The Persistence of Derived Information

Karen L. Myers **David E. Smith**
Department of Computer Science
Stanford University
Stanford, California 94305

Abstract

Work on the problem of reasoning about change has focussed on the persistence of nonderived information, while neglecting the effects of inference within individual states. In this paper, we illustrate how such inferences add a new dimension of complexity to reasoning about change and show that failure to allow for such inferences can result in an unwarranted loss of derived information.

The difficulties arise with a class of deductions having the property that their conclusions should be allowed to persist even though some components of the justifications involved may no longer be valid. We describe this notion of components of a justification being *inessential* to the persistence of that justification. A solution to the persistence problem is presented in terms of a default frame axiom that is sensitive to both justification information and specifications of inessentiality.

1 Introduction

The ability to reason about change is essential for intelligent systems that must interact with the real world. Recently there have been a number of nonmonotonic schemes designed to perform this task [Ginsberg, 1986; Ginsberg and Smith, 1987; Haugh, 1987; Lifschitz, 1987; Shoham, 1986]. Unfortunately, none of these approaches can properly deal with the persistence of information derived *within* a given state.

The frame axioms that these systems employ are too powerful to be applied to inferred facts. The underlying criteria they use in determining which facts persist is the consistency of such facts with the next state. As will be shown, this makes the application of these axioms to inferred facts unsuitable. However, restricting their application to nonderived information can result in the unwarranted disappearance of derived information, as there exist inferred facts that should be retained across state transitions even though the derivations used to justify these facts are no longer valid in the new state.

The principal objective of this paper is to outline the complexities inherent to controlling the persistence of derived information. After presenting a series of examples in Section 2 that illustrate the subtleties of the problem, we introduce the notion of *inessential components* of a justification in Section 3. This concept is fundamental to understanding the persistence problem. A solution in terms of a default frame axiom is then proposed in Section 4. Our axiom handles derived information properly by considering not only which facts hold in a given state, but also why they hold and whether the justifications involved contain inessential components.

2 The nature of the problem

Consider the simple frame axiom schema

$$\frac{p_t : p_{t+1}}{p_{t+1}} \tag{1}$$

expressed as a default rule of Reiter [Reiter, 1980]. The notation p_t represents the fact that fluent p is true in state t.[1] Informally, the default says that facts persist across state transitions provided that they are consistent with information about the new state. For simplicity, we assume discrete time in this presentation.

Unfortunately, unrestricted application of this default can yield absurd results.

Example 1 — The Green Cheese Problem
Let t be a state in which some fluent A is true; that is A_t holds. Now suppose we wish to perform an action that makes A false, resulting in $\neg A_{t+1}$. From A_t, it follows that $(A \vee B)_t$. As $\neg A$ is the only nontrivial fluent known to hold in state $t+1$ and $A \vee B$ is consistent with $\neg A$, the default (1) allows the propagation of $A \vee B$ through to state $t+1$. But then we have both $\neg A_{t+1}$ and $(A \vee B)_{t+1}$, and so B_{t+1} is derivable. This is certainly an anomalous situation since B could be any sentence, such as 'The moon is made of green cheese'.

The nature of the Green Cheese problem seems clear. The reason that $A \vee B$ does not belong in the new state is that it depended on A for justification in state t. Retracting A should further result in the retraction of $A \vee B$. By restricting the application of (1) to *base* facts (*i.e.* nonderived facts), such unsupported information will not be retained. Logical consequences of the base facts would be rederived upon each state transition. For efficiency reasons, justification information [Doyle, 1979] could be maintained to avoid the overhead of recomputing derivations that remain valid across states.

This solution is overly conservative. As the following scenario demonstrates, there are cases where derived information should persist even though the initial justifications are no longer valid.

[1] The formula p_t is an abbreviation for the formula *HOLDS(p,t)* commonly found in the literature. This nonstandard notation is used in order to reduce the unwieldiness of formulas presented below.

