First Steps Toward Revising Ontologies

Márcio M. Ribeiro   Renata Wassermann

Instituto de Matemática e Estatística
Universidade de São Paulo

October 5, 2006

Outline of Topics

Belief Revision

When a knowledge base is modified it may become **inconsistent**.
The problem of changing a knowledge base in a rational way is one of the main purposes of belief revision.

AGM Contraction

Definition

Assume K is a belief set ($K = Cn(K)$) and a is a formula an operation $K - a$ is an **AGM contraction** if it satisfies the following properties:

K-1 **(closure)** $K - a = Cn(K - a)$

AGM Contraction

Definition

Assume K is a belief set ($K = Cn(K)$) and a is a formula an operation $K - a$ is an **AGM contraction** if it satisfies the following properties:

K-1 **(closure)** $K - a = Cn(K - a)$

K-2 **(inclusion)** $K - a \subseteq K$

AGM Contraction

Definition

Assume K is a belief set ($K = Cn(K)$) and a is a formula an operation $K - a$ is an **AGM contraction** if it satisfies the following properties:

K-1 **(closure)** $K - a = Cn(K - a)$

K-2 **(inclusion)** $K - a \subseteq K$

K-3 **(vacuity)** If $a \notin K$ then $K - a = K$

AGM Contraction

Definition

Assume K is a belief set ($K = Cn(K)$) and a is a formula an operation $K - a$ is an **AGM contraction** if it satisfies the following properties:

K-1 **(closure)** $K - a = Cn(K - a)$

K-2 **(inclusion)** $K - a \subseteq K$

K-3 **(vacuity)** If $a \notin K$ then $K - a = K$

K-4 **(success)** If $a \notin Cn(\emptyset)$ then $a \notin K - a$

AGM Contraction

Definition

Assume K is a belief set ($K = Cn(K)$) and a is a formula an operation $K - a$ is an **AGM contraction** if it satisfies the following properties:

K-1 **(closure)** $K - a = Cn(K - a)$

K-2 **(inclusion)** $K - a \subseteq K$

K-3 **(vacuity)** If $a \notin K$ then $K - a = K$

K-4 **(success)** If $a \notin Cn(\emptyset)$ then $a \notin K - a$

K-5 **(recovery)** $K = Cn((K - a) \cup \{a\})$

AGM Contraction

Definition

Assume K is a belief set ($K = Cn(K)$) and a is a formula an operation $K - a$ is an **AGM contraction** if it satisfies the following properties:

K-1 **(closure)** $K - a = Cn(K - a)$

K-2 **(inclusion)** $K - a \subseteq K$

K-3 **(vacuity)** If $a \notin K$ then $K - a = K$

K-4 **(success)** If $a \notin Cn(\emptyset)$ then $a \notin K - a$

K-5 **(recovery)** $K = Cn((K - a) \cup \{a\})$

K-6 **(extension)** If $Cn(\{a\}) = Cn(\{b\})$ then $K - a = K - b$

Partial Meet Contraction

The postulates show us which properties a contraction should have, but they don't tell how to build a contraction. One way of building a contraction is called **partial meet**.

Partial Meet Contraction

Definition (Remainder Set)

A **remainder set of K and a** ($K \perp a$) is a maximal subset of K that doesn't imply a. Formally: $K \perp a = \big\{ K' \subseteq K : a \notin Cn(K')$ $\forall K'' \big(K' \subseteq K'' \subseteq K \Rightarrow a \in Cn(K'') \big) \big\}$

Partial Meet Contraction

Definition (Remainder Set)

A **remainder set of K and a** ($K \perp a$) is a maximal subset of K that doesn't imply a. Formally: $K \perp a = \{K' \subseteq K : a \notin Cn(K')\ \forall K''(K' \subseteq K'' \subseteq K \Rightarrow a \in Cn(K''))\}$

Definition (Selection Function)

