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: $Bird(a) \wedge Red(a)$
- 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={Bird(a)∧Red(a)}
  - New knowledge: ϕ=¬Bird(a)
- Problem: K and ϕ are in conflict
  - K∪{ϕ} 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 $\{p \wedge q\}$ 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={Bird(a)∧Red(a)}
  - New knowledge: ϕ=¬Bird(a)
- Problem: K and ϕ are in conflict
  - $K \cup \{\phi\}$ is inconsistent
- $K * \phi$ ={¬Bird(a)∧Red(a)}
  - Minimal change
  - Primacy of new information
  - …

AGM Postulates

(K_1) $K * \phi$ is a belief set (adequacy of representation)

(K_2) $\phi \in K * \phi$ (primacy of new information)

(K_3) $K * \phi \subseteq K + \phi$

(K_4) If $\neg\phi \notin K$ then $K + \phi \subseteq K * \phi$

(K_5) If ϕ is consistent then $K * \phi$ is also consistent (maintenance of consistency)

(K_6) If $Cn(\phi) = Cn(\psi)$ then $K * \phi = K * \psi$ (independency of syntax)

(K_7) $K * (\phi \wedge \psi) \subseteq (K * \phi) + \psi$

(K_8) If $\neg\psi \notin K * \phi$ then $(K * \phi) + \psi \subseteq K * (\phi \wedge \psi)$

Constructive Models for AGM Postulates

- Selection function
- Epistemic entrenchments
- System of spheres

Partial Meet Belief Revision

- Selection function γ: maps a non-empty collection X of subsets of K to a non-empty subset $\gamma(X)$ of X
- ϕ–remainder of K: a maximal subsets of K that fail to entail ϕ
- $K \perp \phi$: set of all ϕ–remainders of K
- Partial meet belief revision for K and ϕ
  - We first find all the $\neg\phi$–remainders of K (subsets of K that are consistent with ϕ)
  - 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 ϕ 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 $\phi \wedge \mu$ is satisfiable then $\phi * \mu \equiv \phi \wedge \mu$

(R_3) If μ 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) $(\phi * \mu) \wedge \psi$ implies $\phi * (\mu \wedge \psi)$

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

- Theorem: Given a belief set *K*, if ϕ is a formula that satisfies $K = Cn(\phi)$ and $K * \mu = Cn(\phi \circ \mu)$, then $*$ satisfies (K_1) –(K_8) iff $\circ$ satisfies (R_1)–(R_6)

Dalal′ s Revision Operator

- Distance function: Hamming distance between two interpretations

  Example: atoms are p, q, r

$$\omega: \quad 1 \ 1 \ 0$$

$$\omega' : \quad 0 \ 1 \ 0$$

$$d(\omega,\omega')=1$$

1 denotes the atom is assigned T and 0 denote the atom is assigned F

- Idea: to revise formula ϕ by formula ψ
  - Compute the distance $d(\phi,\psi)$ between ϕ and ψ
  - Take models of ψ whose distance with ϕ 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. $K \neq Cn(K)$
- 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 ϕ are combined by intersection
  - Ginsberg-Fagin-Ullman-Vardi
    - Idea: the maximal subsets of $K \cup \{\phi\}$ that are consistent and contain ϕ 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: Bird(Tweety)
- Any bird can fly: $\forall x\,(\text{Bird}(x) \rightarrow \text{Fly}(x))$
  - We can infer that Fly(Tweety)
- Later on, we learn that $\neg$Fly(Tweety) (Inconsistency!)
- Formally
  - $K = \{\text{Bird(Tweety)}, \forall x\,(\text{Bird}(x) \rightarrow \text{Fly}(x))\}$
  - ϕ =:Fly(Tweety)

Example (Cont.)

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

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 contains 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) $X \subseteq K * X$

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

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

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

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

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

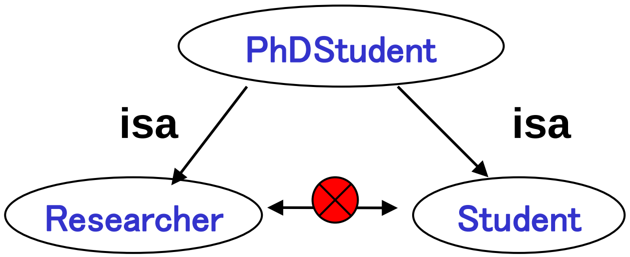
- The negation of an axiom has two different definitions (consistency-negation and coherence-negation)
- Two kinds of logical contradictions

