Inconsistency management in data and knowledge bases (2013)

Rome, Sapienza Università di Roma – June 27th, 2013

Antonella Poggi
Overview

1. Introduction
2. Approaches to updates
3. Updating DL ontologies
4. Several open problems
Overview

1. Introduction
   - Preliminaries
   - Introduction to updates

2. Approaches to updates

3. Updating DL ontologies

4. Several open problems
We already know that a relational database can be seen as a set of first-order formulas, i.e., as a logical first-order theory.

Under the assumption of a complete database, the logical theory is consistent and admits a unique completely determined finite model reflecting the actual world, i.e., the database:

- being a model of the theory, the database instance satisfies all the formulas of the theory, e.g., integrity constraints
- querying and updating a database resort to evaluate queries or perform updates over a finite model
Under the assumption of an incomplete database, the logical theory is consistent and admits more than just one model (finitely or infinitely many).

- incompleteness resorts to not knowing which is the model corresponding to the actual world.

**Remark:** In our investigation of inconsistency management, we showed that in the presence of an inconsistent theory, it is possible to “live with inconsistencies”, by reasoning in terms of a set of repaired models.

- the set of repaired models can be seen as a the set of models represented by an incomplete database!
Closed World vs. Open World Assumption (Reminder)

- According to Reiter’s definition [12]
  - under the Closed World Assumption (CWA), if no “theoretical proof” of a positive fact exists, then the fact is assumed to be false ↦ assuming to have complete information on a database resorts to implicitly make the CWA
  - under the Open World Assumption (OWA), if a fact is not implied or contradicted by the underlying theory, then the fact may be true or false (according to the usual first-order semantics) ↦ assuming to have incomplete information on a database does not resort to make OWA nor CWA, i.e. both have sense in specific distinct scenarios
    - actually, in our investigation of inconsistency management, depending on the scenario, we have been assuming alternatively CWA and OWA
CWA vs. OWA (Example)

- **Scenario 1**: an airline company complete database: all flights and the cities which they connect will be explicitly represented
  - failure to find an entry indicating that there exists a flight connecting Vancouver with Toulouse permits one to conclude that it does not
  - the CWA is the most appropriate in this scenario

- **Scenario 2**: an airline company incomplete database: all flights and the cities which they connect will be explicitly represented, but some null value can appear, e.g., because of an accidental error
  - failure to find an entry indicating that there exists a flight connecting Vancouver with Toulouse, or containing null values that, once replaced with constants, would indicate the same, permits one to conclude that it does not
  - the CWA is the most appropriate in this scenario

- **Scenario 3**: a database resulting from the integration of several autonomous airlines databases
  - nothing can prevent from concluding that a flight connecting Vancouver with Toulouse exists
How knowledge bases come into the picture?

Classical knowledge bases (as opposed to databases and non-monotonic knowledge bases) are typically interpreted according to the usual first-order semantics

- they admit more than one model (and thus implicitly encode incompleteness)
- their semantics is given under the OWA.
Two important features of data and knowledge base systems:

1. answering user queries;
2. computing the data or knowledge base resulting from an update.

In their basic form, updates consist of insertions or deletions

- what happens when an insertion causes a constraint violation, e.g., an egd violation?
- what happens when a deletion causes a constraint violation, e.g., a tgd violation?
In general, an update may concern new knowledge that may or may not be consistent with respect to the original theory.

All approaches to updates (or, in general, to evolution) state that the original theory should change as little as possible if new information is incorporated.

Intuitively, similarly to what happens in consistent query answering, one wants to keep as much knowledge as possible from a theory that has been updated.

Importantly, if the new knowledge is inconsistent with respect to the theory, then one wants to keep as much knowledge as possible that does not contradict the new knowledge.
Basics of updates (2/2)

**Desiderata:** compute a new consistent theory such that:

- it incorporates the new knowledge
- it *minimally differs* from the initial theory

Note that the semantics of updates, similarly to the semantics of consistent query answering, crucially depends on the notion of *minimal distance* between theories and/or set of models.
Updates over complete databases

Suppose that one needs to update a complete database by incorporating a new fact.

A natural semantics:

- if the new fact is consistent with the database, then the update resorts to adding the new fact to the database.
- otherwise, the result of the update is the subset of all repairs that incorporate the new fact.

→ the update over a complete database possibly generates incomplete information, where the incompleteness stems from the fact that it is unknown which repair, among those incorporating the new fact, reflects the actual world.
As expected, the update semantics crucially depends on the notion of minimality that is applied to compute the repairs.