A **selection function** (γ) choose one subset of a set: $\gamma(K) \subseteq K$

Partial Meet Contraction

Definition (Remainder Set)

A **remainder set of K and a** ($K \bot a$) is a maximal subset of K that doesn't imply a. Formally: $K \bot a = \left\{ K' \subseteq K : a \notin Cn(K') \; \forall K'' \left(K' \subseteq K'' \subseteq K \Rightarrow a \in Cn(K'') \right) \right\}$

Definition (Selection Function)

A **selection function** (γ) choose one subset of a set: $\gamma(K) \subseteq K$

Definition (Partial Meet Contraction)

$K -_\gamma a$ is a **partial meet contraction** iff:

$$K -_\gamma a = \bigcap \gamma(K \bot \alpha) \tag{1}$$

Representation Theorem

The following theorem shows the relation between partial meet contraction and AGM contraction.

Theorem (Representation)

A contraction is **partial meet** *if and only if it is a* **AGM contraction***.*

Motivation

There are many reason for the presence of inconsistency in ontologies:

- mis-representation of defaults

Motivation

There are many reason for the presence of inconsistency in ontologies:

- mis-representation of defaults
- polisemy (words with different meanings)

Motivation

There are many reason for the presence of inconsistency in ontologies:

- mis-representation of defaults
- polisemy (words with different meanings)
- problems in translation between formalisms

Motivation

There are many reason for the presence of inconsistency in ontologies:

- mis-representation of defaults
- polisemy (words with different meanings)
- problems in translation between formalisms
- multiple sources

Example

Classic example (mis-representation of defaults):

$Birds \sqsubseteq Fly$

$Bird(Tweety)$

Example

Classic example (mis-representation of defaults):

$Birds \sqsubseteq Fly$

$Bird(Tweety)$

$\neg Fly(Tweety)$

Example

Classic example (mis-representation of defaults):

$$Birds \sqsubseteq Fly$$

$$Bird(Tweety)$$

$$\neg Fly(Tweety)$$

Inconsistency

Approaches

There are different approaches to deal inconsistencies:

- consistent evolution: prevent introduction of inconsistencies.

Approaches

There are different approaches to deal inconsistencies:

- consistent evolution: prevent introduction of inconsistencies.
- repairing: making a inconsistent ontology consistent.

Approaches

There are different approaches to deal inconsistencies:

- consistent evolution: prevent introduction of inconsistencies.
- repairing: making a inconsistent ontology consistent.
- reasoning with inconsistency: try to derive meaningful conclusion from an inconsistent ontology.

Approaches

There are different approaches to deal inconsistencies:

- consistent evolution: prevent introduction of inconsistencies.
- repairing: making a inconsistent ontology consistent.
- reasoning with inconsistency: try to derive meaningful conclusion from an inconsistent ontology.
- versioning: keep track of changes and compatibly issues between versions.

Approaches

There are different approaches to deal inconsistencies:

- consistent evolution: prevent introduction of inconsistencies.
- repairing: making a inconsistent ontology consistent.
- reasoning with inconsistency: try to derive meaningful conclusion from an inconsistent ontology.
- versioning: keep track of changes and compatibly issues between versions.

Description Logics

Description logics (DLs) are a good formalism for representing ontologies:

- well defined semantics

Description Logics

Description logics (DLs) are a good formalism for representing ontologies:

- well defined semantics
- expressive enough for a huge amount of problems

Description Logics

Description logics (DLs) are a good formalism for representing ontologies:

- well defined semantics
- expressive enough for a huge amount of problems
- decidable inference

Description Logics

Description logics (DLs) are a good formalism for representing ontologies:

- well defined semantics
- expressive enough for a huge amount of problems
- decidable inference
- formalism behind the standard ontology language (OWL)

Logics

A logic $< L, Cn >$ will be represented as it's set of symbols (L) and his consequence operator (Cn).