Reformulation of AGM Postulates
–Problems

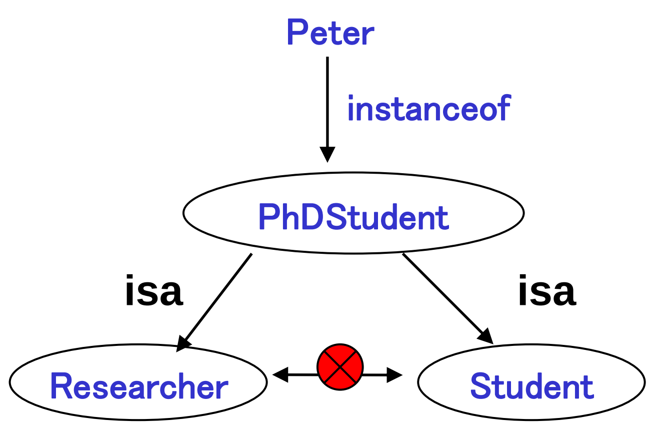
- Their reformulation of AGM postulates deviate the original idea of AGM theory
- Disjunction is not used: the result of revision must be a single ontology
- There are two kinds of contradictions in DLs: inconsistency and incoherence
  - Revision operators defined by these postulates are applied to deal with inconsistency only

Incoherence

- Unsatisfiable concept C: $C^I=\emptyset$, 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'' $\subset$ T'
  - Example: *T* = {Manager $\sqsubseteq$ Employee, Employee $\sqsubseteq$ JobPosition, JobPosition $\sqsubseteq$ $\neg$Employee, Leader $\sqsubseteq$ JobPosition}
    - Manager is unsatisfiable
    - MUPS: {Manager $\sqsubseteq$ Employee, Employee $\sqsubseteq$ JobPosition, JobPosition $\sqsubseteq$ $\neg$Employee}

Minimal sub-TBox of *T* in which *A* is unsatisfiable

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

Minimal sub-TBox of *T* which is incoherent

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 ={Manager ⊑ Employee, Employee ⊑ JobPosition}
  - T_0 ={JobPosition ⊑ ¬Employee, Leader ⊑ JobPosition}
  - A MIPS of T w.r.t. T_0
    - {Employee ⊑ JobPosition}

A Kernel Revision Operator

Which axioms should be removed from MIPS?

- Incision function σ for T: for each TBox T_0 and the set $MIPS_{T_0}(T)$ of all MIPS of T w.r.t. T_0
  - $\sigma(MIPS_{T_0}(T)) \subseteq \cup_{T_i \in MIPS_{T0}(T)} T_i$ (Axioms selected belong to some MIPS)
  - $T' \cap \sigma(MIPS_{T_0}(T)) \neq \emptyset$, for any $T' \in MIPS_{T_0}(T)$ (Each MIPS has at least one axiom selected)
- Naïve incision function: $\sigma(MIPS_{T_0}(T)) = \cup_{T_i \in MIPS_{T0}(T)} T_i$
- Principle: minimal change, i.e., select minimal number or set of axioms

A Kernel Revision Operator

- Kernel revision operator: Given T and σ, for any T_0

  $$T *_\sigma T_0 = (T \setminus \sigma(\mathrm{MIPS}_{T_0}(T))) \cup T_0$$

  – 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 $S \cup S_0$ is coherent, but $S \cup S_0 \cup \{\phi\}$ is not (relevance)

Algorithms

- 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*

Algorithms

- 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 = {Example ⊑ Knowledge, Document ⊑ ¬Knowledge, Form ⊑ Knowledge, Firm ⊑ Organization}

T_0 = {Document ⊑ Example, Knowhow_document ⊑ Document, Form ⊑ 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 = {Example ⊑ Knowledge, Document ⊑ ¬Knowledge, Form ⊑ Knowledge, Firm ⊑ Organization}

