TEODOR C. PRZYMUSIŃSKI

UPDATES OF DEDUCTIVE KNOWLEDGE BASES

STRESZCZENIE

Nieustannie stajemy przed koniecznością *uaktualniania* wiedzy, którą już posiadamy. Przeprowadzanie częstych aktualizacji pozwala nam poznać nowe fakty i odrzucić te, które już nie znajdują zastosowania. W artykule opisano sposoby uaktualniania wiedzy przy użyciu podejścia zwanego *interpretacją* lub *światem możliwym*. Pokazano jak można wykorzystać zdroworozsądkową wersję newtonowskiej zasady bezwładności do scharakteryzowania w sposób elegancki i prosty powyższych uaktualnień, co właściwie podsumowuje istotną klasę programów adiustacyjnych (za Wiktorem Markiem i Mirosławem Truszczyńskim) i podejście uaktualniania formuł (za Marianne Winslett).

INTRODUCTION

We are continuously faced with the need to *update* the knowledge that we possess. Performing frequent updates allows us to learn new facts and discard those that no longer apply. One can reasonably argue that most, if not all, of our knowledge is acquired via suitable sequences of updates. By extension, the same applies to *intelligent artifacts*, such as intelligent knowledge bases and intelligent agents.

However, performing meaningful updates is by no means a trivial task. First of all, we have to be able to properly represent our knowledge and provide it with a precise meaning or *semantics*. Intelligent artifacts, such as databases and intelligent agents, must be provided with a mathematically precise representation of the knowledge. Moreover, such knowledge must be represented in a manner, which is independent of procedural considerations, easy to communicate, exchange and reason about.

We must be able to handle updates in a dynamically changing world and thus we have to decide how to resolve contradictions arising from new updates. Quite often, adding a new fact or a new rule to our knowledge base forces us to remove some other facts or rules that are inconsistent with the newly acquired information. At the same time, we want to make sure that the set of facts that need to be revised or outright deleted is, in some precise sense, *minimal*. Finally, we need to choose between different approaches to and *types of updates*. After all, updates were quite extensively studied in the literature for a number of years now.

In this presentation, I will describe updates using the so-called *interpretation* or *possible world* approach. We will show how a commonsense version of *Newton's principle of inertia* can be used to provide an elegant and simple characterization of such updates which properly subsumes the important class of revision programs, due to Marek and Truszczynski [MT, MT2], and the formula update approach, due to Winslett [W].

The presentation is intended to be introductory and accessible to non-experts. Consequently, it will only touch upon the important issues involved. The presented results were obtained jointly with *Hudson Turner*. For details we refer the reader to our joint paper [PT] and to subsequent papers on dynamic logic programs [APPP, ALPPP].

RELATIONAL AND DEDUCTIVE KNOWLEDGE BASES

Relational knowledge bases are the simplest databases. They consist of a set of facts (atoms) which are to be true in the database. For example:

```
Tigger(tigger),  Food(honey),  Food(acorns), Food(malt), Likes(tigger, malt)
```

Facts which are *not* included in the database are assumed *false:*

```
¬Tigger(pooh), ¬Food(tigger), ¬Likes(tigger, acorns), ¬Likes(tigger, honey)
```

On the other hand, *deductive knowledge bases* are significantly more general, expressive and powerful. In addition to facts, they also allow powerful *deductive rules* like the following one:

```
Tigger(X) ∧ Food(Y) ∧ Y ≠ honey → Like(X,Y)
```

which is meant to say that tigers like all kinds of food except honey. Similarly, the following rule:

```
X = susan → Fly(X, korea) ∨ Fly(X, japan)
```

states that Susan is about to fly either to Korea or to Japan.

A *model* of a deductive knowledge base (KB) is a relational knowledge base (called an interpretation) which *satisfies all the rules* of KB. For example, given the knowledge base KB:

```
Tigger(tigger),  Food(honey)
Tigger(X) ∧ Food(Y) → Like(X,Y)
```

these are *two* sample models of KB:

```
Tigger(tigger),Food(honey),Like(tigger,honey)
Tigger(tigger),Food(honey),Like(tigger,honey),Food(tigger),Like(tigger,tigger)
```

The first of these models is also *minimal*, because it includes a minimal set of facts that must be true is the database. For example, the following interpretation is not a model of KB because it fails to include the fact `Like(tigger,honey)` :

```
Tigger(tigger),Food(honey),Like(tigger,tigger)
```

Examples of Updates

Let us consider the following dialogue based on the story from the famous Milne's book about Winnie-the-Pooh:

Whatever-it-was: *Worraworraworraworra!*
Pooh: *Who is it?*
Whatever-it-was: *I am Tigger.*
Pooh: *Oh! (Pooh had never seen an animal like this)*
Pooh: *Do Tiggers like honey?*
Tigger: *Tiggers like everything*

We can represent part of this dialogue as follows:

```
Tigger(whatever-it-was)
Tigger(X) ∧ Food(Y) → Like(X,Y)
```