Example 2 — The Displaced Cup

Consider a domain in which there exists a robot capable of picking up certain objects. At some point in time, it is known that the set of objects currently resting on a nearby table are all sufficiently lightweight that the robot is able to lift them. In particular, there exists a small cup located there. More formally, the implication

$$\forall x.\ OnTable(x)_{t_0} \supset Liftable(x)_{t_0} \quad (2)$$

as well as the fact

$$OnTable(Cup)_{t_0} \quad (3)$$

both hold. It is easy to see that $Liftable(Cup)_{t_0}$ logically follows. Now suppose the robot picks up *Cup* and then proceeds to set it down on the floor. As a consequence of this action, $OnFloor(Cup)_{t_0+1}$ is obtained. Assuming some sort of domain constraint that prevents objects from being at more than one place at any given point in time, *OnTable(Cup)* is not consistent with the new state and so the frame default (1) will not allow it to persist. If we restrict application of the frame axiom to base facts, *Liftable(Cup)* will not be propagated to the new time point via the frame default since it is a derived rather than base fact. As the justification for *Liftable(Cup)* used in state t_0 is no longer valid, we have no basis for believing *Cup* to be liftable after it is moved to the floor.

Nothing seems amiss in this scenario at first glance. However, if one considers the semantics of the predicates involved rather than the purely syntactic manipulations of the reasoning process, it seems unreasonable to lose *Liftable(Cup)* as a result of the robot having moved *Cup* to the floor. The robot's capacity for lifting *Cup* should be independent of changes in *Cup*'s location.

If the retraction of *OnTable(Cup)* had been made as a correction to some erroneously perceived information, then the subsequent removal of *Liftable(Cup)* would only be natural. The fact that we are *changing* situations due to some event in the world alters the nature of the retractions that should be made. The instance of *Liftable* in the above scenario is an example of a fluent that, once established at some point in time becomes self-justified (subject to consistency constraints) for subsequent states. The removal of its initial justification as a result of changes in the world should not be sufficient grounds for its retraction.

In general, this is not the case for all fluents. Suppose our domain theory also includes the axiom

$$\forall xt.\ OnTable(x)_t \supset SafeFromBaby(x)_t.$$

That is, objects on the table cannot be reached by a child (who is presumably crawling on the floor). Then any state in which *OnTable(Cup)* holds would also have *SafeFromBaby(Cup)* holding. This latter fact should clearly be removed when *Cup* is moved to the floor. Unlike *Liftable(Cup)*, the justification of *SafeFromBaby(x)* requires the continued validity of *OnTable(Cup)*.

The problem clearly lies with the axiomatization of the domain. The implication (2) is sufficient for characterizing the relationship between *OnTable* and *Liftable* *within* a given state, but lacks any information about the relationship between these fluents *across* states. One would hope to solve the problem by simply modifying the axioms.

Rewriting (2) as

$$\forall x.\ OnTable(x)_{t_0} \supset \forall t' > t_0.\ Liftable(x)_{t'}$$

won't suffice. There is nothing to prevent *Cup* from becoming unliftable in the future. For example, filling *Cup* with tea may cause the combined weight of *Cup* and its contents to exceed the robot's threshold for liftability. Rather, some sort of default mechanism is necessary that will allow *Liftable* to persist as long as is consistently possible.

It may seem that what is needed is to allow the frame default (1) to be applied to certain fluents in the theory, such as *Liftable*, even when they appear as derived information. Such applications of (1) can cause the Green Cheese problem to resurface however, as is readily seen in the following example.

Example 3 — Filling the Cup

Suppose we add the formulas

$$\forall t.\ Empty(x)_t \supset Liftable(x)_t$$
$$Empty(Cup)_{t_1}$$

to our example. Then $Liftable(Cup)_{t_1}$ will also hold as it is a logical consequence of these two facts. If the robot decides to fill *Cup* with tea in state t_1, then $\neg Empty(Cup)_{t_1+1}$ will hold. Application of (1) to $Liftable(Cup)_{t_1}$ generates $Liftable(Cup)_{t_1+1}$. This is not what one would desire since *Empty* is essential to the continued belief of this justification for *Liftable*.

Adding the axiom

$$\forall t.\ \neg Empty(x)_t \supset \neg Liftable(Cup)_t \quad (4)$$

would be sufficient to block this extraneous persistence of *Liftable(Cup)*. However, one can certainly envision domains in which this axiom simply is not true. Asserting (4) will have the side-effect of altering the character of the domain theory.

