:** A repaired distributed system

**Algorithm:**

- Step 1: Stratify the mapping $M$
- Step 2: Compute inconsistency degree $d$
- Step 3: Use $O_1 \cup O_2 \cup M_{>d}$ to revise $M_{=d}$
- Step 4: If revised $D$ is still inconsistent, go to Step 2
Algorithm (Step 2)
-------- Compute inconsistency degree

M

document 0.93 Document

program 0.80 Document

program 0.80 Document

article 0.65 Workshop_Paper

article 0.65 Conference_Paper

$0_1 \cup 0_2 \cup M \cdot 0.93$ is consistent

$0_1 \cup 0_2 \cup M \cdot 0.80$ is inconsistent

Inconsistency degree is 0.80
An iterative algorithm for Mapping Revision

**Input:** A distributed system \( D = \langle O_1, O_2, M \rangle \) and a revision operator

**Output:** A repaired distributed system

**Algorithm:**

- Step 1: Stratify the mapping \( M \)
- Step 2: Compute inconsistency degree \( d \)
- Step 3: Use \( O_1 \cup O_2 \cup M_{>d} \) to revise \( M_{=d} \)
- Step 4: If revised \( D \) is still inconsistent, go to Step 2
Algorithm (Step 3)
----- Do revision

Revise $M_{=0.80}$ by $O_1 \cup O_2 \cup M_{>0.80}$

Compute a minimal conflict subset
- e.g. $\{\text{document } \subseteq \text{Document}, \text{Document } \subseteq \text{program}\}$

Remove an axiom with the lowest weight
- e.g. ax: $\text{Document } \subseteq \text{program}$ with weight 0.80

$(O_1 \cup O_2 \cup M_{\geq 0.80} \setminus \text{ax})$ becomes consistent
An iterative algorithm for Mapping Revision

**Input:** A distributed system $D=\langle O_1, O_2, M \rangle$ and a revision operator

**Output:** A repaired distributed system

**Algorithm:**

- Step 1: Stratify the mapping $M$
- Step 2: Compute inconsistency degree $d$
- Step 3: Use $O_1 \cup O_2 \cup M_{\geq d}$ to revise $M_{=d}$
- Step 4: If revised $D$ is still inconsistent, go to Step 2
Conclusions

- We give a short introduction of probabilistic logic and possibilistic logic and a comparison between them.
- We introduce probabilistic description logics and possibilistic description logics.
- We introduce belief revision in propositional logic and description logics.
Thank You!