T_0 = {Document ⊑ Example, Knowhow_document ⊑ Document, Form ⊑ Document}

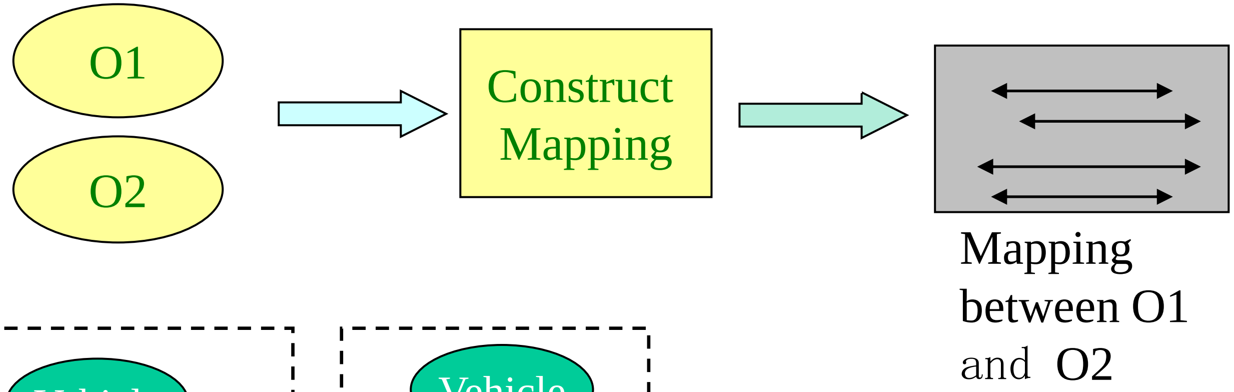
- MIPS of T w.r.t. T_0
  - T_1={Document ⊑ ¬Knowledge (0.8), Form ⊑ Knowledge (0.6) }
  - T_1={Example ⊑ Knowledge (0.4), Document ⊑ ¬Knowledge (0.8)}
- Result of revision

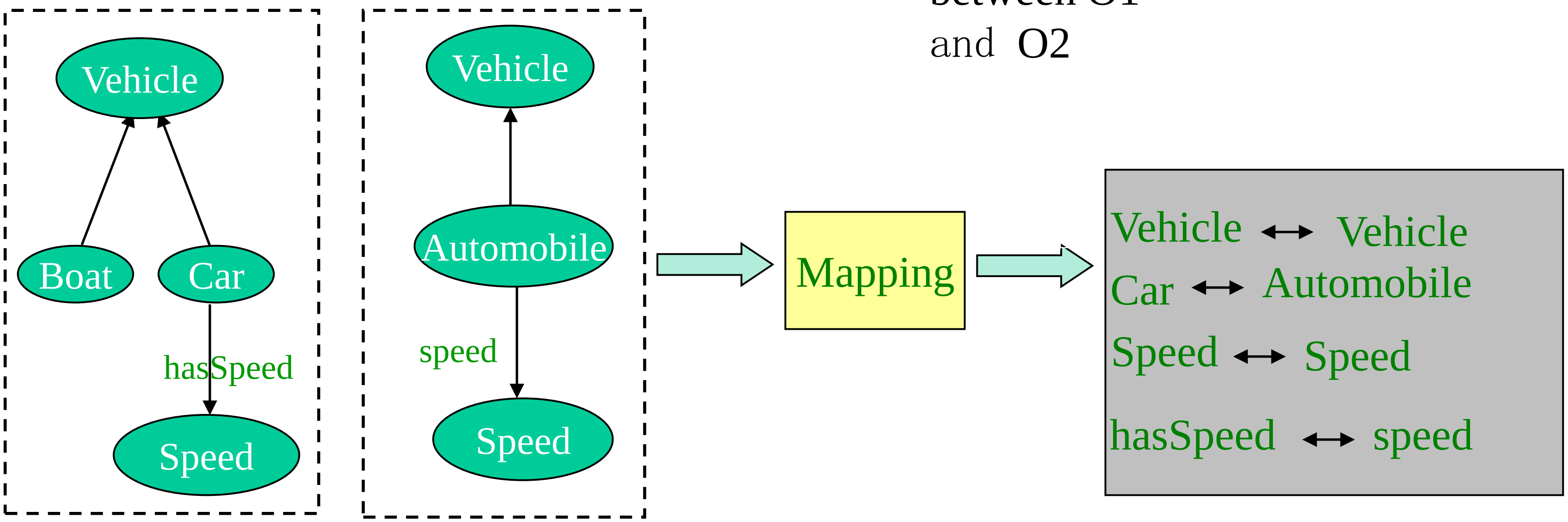
$T *_\sigma T_0 = T \cup T_0 \setminus$ {Example ⊑ Knowledge, Form ⊑ Knowledge}