However, after tasting honey, Tigger says:
Tigger: *Tiggers don't like honey*

```
Tigger(X) →  ¬Like(X, honey)
```

which results in a dynamic update to the previous information about culinary likings of Tigger:
Tigger: *Tiggers like everything except honey*

```
Tigger(X) ∧ Food(Y) ∧ Y ≠ honey → Like(X,Y)
```

Subsequently, Pooh asks:
Pooh: *Do Tiggers like acorns?*
Tigger: (after tasting) *Tiggers don't like acorns*

```
Tigger(X) →   ¬Like(X, acorns)
```

Which results in yet another dynamic update to the previous information:
Tigger: *Tiggers like everything except honey and acorn*

```
Tigger(X) ∧ Food(Y) ∧ Y ≠ honey ∧ Y ≠ acorn → Like(X,Y)
```

Let us look at a slightly different example. Susan is traveling abroad but we are not quite sure about the destination. It may be either Korea or Japan:

```
Fly(susan, korea)  ∨   Fly(susan, japan)
```

However, we subsequently learn that all flights to Japan are cancelled:

```
∀X  ¬Fly(X, japan)
```

Should we conclude that Susan flies to Korea?

```
Fly(susan, korea) ?
```

Not necessarily, because if Susan actually planned to go to Japan and suddenly all flights to Japan got cancelled, it does not mean that she will decide to go to Korea instead. The right update may be that she might not go anywhere. As we can see, updating deductive knowledge is not such a straightforward task...

DEDUCTIVE UPDATES CAN BE REDUCED TO INTERPRETATION UPDATES

As we already mentioned, updates of relational databases (interpretations) are much easier than updates of deductive knowledge bases. However, as shown by Winslett, Katsuno and Mendelzon [W, KM], it is possible to reduce updates of deductive databases to interpretation updates:

Definition 1. ([W], [KM]) In order to update a deductive knowledge base KB proceed as follows:

- Update individually all of its relational models
- The set of formulae (DB rules) that are true in *all* of the so updated models is the updated KB'.

Consider the following sample KB:

```
Tigger(tigger),  Food(honey),  Food(acorns)
Tigger(X) ∧ Food(Y) → Like(X,Y)
```

Sample models of KB (using obvious abbreviations):

```
T(t),  F(h), F (a), L(t, h), L(t, a)
T (t), F(h), F (a), L(t, h), L(t, a), F(t), L(t, t)
```

The first one is also a minimal model. Now suppose that we get an update:

```
¬Like(tigger, honey)
```

Now, updated models of KB, *with ¬Like(tigger, honey)* removed, are:

```
T (t),  F(h), F (a), L(t, a)
T (t), F(h), F (a), L(t, a), F(t), L(t, t)
```

And therefore the following rule holds in the updated database KB':

```
Tigger(X) ∧ Food(Y) ∧  Y ≠  honey → Like(X,Y)
```

Now, let's look at the travelling Susan example:

```
Fly(susan, korea)  ∨  Fly(susan, japan)
```

All minimal models of KB are:

```
Fly(susan, korea)
Fly(susan, japan)
```

Now suppose that we get an update:

```
∀X  ¬Fly(X, japan)
```

Updated models of KB, *with* `Fly(susan, japan)` *removed, are*:

```
Fly(susan, korea)
  <empty model>
```

As we can see, the updated database KB' does *not* imply `Fly(susan, korea)` because it also has an empty model.

REVISION UPDATES

As shown in the previous section, in order to perform updates of deductive knowledge bases, all we need to do is to be able to update individual models od such databases. An interesting approach to such updates has been proposed by Marek and Truszczynski [MT, MT2] and called *revision updates*.

Let us look at the following example due to Marek and Truszczynski:

Example. *([MT], [MT2])* A manager reorganizes her department that currently includes Barbara,

Chuck, Diane, Ed and Fay. She could hire Greg and Hanna. If she keeps Barbara then she must fire Chuck. If Barbara is fired then Dianne will leave. She can't do without Greg. If she hires Greg then she must fire Ed. If she fires Chuck then she must hire Hanna.

Initial knowledge KB:

Barbara, Chuck, Diane, Ed, Fay

Update rules:

Barbara → ¬*Chuck*
¬*Barbara* → ¬*Diane*
Greg
Greg → ¬*Ed*
¬*Chuck* → *Hanna*

Resulting updated knowledge base KB':

Barbara, Greg, Diane, Fay, Hanna

Now, let us add a new update rule: ***Fay*** → ¬***Barbara***
Current update rules:

Barbara → ¬*Chuck*
¬*Barbara* → ¬*Diane*
Greg
Greg → ¬*Ed*
¬*Chuck* → *Hanna*
Fay → ¬*Barbara*

Resulting updated knowledge base KB'':

Chuck, Greg, Fay

AN ELEGANT CHARACTERIZATION OF REVISION UPDATES

We are all familiar with *Newton's First Law*, also known as the *law of inertia*:

> **Law of Inertia:** „*A body remains at rest or moves with constant velocity in a straight line, unless it is compelled to change its state by an unbalanced force acting upon it".*

Consider the following variation of this law that is suitable in the context of database updates:

> **Commonsense Law of Inertia:** *Any fact that is true or false in the initial interpretation* **M** *remains so unless it is forced to be changed by update rules* **U**.

