ARTINT 971

Research Note

On prediction in Theorist

Michael Thielscher

Fachgebiet Intellektik, Fachbereich Informatik, Technische Hochschule Darmstadt, Alexanderstraße 10, D-6100 Darmstadt, Germany

Received December 1991
Revised July 1992

Abstract

Thielscher, M., On prediction in Theorist (Research Note), Artificial Intelligence 60 (1993) 283-292.

Theorist is a well-known framework and system for nonmonotonic reasoning which provides mechanisms for dealing with both explanations for observations and skeptical prediction. Its current implementation, developed by David Poole and co-workers, uses an algorithm for prediction which holds for a restricted part of Theorist. In more general cases, the system produces incorrect results in the case of prediction.

In this note, we present an algorithm for prediction which is shown to be correct within the entire Theorist framework.

1. Introduction

In the past ten years, many formalisms for nonmonotonic reasoning have been developed. One direction of research consists in the use of classical first-order logic along with theory formation. Beside Reiter's default logic [9] the most prominent approach using this idea is described by David Poole in [5].

In Poole's so-called *Theorist* framework, knowledge about a world is represented by several consistent sets of first-order formulas. These sets are used

Correspondence to: M. Thielscher, Fachgebiet Intellektik, Fachbereich Informatik, Technische Hochschule Darmstadt, Alexanderstraße 10, D-6100 Darmstadt, Germany. Telephone: (+49-6151) 16-3638. Fax: (+49-6151) 16-5326. E-mail: thielscher@intellektik.informatik.th-darmstadt.de.

0004-3702/93/$06.00 © 1993 — Elsevier Science Publishers B.V. All rights reserved

to describe different scenarios, or to represent assumptions which provide an explanation for given observations. Initially, two sets are introduced, one containing *facts*, i.e. formulas which are supposed to be true in every scenario. The elements of the other set are *defaults*. Defaults are used as assumptions for explanations as long as they do not lead to an inconsistency. In [5], Poole also argues that some formulas called *constraints* are necessary to restrict the use of defaults in certain contexts. Constraints need to be distinguished from facts, because they are used only for checking consistency but not for providing explanations.

In a subsequent article, Poole describes how to use his framework for skeptical prediction [8], but considers only the basic version without constraints. Poole presents an algorithm for prediction together with a proof of correctness. However, within the entire Theorist framework which includes constraints, these results are no longer valid. This also applies to the current Theorist implementation developed by Poole [4], where this algorithm is used together with constraints.

Constraints in Theorist are used to suppress unintuitive extensions. As regards skeptical prediction, a statement is believed if and only if it is in every extension. Therefore, the ability of using constraints seems to be inevitably necessary for skeptical reasoning close to the intuition.

The aim of this paper is to show how to reconcile prediction with constraints. This is not a trivial problem, as can be seen by the fact that the current implementation produces incorrect answers in the case of prediction. The reason for this abnormal behaviour is that there is a need for a distinction between the consistency of two scenarios and a weaker notion called *compatibility*. We use this term to suppress uniting two scenarios if their union is no valid scenario.

The following section gives a brief summary of Theorist. We present a small example illustrating how to use this framework. The example also shows why the existing algorithm for prediction produces unintuitive and incorrect results (with respect to theory) when using constraints. In Section 3, we describe the idea of our algorithm. In Section 4, we formalize this idea culminating in the main theorem of this paper and the resulting algorithm. In the last section, we discuss our result within the context of other approaches to nonmonotonic reasoning.

2. Theorist framework

The knowledge of the world we want to model is represented by three sets of first-order formulas, i.e. a world W is defined as a triple $(\mathcal{F}, \Delta, \mathcal{C})$ where:

- The set $\mathcal{F}$ is called the set of *facts*. Facts are closed formulas which are assumed to be true.

- The set Δ is called the set of *defaults*. Any ground instance of some (possibly open) default may be used for theory formation.
- The set $\mathcal{C}$ is called the set of *constraints*. A constraint is a closed formula used to suppress one or more defaults under certain circumstances.

It is assumed that $\mathcal{F} \cup \mathcal{C}$ is consistent.

One restriction normally made is to allow a default only to be an atomic formula. In fact, this is done by naming every default $d(\overline{x})$, where $\overline{x}$ denotes the free variables in d, by some atomic formula $n(\overline{x})$, adding this name to the set of defaults and taking the formula $\forall\overline{x}(n(\overline{x}) \rightarrow d(\overline{x}))$ as a fact. Poole has shown that there is no loss of expressiveness making this restriction [5]. Using this method, the defaults mentioned in the set of constraints are referred to by their name.