Given the minimality criterion applied to compute the repairs, updates over complete databases generate sets of complete instances, or, equivalently, sets of databases (repairs) computed according to the CWA.
The problem of computing an update over a complete database, that is inconsistent with respect to the database, is very similar to the problem of computing, starting from an inconsistent database, a repaired instance

- the only difference: the repaired instance has to incorporate the new knowledge
- it can be considered as a particular case of the problem of computing a repaired instance, where an instance, in order to be considered a repair, besides minimally differing from the original database, and satisfying possible integrity constraints, it also has to incorporate the new knowledge

In the rest of this lecture, we will not consider anymore this scenario.
Updates over incomplete databases

In the presence of incomplete information, the update semantics is less clear: what does it mean to update a theory having more than just one model? E.g.,

- should the semantics of the update be defined in terms of the models of the theory or in terms of the theory itself?

In the following, we will focus on the investigation of approaches to updates that were proposed in the literature for classical knowledge base, or, equivalently, for incomplete databases under the OWA. Also, we will mostly focus on the case of new knowledge insertions. Analogous considerations can be done for deletions (even though deletions would deserve to be considered on their own).
Overview

1. Introduction

2. Approaches to updates
   - Classification criteria
   - Formula-based update operators
   - Model-based update operators

3. Updating DL ontologies

4. Several open problems
Several approaches to knowledge base updates were proposed in the literature. Refer to [4], for a survey and a study of their complexity in the case of propositional knowledge bases.

In the following, we will classify the main approaches to updates according to the following criteria:

- **Formula-based vs. model-based**: is the semantics of the update defined in terms of changes to the theory itself or in terms of changes to its models?
Classification criteria (2/2)

- **Measure of closeness:** which measure is used to determine the theory, consistent with the update, that is closest to the original theory?
  - **cardinality-based** measure: the distance between two sets is expressed by the number of items in which they differ
  - **inclusion-based** measure: a set $S_1$ is closer to a set $S$ than a set $S_2$ if the elements in which $S_1$ and $S$ differ is a proper subset of the elements on which $S_2$ and $S$ differ, i.e. if

$$S_1 \Delta S \subset S_2 \Delta S$$

(where $A \Delta B$ is the symmetric set difference $(A \cup B) - (A \cap B)$)
In the area of knowledge bases, another change operator that is often considered in the literature is belief revision. The key difference between update and belief revision stems from different perspectives on what a theory is intended to represent:

- if the theory is intended to represent the beliefs about the state of the world, then a new fact is intended to revise our beliefs about the world, while the world remains unchanged.
  \[ \mapsto \] the change is performed by the use of a belief revision operator.

- if the theory is intended to represent a set of worlds that are possible given a limited information currently available, then a new fact reflects an actual change in the world.
  \[ \mapsto \] the change is performed by the use of an update operator.

In the following we will not consider belief revision operators.
If a theory $\mathcal{T}$ is inconsistent with a new fact $F$, a straight solution to gain consistency is to repair the theory itself, by removing the minimal number of facts that contradict the new fact. This simple idea underlies the so-called Set-Of-Theories (SOT) approach [6, 5]. Formally:

$$\mathcal{T} \circ_{\text{SOT}} F = \{\mathcal{T}_i \cup \{F\}| \mathcal{T}_i \text{is a maximal subset of } \mathcal{T} \text{ consistent with } F\}$$

⇒ a formula is true iff it is true in all the theories that result from the update.

**Problem:** one would like the update to return a unique resulting theory! Two formula-based update operators have then been proposed, both providing a distinct solution to the problem of the possibly multiple theories resulting from the SOT approach: the SOT cross-product and the SOT WIDTIO update operators.
The SOT update operator - Example

Let $\mathcal{T} = \{a \land b \Rightarrow c, a, b\}$ and $F = \neg c$.
The maximal subsets of $\mathcal{T}$ that are consistent with $F$ are

$$\{a \land b \Rightarrow c, a\}, \{a \land b \Rightarrow c, b\}, \{a, b\}$$

Hence:

$$\mathcal{T} \circ_{SOT} F = \{\{a \land b \Rightarrow c, a, \neg c\}, \{a \land b \Rightarrow c, b, \neg c\}, \{a, b, \neg c\}\}$$
The SOT cross-product update operator

Suppose that the SOT update returns a finite number of theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$.
Then, the SOT cross-product update operator is defined as follows:

$$\mathcal{T} \circ_{SOT\text{-}cross-prod} F = \bigvee \{\mathcal{T}_i \mid i \in 1, \ldots, n\}$$

where $\bigvee \{\mathcal{T}_i \mid i \in 1, \ldots, n\} = \{p_1 \lor \ldots \lor p_m \mid p_i \in \mathcal{T}_i, 1 \leq i \leq m\}$
i.e., it returns the theory whose formulas are all disjunctions with one formula from each theory $\mathcal{T}_i$.