Thus simply expanding the range of facts to which (1) can be applied is not the solution. We need to apply the frame default to certain instances of derived information, *depending on the nature of the derivations involved.* Incorporating justification information into the frame axiom is necessary for making this differentiation. In addition, characterizations of the essential and inessential components of each justification must be specified.

It is important to note that the problems outlined above arise in all current nonmonotonic systems for reasoning about change, not just the simple default framework employed here. In general, any system using a nonmonotonic frame axiom that is not sensitive to justification information will be incapable of dealing correctly with the persistence of derived information.

3 Inessentiality

As was illustrated above, there are two modes in which a particular fluent can support the derivation of another. On one hand, a fluent may be required for the continued justification of a related fluent, as was the case with the relationship of *Empty* to *Liftable*; on the other hand, a fluent may be used initially to establish the truth of another fluent, but not be necessary for the persistence of that second fluent. This was the relationship between *OnTable* and *Liftable*. In terms of notation, we will say that *Empty* is *essential* to the persistence of *Liftable* while *OnTable* is *inessential*.

The question arises as to what makes a particular fluent inessential to another fluent within a particular domain theory. The intuition behind inessentiality is related to the notion of causality. When should the retraction of q not bring about the retraction of p? This latter retraction should be blocked whenever there is no causal explanation of $\neg q$ that could also account for $\neg p$. In other words, no actions known to bring about $\neg q$ also bring about $\neg p$.

Returning to our examples from the previous section, we see that this explanation is in accord with our intuitions. Actions that relocate an object should in no way affect the robot's ability to lift them.[2] However, there are numerous situations in which filling an object could increase its mass to the point where it becomes unliftable.

Given the notion of inessentiality, it remains to characterize the type of situations in which inessentiality relationships exist. We have catalogued the following three classes.

Incidental Inessentiality This class is typified by Example 2, where *OnTable* is inessential to *Liftable*. In this case, the validity of *OnTable* is incidental to the validity of *Liftable* in that there is no causal link between the two fluents. It is merely a coincidence that the given logical relationship holds.

Causal Inessentiality Suppose that our robot is known to be stronger than some individual Fred. Then our domain theory might include an axiom such as

$$\forall xt.\ \mathit{Fred\_Holds}(x)_t \supset \mathit{Liftable}(x)_t.$$

If Fred is holding *Cup* at some point in time, then we are able to conclude that *Cup* is liftable. The liftability of *Cup* should not be affected by Fred setting it down; thus *Fred_Holds* is inessential to *Liftable*. In contrast to instances of the previous class, here we have a causal relationship underlying the given axiom. The motivation for the axiom itself is the existence of a common cause for the two fluents.

Definitional Inessentiality The third class consists of definitions stated in terms of logical equivalences. Consider the axiom

$$\forall t.\ \mathit{Working}(\mathit{HalfAdder1})_t \equiv \mathit{Working}(\mathit{XOR1})_t \wedge \mathit{Working}(\mathit{AND1})_t.$$

This formula describes the conditions under which half-adder *HalfAdder1* is working correctly, namely that its two gates (*XOR1* and *AND1*) are functional. Suppose that in a particular state we know that *Working(HalfAdder1)* holds, from which it follows that both *Working(XOR1)* and *Working(AND1)* hold. If in some future state a malfunction occurs in gate *AND1*, then *Working(AND1)* will no longer hold and so neither will *Working(HalfAdder1)*. As this latter fact was our original justification for *Working(XOR1)*, there will no longer be grounds for believing that *XOR1* still works. This is clearly unreasonable as a fault in one gate should not affect the integrity of the other. In this case, *Working(HalfAdder1)* is inessential to both *Working(AND1)* and *Working(XOR1)*.

These three classes are not meant to be exhaustive. It should be clear from their descriptions however, that the persistence problem for derived information is indeed significant.

[2] This statement is not completely accurate — what if the object is moved onto a surface covered with glue? We return to this point in the closing remarks of the paper.