Definition (Tarskian Logics)

A logic $< L, Cn >$ is **tarskian** iff it satisfies the following properties:

- **(idempotency)** $Cn(A) = Cn(Cn(A))$

Logics

A logic $< L, Cn >$ will be represented as it's set of symbols (L) and his consequence operator (Cn).

Definition (Tarskian Logics)

A logic $< L, Cn >$ is **tarskian** iff it satisfies the following properties:

- **(idempotency)** $Cn(A) = Cn(Cn(A))$
- **(inclusion)** $A \subseteq Cn(A)$

Logics

A logic $< L, Cn >$ will be represented as it's set of symbols (L) and his consequence operator (Cn).

Definition (Tarskian Logics)

A logic $< L, Cn >$ is **tarskian** iff it satisfies the following properties:

- **(idempotency)** $Cn(A) = Cn(Cn(A))$
- **(inclusion)** $A \subseteq Cn(A)$
- **(monotonicity)** If $A \subseteq B$ then $Cn(A) \subseteq Cn(B)$

Logics

A logic $< L, Cn >$ will be represented as it's set of symbols (L) and his consequence operator (Cn).

Definition (Tarskian Logics)

A logic $< L, Cn >$ is **tarskian** iff it satisfies the following properties:

- **(idempotency)** $Cn(A) = Cn(Cn(A))$
- **(inclusion)** $A \subseteq Cn(A)$
- **(monotonicity)** If $A \subseteq B$ then $Cn(A) \subseteq Cn(B)$

Definition (Compact Logic)

A logic $< L, Cn >$ is **compact** iff:

$$a \in Cn(A) \Rightarrow \exists B \subseteq A : a \in Cn(B) \text{ and } B \text{ is finite} \quad (2)$$

The Problem

The problem is that not every tarskian logic admits an AGM contraction. There are logics which don't admit any AGM contraction.

Example

Assume a logic $< L, Cn >$ with:

$$L = \{a, b\}$$
$$Cn(\emptyset) = \emptyset$$
$$Cn(\{a\}) = \{a\}$$
$$Cn(\{b\}) = \{a, b\}$$
$$Cn(\{a, b\}) = \{a, b\}$$

Example

Assume a logic $< L, Cn >$ with:

$$L = \{a, b\}$$
$$Cn(\emptyset) = \emptyset$$
$$Cn(\{a\}) = \{a\}$$
$$Cn(\{b\}) = \{a, b\}$$
$$Cn(\{a, b\}) = \{a, b\}$$
$$K = \{a, b\}$$

Example

Assume a logic $< L, Cn >$ with:

$L = \{a, b\}$

$Cn(\emptyset) = \emptyset$

$Cn(\{a\}) = \{a\}$

$Cn(\{b\}) = \{a, b\}$

$Cn(\{a, b\}) = \{a, b\}$

$K = \{a, b\}$

If $b \in K - a$ then by closure $Cn(\{b\}) = \{a, b\} \subseteq K - a$ but that contradicts success.

Example

Assume a logic $< L, Cn >$ with:

$L = \{a, b\}$

$Cn(\emptyset) = \emptyset$

$Cn(\{a\}) = \{a\}$

$Cn(\{b\}) = \{a, b\}$

$Cn(\{a, b\}) = \{a, b\}$

$K = \{a, b\}$

If $b \in K - a$ then by closure $Cn(\{b\}) = \{a, b\} \subseteq K - a$ but that contradicts success.

But if $K - a = \emptyset$ then $K - a \cup \{a\} = \{a\} \neq K$

AGM compliance [Flouris, Plexousakis and Antoniou (FPA)]

Definition (AGM Compliance)

A logic L is **AGM compliant** iff there is an operator of AGM contraction in L.

AGM compliance [Flouris, Plexousakis and Antoniou (FPA)]

Definition (AGM Compliance)

A logic L is **AGM compliant** iff there is an operator of AGM contraction in L.