We are now ready to state the basic definitions for using these sets.

Definition 2.1. A *scenario* of $\mathcal{W} = (\mathcal{F}, \Delta, \mathcal{C})$ is a set $\mathcal{F} \cup D$ such that D is a set of ground instances of elements of Δ and $\mathcal{F} \cup D \cup \mathcal{C}$ is consistent.

Definition 2.2. An *explanation* of a closed formula g from $\mathcal{W} = (\mathcal{F}, \Delta, \mathcal{C})$ is a scenario of $\mathcal{W}$ which (classically) implies g.

Definition 2.3. An *extension* of $\mathcal{W} = (\mathcal{F}, \Delta, \mathcal{C})$ is the set of logical consequences of a maximal (with respect to set inclusion) scenario of $\mathcal{W}$.

Definition 2.4. A closed formula g is *predictable* in $\mathcal{W} = (\mathcal{F}, \Delta, \mathcal{C})$ iff it is in every extension of $\mathcal{W}$.

For the following discussion we introduce a new relation between scenarios. Two scenarios of a world $\mathcal{W}$ are called *incompatible* if their union is no scenario of $\mathcal{W}$.

Definition 2.5. Two scenarios $\mathcal{S}_1 = \mathcal{F} \cup D_1$ and $\mathcal{S}_2 = \mathcal{F} \cup D_2$ of $(\mathcal{F}, \Delta, \mathcal{C})$ are *incompatible* iff $\mathcal{F} \cup D_1 \cup D_2 \cup \mathcal{C} \models \perp$.

The following example shows how to use these definitions to model nonmonotonic knowledge. It also shows why the notion of compatibility is essential for computing predictions.

Example 2.6. Let Joe be the protagonist of our example. Joe was born in New Orleans but some years ago he moved to San Francisco. Joe is a great football fan.

It is a well-known fact that anybody born in New Orleans identifies himself with anything from his city, as does anybody living in San Francisco. To represent this in our formalism we need the facts

$$\begin{aligned}\mathcal{F} = \{&born\_in\,(joe,\ new\_Orleans),\\ &lives\_in\,(joe,\ san\_Francisco),\\ &likes\_football\,(joe),\\ &\forall X, Y\,(born\_in\,(X,Y) \rightarrow identifies\_himself\_with\,(X,Y)),\\ &\forall X, Y\,(lives\_in\,(X,Y) \rightarrow identifies\_himself\_with\,(X,Y))\}\end{aligned}$$

If we know that someone is a football fan and that he identifies himself with everything from New Orleans, then we may assume that he is a fan of the New Orleans Saints. On the other hand, if we know that some football fan identifies himself with everything from San Francisco, then we may assume that he is a fan of the San Francisco 49ers. Formalizing these two statements as defaults, we get

$$\begin{aligned}\Delta = \{&default\_Saints\,(X),\\ &default\_S.F.49ers\,(X)\}\end{aligned}$$

and

$$\begin{aligned}\mathcal{F}' = \mathcal{F} \cup \{&\forall X\,(default\_Saints\,(X) \rightarrow likes\_football\,(X)\\ &\qquad \wedge\ identifies\_himself\_with\,(X,\ new\_Orleans)\\ &\qquad\quad \rightarrow fan\_of\_Saints\,(X)),\\ &\forall X\,(default\_S.F.49ers\,(X) \rightarrow likes\_football\,(X)\\ &\qquad \wedge\ identifies\_himself\_with\,(X,\ san\_Francisco)\\ &\qquad\quad \rightarrow fan\_of\_S.F.49ers\,(X)).\end{aligned}$$

A conflict arises if a person identifies himself with everything from New Orleans and San Francisco at the same time, since normally nobody is a fan of both football teams (especially when there is a match between the Saints and the 49ers). Therefore, we need a constraint to suppress the use of the defaults with the same instantiation.

$$\mathcal{C} := \{\forall X\,(\neg\,(default\_Saints\,(X) \wedge default\_S.F.49ers\,(X)))\}$$

Now, if we observe Joe being a fan of the Saints, then the constraint does not prevent us to *explain* this by using the first default. But we are not able to *predict* Joe being a fan of one of these teams from what we know, since there are two different scenarios

$$\begin{aligned}\mathcal{S}_1 &= \mathcal{F}' \cup \{default\_Saints(joe)\},\\ \mathcal{S}_2 &= \mathcal{F}' \cup \{default\_S.F.49ers(joe)\}.\end{aligned}$$

We do not want to add as a fact that anybody cannot be a fan of both teams, because we will not exclude this. The aim of the constraint is to permit the conclusion that someone is a fan of both teams just from the fact that he identifies himself with everything from New Orleans and San Francisco.