It can be shown that $\circ_{SOT\text{-}cross-prod}$ fully captures the semantics of the SOT approach to updates, i.e., the set of models of the resulting theory is exactly the union of all models of the theories resulting from the SOT update, i.e.

$$\bigcup_{i=1,\ldots,n} Mod(\mathcal{T}_i) = Mod(\bigvee \{\mathcal{T}_i \mid i \in 1, \ldots, n\})$$
Of course, computing a unique theory as discussed above may not be feasible in practice, given that the number of formulas may grow exponentially in the number of theories resulting from the SOT update. The WIDTIO (When In Doubt Throw It Out) approach \([7, 8]\) solves this problem by defining the following update operator \(\circ_W\):

\[
\mathcal{T} \circ_S OTWF = \bigcap_{i=1,...,n} \mathcal{T}_i
\]

i.e., only the formulas that appear in all the repairs of the theory are retained

- much less expensive to compute
- big loss of knowledge in “bad” cases (all knowledge can be retracted in the worst-case!)
The SOT WIDTIO update operator - Example

Consider again $\mathcal{T} = \{a \land b \Rightarrow c, a, b\}$ and $F = \neg c$.
Then, by following the WIDTIO approach, we have:

$$\mathcal{T} \circ_{SOTW} F = \{a \land b \Rightarrow c, a, \neg c\} \cap \{a \land b \Rightarrow c, b, \neg c\} \cap \{a, b, \neg c\} = \{\neg c\}$$

Note, in particular, that $a \lor b$ is not implied by $\mathcal{T} \circ_{SOTW} F$. 
Pros and cons of formula-based approaches

- **Cons:** in general, formula-based approaches may violate the Dalal’s *Principle of irrelevance of syntax* [3], i.e., the same update may have different effects on equivalent but distinct theories.
- **Pros:** formula-based approaches look easier and often more intuitive to implement.
- **Pros:** formula-based approaches allow to assign priorities to facts, to be taken into account when updating the theory [6].
The semantics of model-based update operators is defined in terms of the models of the theory to be updated. In particular, a model-based update aims at updating each model of theory *separately*. We will see two different update operators, which differ because of the adopted measure of closeness among models: the Forbus’s and the Winslett’s update operators.
The Forbus’s update operator

Define the distance between a model $M$ and a fact $F$ as follows:

$$|\Delta|_M^{min}(F) = \min(|M \Delta M'| : M' \in Mod(F))$$

i.e., as the minimum number of facts in which $M$ and a model of $F$ diverge.

Now, suppose to have a theory $\mathcal{T}$ and a new fact $F$. Forbus’s update operator is defined as follows:

$$Mod(\mathcal{T} \circ_F F) = \bigcup_{M \in Mod(\mathcal{T})} \{ M' \in Mod(F) : |M \Delta M'| = |\Delta|_M^{min}(F) \}$$

i.e., the theory resulting from the Forbus’s operator is such that it captures, for each model $M$ of $\mathcal{T}$, the models of $F$ that are at minimum distance from $M$. 
The Forbus’s update operator - Example

Consider $T = \{a, b, c\}$ and $F = \{\neg a \land d \lor a \land \neg b \land \neg c \land \neg d\}$. On facts $a, b, c, d$, $T$ has two models that are $M_1 = \{a, b, c\}$ and $M_2 = \{a, b, c, d\}$. $F$ is inconsistent with $T$, then $|\Delta|_{M_i}^{\min}(F) > 0$ for $i = 1, 2$.

- Consider $M_1$:
  - there are no models of $F$ that are at distance 1 from $M_1$;
  - there are two models of $F$ that are at distance 2 from $M_1$:
    $F_{11} = \{a\}$ and $F_{12} = \{b, c, d\}$

- Consider $M_2$: there is one model that is at distance 1 from $M_2$:
  $F_{21} = \{b, c, d\}$.