4 A New Frame Axiom

Solving the persistence problem for derived information requires the development of some mechanism by which instances of inessentiality can be specified and enforced. In this section, we proceed to develop a solution in terms of a default frame axiom that takes into account both justification and inessentiality information.

Consider our sequence of robot examples once again. We need to construct a default frame axiom that applies to instances of *Liftable* derived from the axiom $\forall x.\ \mathit{OnTable}(x)_{t_0} \supset \mathit{Liftable}(x)_{t_0}$, while excluding instances derived from $\forall xt.\ \mathit{Empty}(x)_t \supset \mathit{Liftable}(x)_t$.

What does it mean for an instance of *Liftable(x)* to be justified by the formula $\forall x.\ \mathit{OnTable}(x)_{t_0} \supset \mathit{Liftable}(x)_{t_0}$ in a particular state t? Clearly $\mathit{OnTable}(x)$ must have held in state t_0. Further, it is necessary that $\mathit{Liftable}(x)$ persisted in all intervening states from t_0 through t. This restriction ensures that the justification cited in t_0 is still the reason for $\mathit{Liftable}(x)$ holding in the current state. If there existed an intermediate state in which $\mathit{Liftable}(x)$ did not hold, then some other justification would be responsible for the rederivation of $\mathit{Liftable}(x)$ at some later point. This justification could assume the form of either a different deduction or the direct effect of an action.

Using the notation introduced above, these intuitions translate into the rule

$$\frac{\left(\begin{array}{l}\mathit{OnTable}(x)_{t_0}\\ \quad \wedge\ \forall t'\ t_0 \leq t' \leq t.\ \ \mathit{Liftable}(x)_{t'}\end{array}\right) : \mathit{Liftable}(x)_{t+1}}{\mathit{Liftable}(x)_{t+1}}. \tag{5}$$

The conjunct $\mathit{OnTable}(x)_{t_0}$ in the precondition of the default ensures that $\mathit{Liftable}(x)_t$ was indeed established using the axiom $\forall x.\ \mathit{OnTable}(x)_{t_0} \supset \mathit{Liftable}(x)_{t_0}$; the conjunct $\forall t'\ t_0 \leq t' \leq t.\ \mathit{Liftable}(x)_{t'}$ guarantees that $\mathit{Liftable}(x)$ continued to persist for the same reason from t_0 through the current state t.

Now suppose we complicate our example somewhat by modifying the conditions for liftability to include both essential and inessential components in the antecedent of the implication.

Example 4 — Slippery Cups

Assuming that the robot can only lift dry objects, we rewrite (2) as

$$\forall x.\ \mathit{OnTable}(x)_{t_0} \wedge \mathit{Dry}(x)_{t_0} \supset \mathit{Liftable}(x)_{t_0}.$$

In the spirit of (5), we might postulate

$$\frac{\left(\begin{array}{l} OnTable(x)_{t_0} \wedge Dry(x)_{t_0} \\ \quad \wedge \forall t' \; t_0 \leq t' \leq t. \;\; Liftable(x)_{t'} \end{array}\right) : Liftable(x)_{t+1}}{Liftable(x)_{t+1}}$$

as the appropriate default. This is not satisfactory however, as it fails to capture the indispensability of $Dry(x)$ to the persistence of $Liftable(x)$. It is necessary to demand that $Dry(x)$ hold at each state from t_0 through the new situation. Allowing for this further condition, we have the rule

$$\frac{\left(\begin{array}{l} OnTable(x)_{t_0} \wedge Dry(x)_{t_0} \\ \quad \wedge \forall t' \; t_0 \leq t' \leq t. \;\; Liftable(x)_{t'} \\ \quad \wedge \forall t' \; t_0 \leq t' \leq t+1. \;\; Dry(x)_{t'} \end{array}\right) : Liftable(x)_{t+1}}{Liftable(x)_{t+1}}. \tag{6}$$

The defaults (5) and (6) are sufficient to ensure the desired persistence properties for the given examples. However, it is not practical to explicitly write out default rules such as these for every derivation containing inessential components. In general, there will simply be too many of these defaults. A better approach would be to formulate a general-purpose default schema that subsumes these highly specific rules. Such a schema would have additional benefits. Not only would it allow domain-specific information to be confined to an initial theory rather than being dispersed throughout a series of default rules, but it would also provide a perspicuous encapsulation of the persistence policy in effect.