Here, the crucial point is that $\mathcal{S}_1$ and $\mathcal{S}_2$ are incompatible, since $\mathcal{S}_1 \cup \mathcal{S}_2$ is no scenario of $(\mathcal{F}', \Delta, \mathcal{C})$, but $\mathcal{S}_1 \cup \mathcal{S}_2$ is not inconsistent. The algorithm used within the Theorist implementation [4] predicts a formula if it is implied by a scenario which is consistent with all other scenarios. Therefore, this algorithm predicts any formula implied by $\mathcal{S}_1$, especially "*fan_of_Saints(joe)*" although this formula is not in every extension.

3. The idea of the algorithm

In Theorist, a formula g can be predicted in $\mathcal{W}$ if and only if it is in every extension of $\mathcal{W}$. This is a statement about every scenario, whereas explaining is searching for exactly one scenario. Therefore, our aim of developing an algorithm for prediction is to reduce the problem of prediction to the use of explanations. We assume that we have an algorithm for explaining which returns all minimal explanations of a formula g within a world $\mathcal{W}$. An analogous idea is used in the original algorithm [4,8].

For the following discussion we need to clarify what statements like "explaining from a scenario", "predictable in a scenario", etc. mean. Given a scenario $\mathcal{F} \cup D$ of a world $\mathcal{W} = (\mathcal{F}, \Delta, \mathcal{C})$ we can create a new world $\mathcal{W}'$ by using the scenario as the new set of facts, i.e. $\mathcal{W}' = (\mathcal{F} \cup D, \Delta, \mathcal{C})$. Any definition stated in the previous section can be used with regard to this new world.

In the sequel, we describe an algorithm which determines whether a formula g is explainable from every scenario of a world $\mathcal{W}$. It is easy to verify that this is logically equivalent to g being in every extension of $\mathcal{W}$, i.e. g being predictable in $\mathcal{W}$.

(1) Try to find an arbitrary explanation of g from $\mathcal{W}$ using the available function for explaining. If there is no such explanation, the formula is not explainable from every scenario, as there is at least one trivial scenario, the set of facts itself (i.e. without using any default). In this case, return "no".

(2) If, on the other hand, there is an explanation $\mathcal{E}$ of the formula, then this explanation is a scenario of $\mathcal{W}$, so it is either compatible or incompatible with any other scenario. If there is no scenario incompatible with $\mathcal{E}$, the computed explanation may be seen as a universal explanation of g and therefore, return "yes".

(3) For every incompatible scenario there have to be other explanations of g if g is in every extension. To simplify this search, regard only minimal incompatible scenarios (with respect to set inclusion) and

show by a recursive call that the formula is explainable from every scenario of such a minimal scenario. Return "yes" if and only if the recursive call is successful for every incompatible minimal scenario.

The problem left is how to find all minimal scenarios which are incompatible with a given explanation $\mathcal{E}$. As the following section shows, every such scenario can be seen as a scenario which explains the negation of one element of $\mathcal{E}$ from a special world. The facts of this new world contain the original facts together with some subset of $\mathcal{E}$. The crucial point is that also every constraint in $\mathcal{W}$ must be regarded as a fact in the new world when searching for incompatibility instead of inconsistency. Using the function for explaining together with this new world provides all minimal incompatible scenarios.

One might object that taking constraints as facts and therefore using them for explanations contradicts the intention to use constraints for checking consistency only. However, the aim of using constraints for explaining is just to find incompatibilities.

In the following section, we justify every step towards the algorithm described above. Combining these justifications leads to a theorem our algorithm is based on.

4. The formal description

First of all, the idea to show that a formula is explainable from every scenario instead of proving membership in every extension has to be justified. The following lemma was originally stated and proven in [6].

Lemma 4.1. *A closed formula g is in every extension of $(\mathcal{F},\Delta,\mathcal{C})$ iff g is explainable from every scenario of $(\mathcal{F},\Delta,\mathcal{C})$.*

The next step towards the algorithm is to find an arbitrary explanation of g, and then to examine incompatible scenarios.