Introduce the following meta-rule:

Commonsense Inertia Rules I_M: [PT] Given the initial interpretation **M**:

- For any atom *A* in M that is initially *true* we add **Unless(¬A) →A;**
- For any atom *A* that is initially *false* we add **Unless(A) → ¬A.**

Let us now see how this will work:

Initial: Barbara, Chuck, Diane, Ed, Fay		
$Barbara \rightarrow \neg Chuck$ $\neg Barbara \rightarrow \neg Diane$ $Greg$ $Greg \rightarrow \neg Ed$ $\neg Chuck \rightarrow Hanna$ $Fay \rightarrow \neg Barbara$	$Unless(\neg B) \rightarrow B$ $Unless(\neg C) \rightarrow C$ $Unless(\neg D) \rightarrow D$ $Unless(\neg E) \rightarrow E$ $Unless(\neg F) \rightarrow F$ $Unless(G) \rightarrow \neg G$ $Unless(H) \rightarrow \neg H$	1 G 2 $\neg E$ 3 F 4 $\neg B$ 5 $\neg D$ 6 C 7 $\neg H$
Result: Chuck, Fay, Greg		

On the left side, we have a list of update rules. In the middle, we added the inertia rules I_M. *In the table on the right, we list a sequence of derivations obtained by applying both the update rules and the commonsense inertia rules. As one can see, we obtained the same result as before.*

Definition. *Let M be any interpretation. Define the full interpretation* $\underline{M}$ *to be:*

$$\underline{M} = \{A : A \in M\} \cup \{\neg A : A \notin M\}$$

Now we are ready to present the main results due to Przymusinski and Turner [PT]:

Theorem 1. *[PT] Let M be any interpretation and let U be any set of revision (update) rules. An interpretation M' is a revision update of M if and only if* $\underline{M'}$ *is a model of the database* $P = U \cup I_M$ *obtained by augmenting U with the inertia rules* I_M*:*

$$M' \text{ is update of } M \Leftrightarrow \underline{M'} \text{ is model of } U \cup I_M$$

Theorem 2. *[PT] Let M be any interpretation and let U be any set of revision rules. An interpretation M' is a revision update of M if and only if it coincides with the set of facts derivable from the database* $\mathbf{P = U \cup (\underline{M} \cap \underline{M'})}$*, obtained by augmenting U with facts that are true in both interpretations. In other words:*

$$M' \text{ updates } M \Leftrightarrow \underline{M'} \text{ derivable from } U \cup (\underline{M} \cap \underline{M'})$$

Clearly, the set $\mathbf{\underline{M} \cap \underline{M'}}$ *represents those facts that are inherited by M' from M by inertia.*

The above results provide a very simple and elegant characterization of revision updates by means of the commonsense inertia rules.

EPILOGUE

We described a broad and yet simple framework for handling interpretation updates. Our framework strictly generalizes and significantly simplifies previously introduced concept of *revision updates*. It also generalizes the so called *formula updates* [PT].

It can be shown that the framework is naturally embeddable into *default logic*, a well-known non-monotonic logic due to Reiter [R]. It also allows powerful extensions that are capable to relate updates more closely to the structure of a KB:

- *Dealing with uncertainty:* we have to be able to handle partial updates
- *Handling rule inertia:* the rule of inertia must be applied not just to interpretations but rather to the rules of the program.
- *Truly Dynamic Updates:* a knowledge base encodes much more than just the set of its models, namely, knowledge rules encode important relationships between their elements

These extensions, discussed in detail in [APPP, ALPPP], represent yet much more powerful and complex update frameworks.

REFERENCES

[ALPPP] Alferes J., Leite J., Pereira L., Przymusinska H., Przymusinski T., *Dynamic Updates of Non--Monotonic Knowledge Bases*, „Journal of Logic Programming" 2000, Vol. 45, No. 1-3, pages 43-70

[APPP] Alferes J., Leite J., Pereira L., Przymusinska H., Przymusinski T., *LUPS – a language for updating logic programs*, „Artificial Intelligence Journal" 2002, Vol. 138, No. 1-2, pages 87-116

[KM] Katsuno H., Mendelzon A.O., *On the difference between updating a knowledge base and revising*, in: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, 1991, pages 387-394

[MT] Marek W., Truszczynski M., *Revision specifications by means of revision programs*, in: *Logics in AI: Proceedings of JELIA '94, Lecture Notes in Artificial Intelligence*, Springer-Verlag, 1994

[MT2] Marek W., Truszczynski M., *Revision programming, database updates and integrity constraints*, in: *Proceedings of the 5th International Conference on Database Theory – ICDT 95*, Springer-Verlag, 1995, pages 368-382

[PT] Przymusinski T., Turner H., *Update By Means of Inference Rules*, „Journal of Logic Programming" 1997, Vol. 30, No. 2, pages 125-143

[R] Reiter R., *A logic for default reasoning*, „Artificial Intelligence"1980, Vol. 13, No. 1-2, pages 81-132

[W] Winslett M., *Reasoning about action using a possible models approach*, in: *Proceedings AAAI-88*, 1988, pages 89-93