Therefore:

$$T \circ_F F = (a \land \neg b \land \neg c \land \neg d) \lor (\neg a \land b \land c \land d)$$
The Winslett’s update operator

Define the distance between a model $M$ and a fact $F$ as follows:

$$\Delta_{\text{min}}^{M}(F) = \min_{\subseteq}(\{ M \Delta M' : M' \in \text{Mod}(F) \})$$

i.e., as the set of minimal symmetric differences between $M$ and the models of $F$ diverge.

Now, suppose to have a theory $T$ and a new fact $F$. Winslett’s update operator is defined as follows:

$$\text{Mod}(T \circ_{W} F) = \bigcup_{M \in \text{Mod}(T)} \{ M' \in \text{Mod}(F) : M \Delta M' \in \Delta_{\text{min}}^{M}(F) \}$$

i.e., the theory resulting from the Winslett’s operator is such that it captures, for each model $M$ of $T$, the models of $F$ that minimally differ from $M$, according to the inclusion-based measure of closeness among models.
Consider $\mathcal{T} = \{a \equiv \neg b\}$ and $F = a$. Note that $F$ is consistent with $\mathcal{T}$. On facts $a, b$, $\mathcal{T}$ has two models that are $M_1 = \{a\}$ and $M_2 = \{b\}$.

- Consider $M_1$: the model of $F$ that minimally differs from $M_1$ is $M_1$ itself.
- Consider $M_2$: the model of $F$ that minimally differs from $M_2$ is $\{a, b\}$.

Therefore:

$$\mathcal{T} \circ_W F = a$$
Overview

1. Introduction

2. Approaches to updates

3. Updating DL ontologies
   - Applying model-based approaches to DL evolution
   - Instance-level update

4. Several open problems
Updating DL ontologies

Suppose to have a DL ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ expressed in a fixed DL $\mathcal{L}$ (in which an ABox contains only ground atoms).

Depending on the particular scenario, evolution might affect:

- the intensional level, i.e., the TBox $\mathcal{T}$
- the extensional level, i.e., the ABox $\mathcal{A}$

Motivated by Ontology-Based Data Access (OBDA), we will focus here on evolution operators under the following assumptions:

- we are interested in both insertion and deletion of chunks of knowledge (for the sake of clarity, we will consider here only the case of insertion)
- evolution affects only the instance level of the KB, i.e., the TBox remains unchanged
- the result should be expressed in $\mathcal{L}$
- the result should be independent of the syntactic form of the original KB
Defining the evolution operators in terms of sets of models, gives rise to the following **evolution expressibility problem**:

*Given an ontology $\mathcal{O}$ expressed in a language $\mathcal{L}$, a ground fact $F$ and an evolution operator $\circ$ such that $\mathcal{O} \circ F = \mathcal{M}$ where $\mathcal{M}$ is the set of models obtained through the update.*

*Is there an ontology $\mathcal{O}'$ expressed in $\mathcal{L}$ such that $\text{Mod}(\mathcal{O}') = \mathcal{M}$?*

It has been shown that for very simple DL languages, the update is not expressible (see e.g., [11]).
Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, expressed in a DL $\mathcal{L}$, an instance-level update of $\mathcal{O}$ with a new ground fact $F$ is such that it returns a theory that:

- results from the update of $\mathcal{O}$ with $F$, and, as such, it
  - incorporates $F$
  - is “as close as possible” to $\mathcal{O}$ (either in terms of its models or in terms of its formulas)
- is consistent with $\mathcal{T}$. 
Instance-level updates over DL-Lite ontologies

Model-based approaches to instance-level update have been proved to be not suitable: even over DL-Lite core ontologies, updates are not expressible [2, 9].
Another issue with model-based evolution operator is that their behavior might be very counterintuitive.
A counterintuitive behaviour of the Winslett’s update operator

Example

\[ \mathcal{T} : \begin{align*} & \text{Male} \sqsubseteq \text{Human}, \quad \text{Female} \sqsubseteq \text{Human}, \quad \text{Male} \sqsubseteq \neg \text{Female} \\ & \text{Human} \sqsubseteq \exists \text{HasFather}, \quad \exists \text{HasFather} \neg \sqsubseteq \text{Male} \end{align*} \]

\[ \mathcal{A} : \{ \text{Male}(\text{Mario}), \text{Human}(\text{Andrea}) \}, \]

\text{insert } F = \{ \text{Female}(\text{Andrea}) \}

There exists at least one model \( M \) of \( \langle \mathcal{T}, \mathcal{A} \rangle \) in which \( \text{HasFather}(\text{Mario}, \text{Andrea}) \) is true. Now, suppose to update such a model to make it satisfy \( F \). We have to remove from \( M \) the fact \( \text{HasFather}(\text{Mario}, \text{Andrea}) \).
A counterintuitive behaviour of the Winslett’s update operator

Example (cont.)