Outline

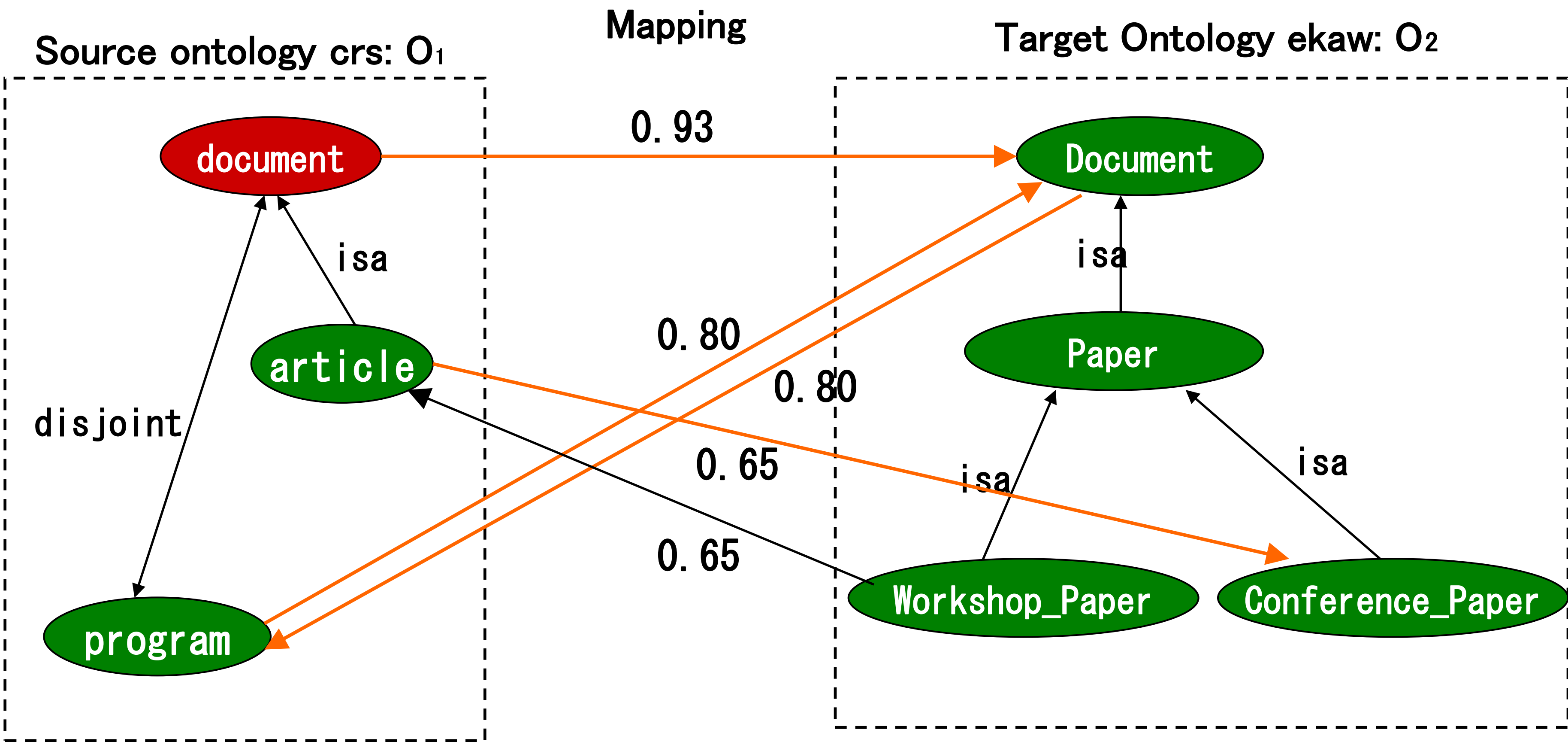
- Probabilistic logic vs possibilistic logic
- Probabilistic description logics
- Possibilistic description logics and its extension
- Revising ontologies in description logics
- **Mapping repair in description logics**