Definition (Decomposability)

A logic is L is **decomposable** iff:

$$\forall X, K \subseteq L : Cn(\emptyset) \subset Cn(X) \subset Cn(K)\Big(\exists Z \subseteq L : Cn(X \cup Z) = Cn(K)\Big) \tag{3}$$

AGM compliance [Flouris, Plexousakis and Antoniou (FPA)]

Definition (AGM Compliance)

A logic L is **AGM compliant** iff there is an operator of AGM contraction in L.

Definition (Decomposability)

A logic is L is **decomposable** iff:

$$\forall X, K \subseteq L : Cn(\emptyset) \subset Cn(X) \subset Cn(K) \Big(\exists Z \subseteq L : Cn(X \cup Z) = Cn(K) \Big) \tag{3}$$

Theorem

A logic is **AGM compliant** *iff it is* **decomposable**

Examples

Theorem

Consider a description logic (L) with at least one concept, 2 roles, one of these constructors ($\leqslant_n R$, $\geqslant_n R$, $\forall R.C$ or $\exists R.C$), and that admits the connective $\sqsubseteq$ between concepts and roles and doesn't have constructors for roles ($\neg$, $\sqcup$, $\sqcap$...), then L is not decomposable.

Examples

Theorem

Consider a description logic (L) with at least one concept, 2 roles, one of these constructors ($\leqslant_n R$, $\geqslant_n R$, $\forall R.C$ or $\exists R.C$), and that admits the connective $\sqsubseteq$ between concepts and roles and doesn't have constructors for roles ($\neg$, $\sqcup$, $\sqcap$...), then L is not decomposable.

Follows from this theorem that some important DLs are not decomposable:

- SHIF

Examples

Theorem

Consider a description logic (L) with at least one concept, 2 roles, one of these constructors ($\leqslant_n R$, $\geqslant_n R$, $\forall R.C$ or $\exists R.C$), and that admits the connective $\sqsubseteq$ between concepts and roles and doesn't have constructors for roles ($\neg$, $\sqcup$, $\sqcap$...), then L is not decomposable.

Follows from this theorem that some important DLs are not decomposable:

- SHIF
- SHOIN

Examples

Theorem

Consider a description logic (L) with at least one concept, 2 roles, one of these constructors ($\leqslant_n R$, $\geqslant_n R$, $\forall R.C$ or $\exists R.C$), and that admits the connective $\sqsubseteq$ between concepts and roles and doesn't have constructors for roles ($\neg$, $\sqcup$, $\sqcap$...), then L is not decomposable.

Follows from this theorem that some important DLs are not decomposable:

- SHIF
- SHOIN
- SHIQ

Where does this problem come from?

There are some evidences associating the problem with the recovery postulate. The main evidence is this:

Theorem

Every tarskian logic admits a contraction operator that satisfies the AGM postulates without the recovery postulates.

Where does this problem come from?

There are some evidences associating the problem with the recovery postulate. The main evidence is this:

Theorem

Every tarskian logic admits a contraction operator that satisfies the AGM postulates without the recovery postulates.

So a possible solution should be to replace the recovery postulate.

How should we replace Recovery?

FPA proposed that recovery should be replaced by some postulate with the following properties:

Existence:

Every tarskian logic should admit a contraction satisfying the new set of postulates.

How should we replace Recovery?

FPA proposed that recovery should be replaced by some postulate with the following properties:

Existence:

Every tarskian logic should admit a contraction satisfying the new set of postulates.

Rationality:

For every AGM compliant logic the new set of postulates should be equivalent to the AGM postulates.