Then, we obtain the following models of $\langle T, F \rangle$ that are at minimum distance from $M$:

- $M'$: obtained by removing the fact $\text{Human}(\text{Mario})$
- the family of models $M''_{c_i}$: obtained by adding the facts $\text{HasFather}(\text{Mario}, c_i)$, for all constants $c_i$ within the alphabet of constants

Hence, by stating $\text{Female}(\text{Andrea})$ we loose certainty about the fact $\text{Human}(\text{Mario})$!
A “formula-based” approach

In order to overcome model-based approaches drawbacks, in [2, 10] the authors propose to adapt to DL-Lite ontologies, the SOT approach but then one has to deal with the problem of multiple update results.
The problem of multiple results

Example

\( \mathcal{T} : \exists R.C \sqsubseteq B, \ B \sqsubseteq \neg D, \ B \sqsubseteq E \)

\( \mathcal{A} : \{ R(a_1, a_2), C(a_2) \}, \ \text{with} \)

\( \text{cl}_\mathcal{T}(\mathcal{A}) = \{ R(a_1, a_2), C(a_2), B(a_1), E(a_1) \} \)

insert \( F = \{ D(a_1) \} \)

\( \mathcal{A}_1 = \{ R(a_1, a_2), D(a_1), E(a_1) \}, \ \text{with} \ \text{cl}_\mathcal{T}(\mathcal{A}_1) = \mathcal{A}_1 \)

\( \mathcal{A}_2 = \{ C(a_2), D(a_1), E(a_1) \}, \ \text{with} \ \text{cl}_\mathcal{T}(\mathcal{A}_2) = \mathcal{A}_2 \)
The problem of multiple results

In order to get a unique DL-Lite ontologies through the update, two different approaches have been proposed:

- in [10], the authors propose to follow an adaptation of the WIDTIO approach
- in [2], the authors propose to choose one ABox nondeterministically, which they call **bold semantics**.
The problem of multiple results

Example

\[ \mathcal{T} : \exists R.C \sqsubseteq B, \ B \sqsubseteq \neg D, \ B \sqsubseteq E \]
\[ \mathcal{A} : \{R(a_1, a_2), C(a_2)\}, \ \text{with} \]
\[ \text{cl}_\mathcal{T}(\mathcal{A}) = \{R(a_1, a_2), C(a_2), B(a_1), E(a_1)\} \]

insert \( F = \{D(a_1)\} \)

According to the WIDTIO approach, the ABox resulting from the update is the following:
\( \mathcal{A}' = \{D(a_1), E(a_1)\} \).

While according to the bold semantics, the ABox resulting from the update is one among \( \mathcal{A}_1 \) and \( \mathcal{A}_2 \).
An algorithm for computing the result of an update by insertion:

1. compute all ABoxes accomplishing the insertion minimally,
2. compute their intersection.

**Challenge:** compute the result without computing all ABoxes accomplishing the insertion minimally.
Overview

1. Introduction
2. Approaches to updates
3. Updating DL ontologies
4. Several open problems
Several open problems

- “Write-also” OBDA
- Investigate a model-based semantics for DL knowledge bases, that does not give rise to any counterintuitive behavior
- Investigate updates over incomplete databases, under CWA
  - non-monotonic reasoning is likely to drastically change the picture!
  - cf. algebraic approach proposed in [1]
- Many many others....
Acknowledgements

- Maurizio Lenzerini
- Domenico Fabio Savo
References

Update semantics for incomplete databases. 
In Proc. of the 11th Int. Conf. on Very Large Data Bases (VLDB’85), pages 1–12, 1985.

Evolution of DL-Lite knowledge bases. 

Investigations into a theory of knowledge base revision. 
References II

On the complexity of propositional knowledge base revision, updates and counterfactuals.

Updating logical databases.

On the semantics of updates in databases.

Counterfactuals.
References III

Reasoning about action i: A possible worlds approach.

Capturing model-based ontology evolution at the instance level: the case of dl-lite.

On the evolution of the instance level of *DL-Lite* knowledge bases.
Updating description logic ABoxes.

On closed world data bases.