Ontology Mapping

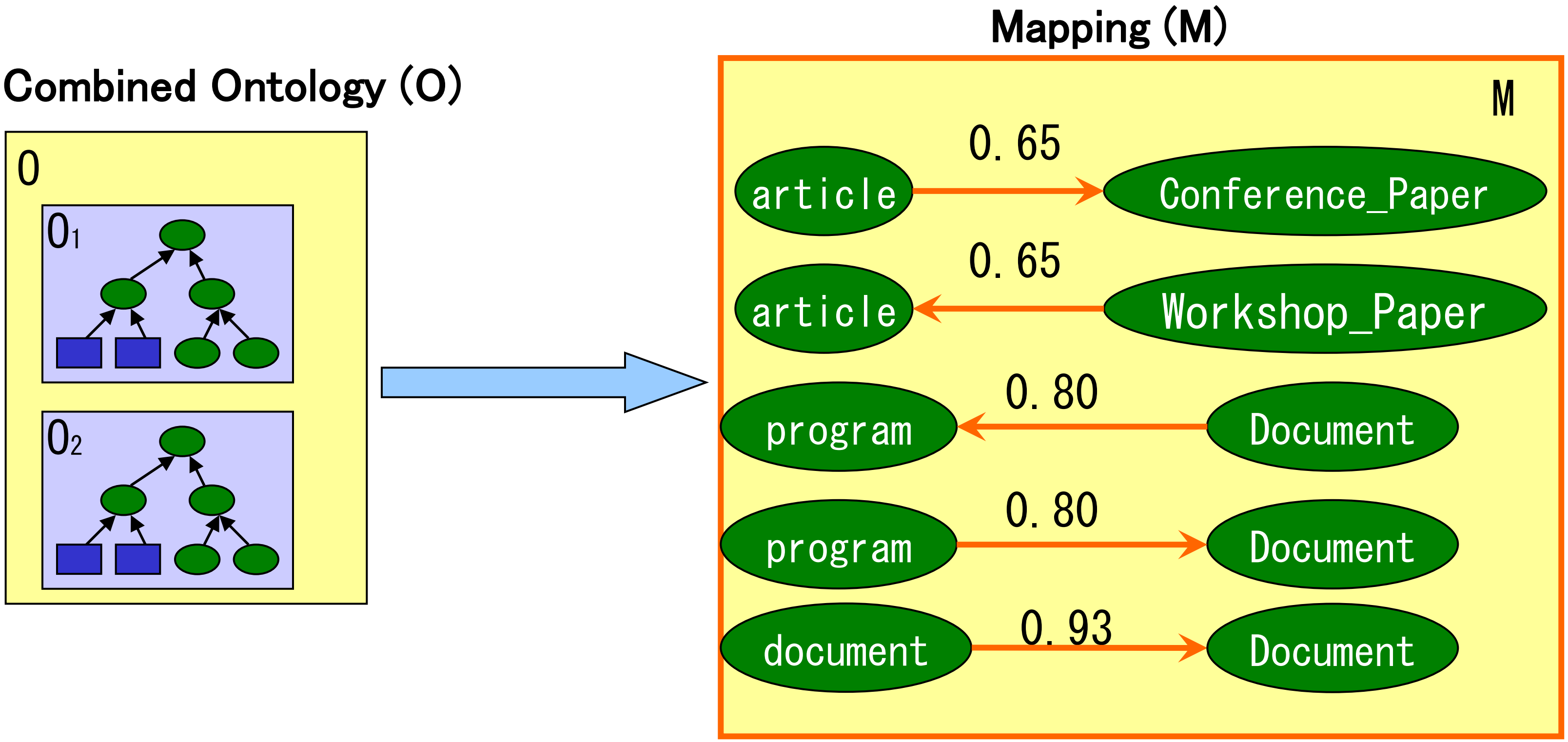




Example


Mapping
Source ontology crs: O1
Target Ontology ekaw: O2
document
article
program
isa
disjoint
0.93
0.80
0.80
0.65
0.65
Document
Paper
Workshop_Paper
Conference_Paper
isa
isa
isa