Lemma 4.2. *A closed formula g is explainable from every scenario of $(\mathcal{F},\Delta,\mathcal{C})$ iff there is some scenario $\mathcal{E}$ explaining g from $(\mathcal{F},\Delta,\mathcal{C})$ and for every scenario $\mathcal{S}$ of $(\mathcal{F},\Delta,\mathcal{C})$ which is incompatible with $\mathcal{E}$, g is explainable from $(\mathcal{S},\Delta,\mathcal{C})$.*

Proof. $(\Rightarrow)$ Suppose g is explainable from every scenario. Then,

(1) there is an explanation of g from $(\mathcal{F},\Delta,\mathcal{C})$, since g is explainable from $\mathcal{F} \cup \{\}$, and
(2) g is explainable from $(\mathcal{S},\Delta,\mathcal{C})$, as $\mathcal{S}$ is a scenario of $(\mathcal{F},\Delta,\mathcal{C})$.

($\Leftarrow$) Let $\mathcal{S}$ be a compatible scenario (with respect to $\mathcal{E}$). Then, $\mathcal{E}$ is a scenario of $(\mathcal{S},\Delta,\mathcal{C})$, as $\mathcal{E} \cup \mathcal{S} \cup \mathcal{C}$ is consistent. Therefore, g is explainable from $\mathcal{S}$ by using $\mathcal{E}$. □

Our intention is to try to prove that the formula to predict is in every extension of every *minimal* incompatible scenario instead of showing that it is explainable from every incompatible scenario. The following lemma justifies this.

Lemma 4.3. *Let $\mathcal{S}$ be a scenario of $(\mathcal{F},\Delta,\mathcal{C})$. Then, a closed formula g is explainable from every scenario $\mathcal{S}' \supseteq \mathcal{S}$ of $(\mathcal{F},\Delta,\mathcal{C})$ iff g is in every extension of $\mathcal{S}$.*

Proof. Since every scenario $\mathcal{S}' \supseteq \mathcal{S}$ of $(\mathcal{F},\Delta,\mathcal{C})$ is also a scenario of $(\mathcal{S},\Delta,\mathcal{C})$, the two statements are equivalent according to Lemma 4.1. □

At the heart of our algorithm is the use of explanations to find incompatible minimal scenarios. If two scenarios are incompatible, then their union and the set of constraints are inconsistent. Therefore, some part of this union along with the constraints can be seen as an explanation of some negated element of one of these scenarios.

Lemma 4.4. *Let $\mathcal{F} \cup D$ and $\mathcal{E} = \mathcal{F} \cup \{e_1,\dots,e_n\}$ be two scenarios of $(\mathcal{F},\Delta,\mathcal{C})$. Then, $\mathcal{F} \cup D$ is incompatible with $\mathcal{E}$ iff there is some $i \in \{1,\dots,n\}$ such that $\mathcal{F} \cup D \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C}$ is an explanation of $\neg e_i$ from $(\mathcal{F} \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C},\Delta,\emptyset)$.*

Proof. ($\Rightarrow$) Since $\mathcal{F} \cup D$ and $\mathcal{E}$ are incompatible, we have

$$\mathcal{F} \cup D \cup \{e_1,\dots,e_n\} \cup \mathcal{C} \models \bot .$$

Following the deduction theorem, there is some $i \in \{1,\dots,n\}$ such that

$$\mathcal{F} \cup D \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C} \models \neg e_i$$

and

$$\mathcal{F} \cup D \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C} \text{ is consistent.}$$

Using the definition of explanation, $\mathcal{F} \cup D \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C}$ can be seen as explaining $\neg e_i$ from $(\mathcal{F} \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C},\Delta,\emptyset)$.

($\Leftarrow$) Let $\mathcal{F} \cup D \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C}$ be an explanation of $\neg e_i$, i.e.

$$\mathcal{F} \cup D \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C} \models \neg e_i.$$

```
FUNCTION predict $(g,\ (\mathcal{F},\Delta,\mathcal{C}))$;
  $\mathcal{E} := explain\,(g,\ (\mathcal{F},\Delta,\mathcal{C}))$;
  IF $\mathcal{E}\ =\ \square$ THEN RETURN NO        "no explanation"
  ELSE
    FOR ALL $e_i \in \mathcal{E} = \{e_1,\dots,e_n\}$ DO
      FOR ALL $\mathcal{E}^* := explain\,(\neg e_i,\ (\mathcal{F} \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C},\Delta,\emptyset))$ DO
        LET $\mathcal{F} \cup D \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C}\ =\ \mathcal{E}^*$;
        IF $\neg predict\,(g,\ (\mathcal{F} \cup D,\Delta,\mathcal{C}))$ THEN
          RETURN NO
        END-IF
      END-FOR
    END-FOR;
    RETURN YES
  END-IF
END-FUNCTION
```

Fig. 1. This algorithm tries to determine whether or not a formula can be predicted in a Theorist world.