One of the problems with the two given defaults is that they implicitly embody information about inessentiality. In particular, components of justifications that are inessential to the conclusion are not required to persist in order for the justification as a whole to remain valid. In constructing a general-purpose frame default, this information must be distilled from the rules and stated explicitly within the domain theory itself. To this end, we introduce the predicate *INESSENTIAL*. Intuitively, $INESSENTIAL(p,q)$ represents that fluent q is inessential in any justification of fluent p.

The premise behind the uniform frame axiom is simple. Once a fluent is established in some particular state, it should continue to persist provided that the essential components of its derivation remain true and this persistence will not produce an inconsistency. Note that in order to evaluate such conditions, it is necessary to know not only whether or not a fluent is true, but also *why* it is true. Thus justification information has been elevated from the status of an efficiency mechanism to being an integral part of the reasoning process.

As it is necessary to reason about justification information, our formalization requires the introduction of the predicate *JUSTIFIES*, where *JUSTIFIES(J,p,t)* indicates that J is a minimal set of fluents that is sufficient for deriving p in state t. The validity of these fluents in state t must be a consequence of the domain theory combined with those fluents that have been directly posited within state t (either as a result of the frame axiom or as the direct effect of an action). For example, in the scenario where *Cup* is known to be on the table in state t_0 we have

$$JUSTIFIES(J, Liftable(Cup), t_0)$$

where

$$J = \{OnTable(Cup) \supset Liftable(Cup), OnTable(Cup)\}.$$

Using this machinery, the uniform frame axiom can be expressed as

$$\frac{PRECOND(p,t) : p_{t+1}}{p_{t+1}} \tag{7}$$

where the predicate *PRECOND(p,t)* is defined by

$$\begin{array}{l} \exists t_0 \; \exists J. \; JUSTIFIES(J, p, t_0) \\ \quad \wedge \forall t' \; t_0 \leq t' \leq t. \;\; p_{t'} \\ \quad \wedge \forall t' \; t_0 \leq t' \leq t+1. \\ \qquad\qquad \forall j \in J. \; (j_{t'} \vee INESSENTIAL(p,j)). \end{array} \tag{8}$$

Intuitively, the formula (8) ensures that there exists a derivation of p from some set J of fluents such that three basic conditions are met. The first condition is that this derivation holds at some earlier state t_0. Secondly, for each state between t_0 and the current state t, the derived fluent p is true. The third condition is that from state t_0 through the new state $t+1$, each element of J is either true or is inessential to p.

5 Overlapping Derivations

The definition of *PRECOND* stated in the previous section is not quite complete. As it stands, the definition is inappropriate for use when concurrent derivations of a given fact exist, some of which rely on inessential information.

Consider the axioms for liftability once again:

$$\begin{array}{c} \forall x. \; OnTable(x)_{t_0} \supset Liftable(x)_{t_0} \\ \forall t. \; Empty(x)_t \supset Liftable(x)_t \; . \end{array}$$

If both *OnTable(Cup)* and *Empty(Cup)* hold in state t_0, there are two separate derivations of *Liftable(Cup)*. However, the distinction between the two may be purely superficial. It is certainly plausible that the underlying reason for the validity of $\forall x. \; OnTable(x)_{t_0} \supset Liftable(x)_{t_0}$ is simply that all objects on the table in state t_0 are empty; in a semantic sense, the two derivations overlap. Should the derivation of *Liftable(Cup)* from *Empty(Cup)* become invalid in some future state due to the retraction of *Empty(Cup)*, one would further expect the persistence of *Liftable(Cup)* based on *OnTable(Cup)* to terminate.

In more general terms, let p be a fluent that is initially established in state t by a derivation containing at least one fluent that is inessential to p. If there exists any other derivation of p in state t, then the persistence of p stemming from the first justification should be blocked at any point where one of these simultaneous justifications becomes invalid.