Example

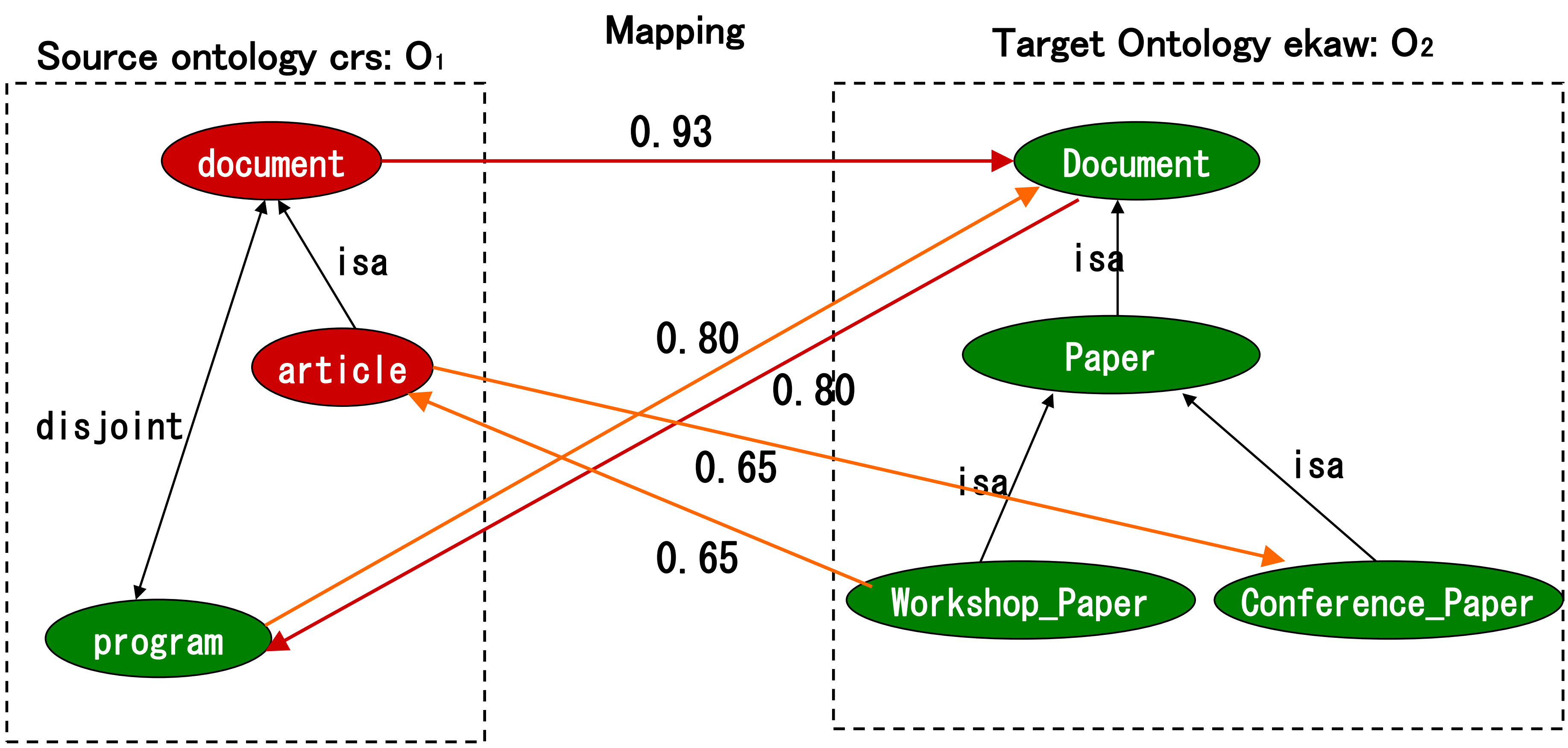


Combined Ontology (O)
O
O1
O2
Mapping (M)
M
article
0.65
Conference_Paper
article
0.65
Workshop_Paper
program
0.80
Document
program
0.80
Document
document
0.93
Document

Formal Definition of Mapping Revision

- Distributed system D: $\langle O_1, O_2, M \rangle$
- Union: $O_1 \cup_M O_2 = O_1 \cup O_2 \cup \{t(m): m \in M\}$
  - $t(\langle crs{:}article, ekaw{:}Conference\_paper, \sqsubseteq, 0.65 \rangle) = crs{:}article \sqsubseteq ekaw{:}Conference\_paper$
- 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: $*\langle O_1, O_2, M \rangle = \langle O_1, O_2, M' \rangle$ with $M' \subseteq M$

Example



Mapping
Source ontology crs: O1
Target Ontology ekaw: O2
document
article
program
isa
disjoint
0.93
0.80
0.80
0.65
0.65
Document
Paper
Workshop_Paper
Conference_Paper
isa
isa
isa

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 $O_1 \cup_C O_2$
  - Minimal conflict set: conflict set which is minimal w.r.t. set inclusion
  - $MCS_{O_1,O_2}(M)$: all the minimal conflict sets for all the unsatisfiable concepts
- Incision function σ for D
  - $\sigma(D) \subseteq \cup (MCS_{O_1,O_2}(M))$
  - If $C \neq \emptyset$ and $C \in MCS_{O_1,O_2}(M)$, then $C \cap \sigma(D) \neq \emptyset$;
  - If $m = \langle C,C',r,\alpha\rangle \in\sigma(D)$, then there exists $C\in MCS_{O_1,O_2}(M)$ such that $m \in C$, $\alpha=\min\{\alpha_i: \langle C_i,C'_i,r_i,\alpha_i\rangle \in C\}$

selects at least one element from each minimal conflict set

- Conflict-based Revision operator:
  - $* \langle O_1,O_2,M\rangle = \langle O_1,O_2, M\setminus \sigma(MCS_{O_1,O_2}(M))\rangle$