Then,

$$\mathcal{F} \cup D \cup \{e_1,\dots,e_i\} \cup \mathcal{C}\ \models \bot$$

and therefore

$$\mathcal{F} \cup D \cup \{e_1,\dots,e_n\} \cup \mathcal{C}\ \models \bot\,.$$

Hence, $\mathcal{F} \cup D$ and $\mathcal{E}$ are incompatible. $\square$

Combining the statements above leads to the following theorem.

Theorem 4.5. *A closed formula g is in every extension of $(\mathcal{F},\Delta,\mathcal{C})$ iff there is some explanation $\mathcal{F} \cup \{e_1,\dots,e_n\}$ explaining g from $(\mathcal{F},\Delta,\mathcal{C})$ such that for all $i \in \{1,\dots,n\}$ and every minimal $\mathcal{F} \cup D \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C}$ explaining $\neg e_i$ from $(\mathcal{F} \cup \{e_1,\dots,e_{i-1}\} \cup \mathcal{C},\Delta,\emptyset)$, g is in every extension of $(\mathcal{F} \cup D,\Delta,\mathcal{C})$.*

If we assume that we have a function *explain* $(g,\mathcal{W})$ which generates all minimal explanations of g from $\mathcal{W}$, then the algorithm extracted from this theorem is shown in Fig. 1.

5. Discussion

We have presented an algorithm for implementing skeptical reasoning in Theorist. We have defined the notion of incompatibility as a principle underlying prediction. The notion of incompatibility is related to the notion of *commitment to assumptions* in default logic. Default logic as defined in [9] lacks commitment as pointed out in [7]. Since commitment seems to be an intuitively indispensable property of nonmonotonic reasoning, recently a number of variants of default logic have been developed (cf. [3]). Some of these variants turned out to be closely related to Theorist with constraints. For example, a transformation from *semi-normal prerequisite-free default theories* to Theorist worlds including constraints is given in [3], and it is shown that every so-called *constrained extension* corresponds to a Theorist extension. Because of this transformation, we can use our algorithm for skeptical reasoning in default logics such as [1,2].

Moreover, we believe that our algorithm can be generally modified to implement skeptical reasoning within default logics which consider commitment. In [9], an algorithm solving the extension membership problem for closed normal default theories is given. This algorithm generates *default proofs*, which resembles generating explanations in Theorist. We think that combining default proof algorithms with the result presented in this paper would straightforwardly lead to a procedure for skeptical reasoning in many variants of default logic.

6. Conclusion

We have developed an algorithm for prediction within the entire Theorist framework and we have shown that this algorithm is correct.

In addition, we have implemented our algorithm within the current implementation of Theorist which is written in Prolog [4]. The system now gives correct answers in any case using prediction together with constraints.

The extended code is available from the author and described in [10].

Acknowledgement

I am indebted to Wolfgang Bibel for his support and helpful discussions during the preparation of this work. I am also grateful to Norbert Hanf and Torsten Schaub for valuable remarks and to Christoph Kreitz and an anonymous referee for their constructive criticism on an earlier version of this paper.

References

[1] G. Brewka, Cumulative Default Logic: in defense of nonmonotonic inference rules, *Artif. Intell.* **50** (1991) 183–205.

[2] J.P. Delgrande and W.K. Jackson, Default logic revisited, in: *Proceedings Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA (1991) 118–127.

[3] J.P. Delgrande, W.K. Jackson and T. Schaub, Alternative approaches to default logic, in: *Proceedings ECAI-92*, Vienna, Austria (1992).

[4] D. Poole, A Theorist to Prolog compiler, Logic Programming and AI Group, Department of Computer Science, University of Waterloo, Ont. (1987).

[5] D. Poole, A logical framework for default reasoning, *Artif. Intell.* **36** (1) (1988) 27–47.

[6] D. Poole, Explanation and prediction: an architecture for default and abductive reasoning, *Comput. Intell.* **5** (2) (1989) 97–110.

[7] D. Poole, What the Lottery Paradox tells us about default reasoning, in: *Proceedings First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Ont. (1989) 333–340.

[8] D. Poole, Compiling a default reasoning system into Prolog, *New Gen. Comput.* **9** (1) (1991) 3–38.

[9] R. Reiter, A logic for default reasoning, *Artif. Intell.* **13** (1980) 81–132.

[10] M. Thielscher, Prognostizieren im Poole'schen Ansatz—Arbeiten mit Theorist, Tasso Rept. 18, Technische Hochschule Darmstadt, Germany (1991).