Relevance

Hansson has proposed the postulate of **relevance**:

Definition (Relevance)

$K - a$ satisfies **relevance** iff:

$$\forall b \in K \setminus K - a \big(\exists K' : K - a \subseteq K' \subseteq K \wedge a \in Cn(K' \cup \{b\}) \setminus Cn(K')\big) \tag{4}$$

Results

Theorem (Weak Existence)

Every tarskian compact logic admits the contraction operator that satisfies AGM postulates with relevance instead of recovery

Results

Theorem (Weak Existence)

Every tarskian compact logic admits the contraction operator that satisfies AGM postulates with relevance instead of recovery

Theorem (Weak Rationality)

For propositional logic the AGM postulates are equivalent to this new set of postulates

Results

Theorem (Weak Existence)

Every tarskian compact logic admits the contraction operator that satisfies AGM postulates with relevance instead of recovery

Theorem (Weak Rationality)

For propositional logic the AGM postulates are equivalent to this new set of postulates

Theorem (Representation)

For every belief set K closed under compact and tarskian logical consequence, - is a partial meet contraction operation over K if and only if - satisfies the postulates (K-1)-(K-4), (relevance) and (K-6).

Example

Assume a description logic $< L, Cn >$ that admits the connective $\sqsubseteq$ between concepts and roles, and the constructor $\forall$:

- Roles = {enrolledAt, haveClassAt}
- Concept = {SpecialStudent}
- $SS = SpecialStudent$, $e = enrolledAt$, $h = haveClassAt$
- $K = Cn(\{h \sqsubseteq e\}) = Cn(\{h \sqsubseteq e, \forall h.SS \sqsubseteq \forall e.SS\})$

Example

Assume a description logic $< L, Cn >$ that admits the connective $\sqsubseteq$ between concepts and roles, and the constructor $\forall$:

- Roles $=$ {enrolledAt, haveClassAt}
- Concept $=$ {SpecialStudent}
- $SS = SpecialStudent$, $e = enrolledAt$, $h = haveClassAt$
- $K = Cn(\{h \sqsubseteq e\}) = Cn(\{h \sqsubseteq e, \forall h.SS \sqsubseteq \forall e.SS\})$
- By inclusion, success and closure we have: $K - (\forall h.SS \sqsubseteq \forall e.SS) = Cn(\emptyset)$

Example

Assume a description logic $< L, Cn >$ that admits the connective $\sqsubseteq$ between concepts and roles, and the constructor $\forall$:

- Roles = {enrolledAt, haveClassAt}
- Concept = {SpecialStudent}
- $SS = SpecialStudent$, $e = enrolledAt$, $h = haveClassAt$
- $K = Cn(\{h \sqsubseteq e\}) = Cn(\{h \sqsubseteq e, \forall h.SS \sqsubseteq \forall e.SS\})$
- By inclusion, success and closure we have: $K - (\forall h.SS \sqsubseteq \forall e.SS) = Cn(\emptyset)$
- Recovery is not satisfied: $Cn\big(\{\forall h.SS \sqsubseteq \forall e.SS\}\big) \neq K$

Example

Assume a description logic $< L, Cn >$ that admits the connective $\sqsubseteq$ between concepts and roles, and the constructor $\forall$:

- Roles = {enrolledAt, haveClassAt}
- Concept = {SpecialStudent}
- $SS = SpecialStudent$, $e = enrolledAt$, $h = haveClassAt$
- $K = Cn(\{h \sqsubseteq e\}) = Cn(\{h \sqsubseteq e, \forall h.SS \sqsubseteq \forall e.SS\})$
- By inclusion, success and closure we have: $K - (\forall h.SS \sqsubseteq \forall e.SS) = Cn(\emptyset)$
- Recovery is not satisfied: $Cn\big(\{\forall h.SS \sqsubseteq \forall e.SS\}\big) \neq K$
- Relevance is satisfied: Let $K' = Cn(\emptyset)$ and consider the 2 options for β: $h \sqsubseteq e$ and $\forall h.SS \sqsubseteq \forall e.SS$, in both cases $\forall h.SS \sqsubseteq \forall e.SS \in Cn(K' \cup \beta)$.