Representation Theorem

- α-cut of D: $D_{\geq\alpha}=(O_1,O_2,\{\langle C,C',r,\beta\rangle \in M, \beta\geq\alpha\})$
- Inconsistency degree of D
  - $Inc(D)=max\{\alpha$: there is an unsatisfiable concept in $D_{\geq\alpha}\}$

Maximum degree α such that α-cut of D is inconsistent

- 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 $*$ is a conflict-based mapping revision operator iff it satisfies (Relevance) and (Consistency)

An iterative algorithm for Mapping Revision

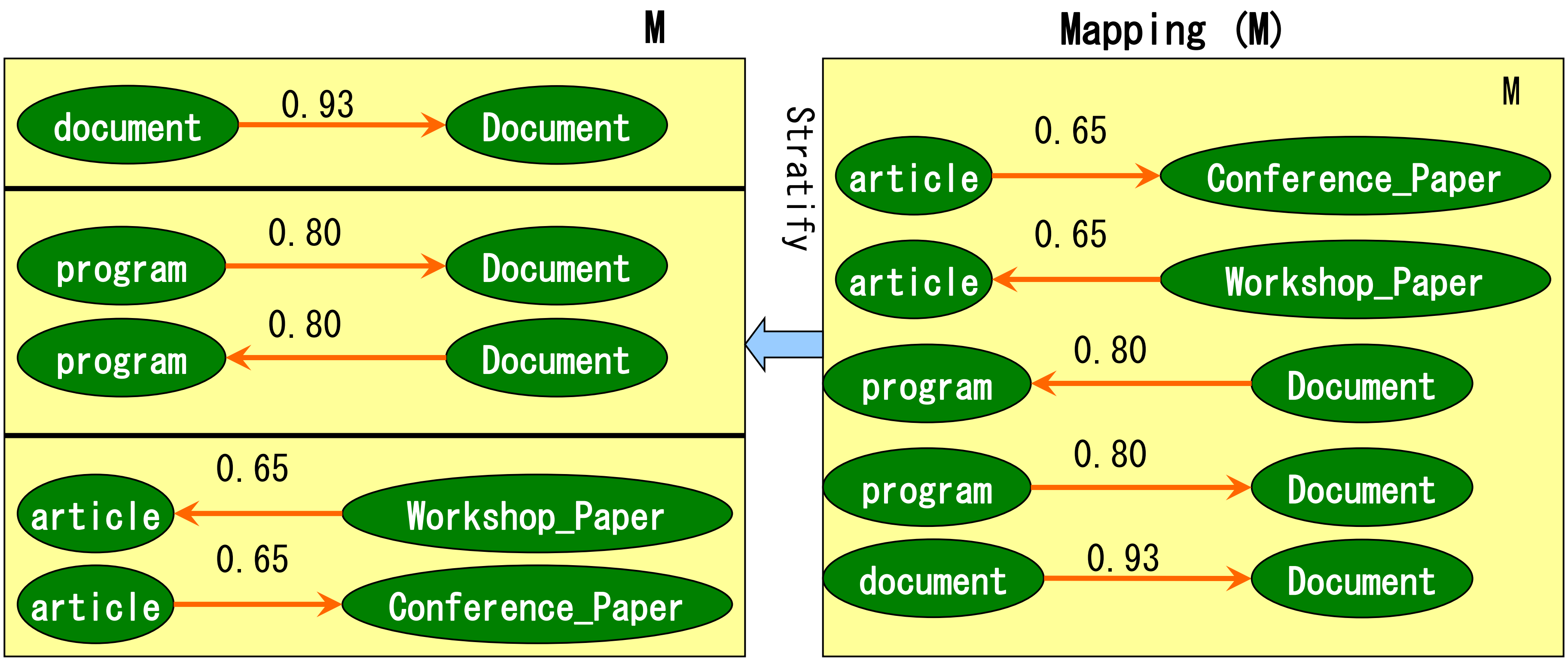
Input: A distributed system $D=\langle O_1,O_2,M\rangle$ and a revision operator