Note that the simultaneous justifications may or may not contain inessential components. Two justifications containing inessential components can easily overlap each other in the same manner as the *Ontable* derivation overlapped the essential *Empty* justification in the example above. Further, it should be pointed out that we are adopting a conservative stance with respect to overlapping derivations. In particular, persistences based on derivations with inessential components that *potentially* overlap other derivations are treated as though the overlap actually exists.

Compensating for such potential overlaps requires redefining *PRECOND* as

$$\begin{aligned}&\exists t_0\ \exists J.\ JUSTIFIES(J,p,t_0)\\ &\quad\wedge \forall t'\ t_0 \leq t' \leq t.\ \ p_{t'}\\ &\quad\wedge \forall t'\ t_0 \leq t' \leq t+1\ \ \forall J'\ JUSTIFIES(J',p,t_0)\\ &\qquad\qquad\qquad \forall j \in J'.\ \ (j_{t'} \vee INESSENTIAL(p,j)).\end{aligned} \tag{9}$$

Here the third condition in the original definition of *PRECOND* has been modified to reflect the fact that *all* justifications of p that hold in state t_0 must have their essential components remain valid, not simply the justification represented by J.

6 Concluding Remarks

The question arises as to whether or not representing inessentiality as a binary relation on formulas is epistemologically adequate.

One can certainly envision the need for introducing *conditional* inessentiality. Extending the formalism given above to accommodate this generalization is straightforward. A more significant problem is ensuring that some analogue of the *qualification problem* [McCarthy, 1977] does not exist.

Given our definition of inessentiality, it seems that the approach given here is safe provided that an adequate representation of actions is given. This belief is based on the fact that the specification of inessentiality relationships only leads to the persistence of information *by default.* Should some unanticipated situation arise in which the indirect effect of an action conflicts with a persistence prescribed by inessentiality specifications, the information defining this effect would be sufficient to block the default persistence. Returning to our robot scenario, should *Cup* be moved onto a floor covered with glue, the presence of the glue together with axioms describing the immobility of glued objects would block the default persistence of *Cup*'s liftability.

It is interesting to note that implementing the default frame axiom (9) is fairly straightforward in a system equipped with reason maintenance information [Doyle, 1979]. Details of the algorithm are left to another paper, but the fundamental idea is to alter the maintenance mechanism to include contextual information about the recursive sequence of retractions that has initiated the current retraction. Before performing any retraction, a check would be made on the current context. If any fact in this context is inessential to the fact being retracted, then the retraction is simply not carried out. The overhead of this modification is quite small and will not significantly affect the performance of the system.

Acknowledgements

The authors would like to thank Don Geddis and Peter Ladkin for pointing out the possibility of overlapping derivations. Matthew Ginsberg, Nils Nilsson and Eunok Paek also provided many useful comments.

The work of the first author has been supported by DARPA and NASA under grant NCC2-494 while that of the second author has been supported by DARPA under grant N00039-86-C-0033 and ONR under grant N00014-81-K-0004.

References

[Doyle, 1979] Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[Ginsberg and Smith, 1987] Matthew L. Ginsberg and David E. Smith. Reasoning about action I: A possible worlds approach. In Matthew L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning.* Morgan Kauffman, Los Altos, CA, 1987. To appear in *Artificial Intelligence.*

[Ginsberg, 1986] Matthew L. Ginsberg. Possible worlds planning. In *Proceedings of the 1986 Workshop on Planning and Reasoning about Action*, pages 213–243, Timberline, Oregon, 1986. Morgan Kaufmann.

[Haugh, 1987] Brian Haugh. Simple causal minimizations for temporal persistance and projection. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 218–223, 1987.

[Lifschitz, 1987] Vladimir Lifschitz. Formal theories of action. In *Proceedings of the 1987 Workshop on the Frame Problem in Artificial Intelligence*, Lawrence, Kansas, 1987.

[McCarthy, 1977] John McCarthy. Epistemological problems of artificial intelligence. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 1038–1044, Cambridge, MA, 1977.

[Reiter, 1980] Ray Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[Shoham, 1986] Yoav Shoham. Chronological ignorance. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 389–393, 1986.