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



An iterative algorithm for Mapping Revision

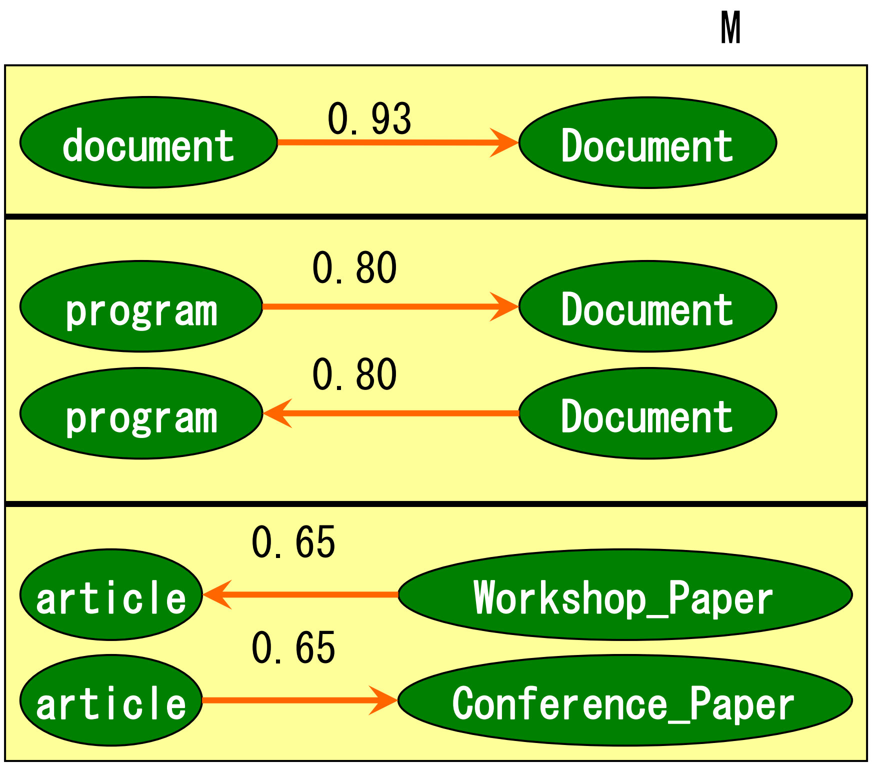
Input: A distributed system $D=\langle O_1,O_2,M\rangle$ and a revision operator

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



$O_1 \cup O_2 \cup M_{.\,0.93}$ is consistent

$O_1 \cup O_2 \cup M_{.\,0.80}$ is inconsistent

Inconsistency degree is 0.80

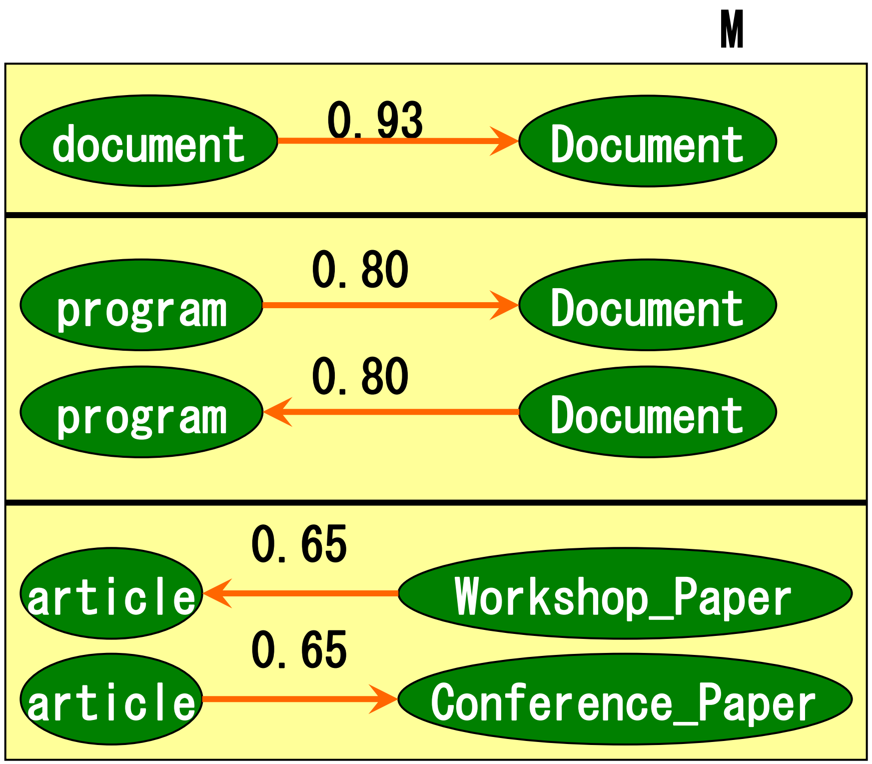
An iterative algorithm for Mapping Revision

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. {document $\subseteq$ Document, Document $\subseteq$ program}

Remove an axiom with the lowest weight

e.g. ax: Document $\subseteq$ program with weight 0.80

$(O_1 \cup O_2 \cup M_{\geq 0.80} \setminus ax)$ 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 $O_1 \cup O_2 \cup M_{>d}$ 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!