Lecture Notes 11 – Solutions to the Frame Problem(s) II

Patrick Doherty

February 7, 1998

1 Outline

1. Causal Minimization
   (a) Lifschitz's Work
   (b) Haugh's Work

2 Lifschitz's Logic of Action

Lifschitz argues that there is a better way than the previous approaches using chronological minimization for solving the temporal projection problem. In fact, he claims this solves the frame problem, but many would disagree. In any event, the causal approach proposed by Lifschitz is worth considering. Key to the approach is the use of a new ontology for representing scenario descriptions, where both causes of changes to fluent values and preconditions to action descriptions are made explicit in terms of two new predicate constants $Causes$ and $Preconds$. In the new formalism, the intended models of a scenario description are characterized by the fact that *all changes in the values of fluents are caused by actions*, whereas in other models, this is not necessarily the case. In his approach, Lifschitz deals with both the frame and qualification problems.

Another good reason for considering Lifschitz's approach is the article itself, which is extremely well-written and provides one of the better descriptions of the use of the situation calculus, the unique names and domain closure assumptions in the literature. Lifschitz first solves the YSP using term models and then generalizes the solution to arbitrary models. We will only consider the formalism in the context of term models, but it is useful to try and understand the rest of the article.

2.1 Example1

We will use the following sorts: actions (a), situations (s), truth-valued fluents (f), and truth values (v).

Our vocabulary, $\Lambda = \langle C, F, R, d \rangle$, contains the following:

1. $C = C_s \cup C_a \cup C_f \cup C_v$ where

$$C_s = \{S0\} \text{ and } \forall x \in C_s . d(x) = s,$$

$$C_a = \{load, wait, shoot\} \text{ and } \forall x \in C_a . d(x) = a,$$

$$C_f = \{loaded, alive\} \text{ and } \forall x \in C_f . d(x) = f,$$

$$C_v = \{true, false\} \text{ and } \forall x \in C_v . d(x) = v.$$

2. $F = \{result\}$ where
$$d(result) = \langle a \times s \rightarrow s \rangle.$$

3. $R = \{Holds, Causes, Precond, Success, Affects\}$ where[1]
$$d(Holds) = \langle f \times s \rangle,$$
$$d(Causes) = \langle a \times f \times v \rangle,$$
$$d(Precond) = \langle f \times a \rangle,$$
$$d(Success) = \langle a \times s \rangle,$$
$$d(Affects) = \langle a \times f \times s \rangle,$$

The list of variables for sorts is:

$$s, s_1, s_2, s_3, \ldots, a, a_1, a_2, a_3, \ldots \text{etc.}$$

1. $Causes(a, f, v)$ expresses that the action a causes the fluent f to take on the value v when a is successfully performed.
2. $Precond(f, a)$ expresses that the fluent f is a precondition for the successful execution of a. $Precond$ is used to deal with the qualification problem.

2.1.1 The YSP Axiomatization

The following axiomatization is provided for the YSP scenario. The first group of axioms describes the initial situation $S0$:

$$Holds(alive, S0) \tag{1}$$
$$\neg Holds(loaded, S0) \tag{2}$$

The second group of axioms describes how actions effect fluent values:

$$Causes(load, loaded, true) \tag{3}$$
$$Causes(shoot, loaded, false) \tag{4}$$
$$Causes(shoot, alive, false) \tag{5}$$

To deal with qualification, the notion of a successful action in a situation is represented as

$$Success(a, s) \equiv \forall f.[Precond(f, a) \supset Holds(f, s)]$$

where $Precond$ represents the explicit qualifications known about the action. The following action qualification is used in the YSP,

$$Precond(loaded, shoot). \tag{6}$$

The next axiom describes whether a fluent changes value to $true$ or $false$ given a successful action that influences the fluent:

$$Success(a, s) \wedge Causes(a, f, v) \supset [Holds(f, result(a, s)) \equiv v = true]. \tag{7}$$

[1] Both $Success$ and $Affects$ are defined predicate constants and not primitive.

A fluent is affected if there is a successful action and the action can cause a change to the fluent:

$$Affects(a, f, s) \equiv Success(a, s) \wedge \exists v. Causes(a, f, v)$$

The following axiom represents the inertia assumption. If the fluent is not affected by an action then its value remains the same,

$$\neg Affects(a, f, s) \supset [Holds(f, result(a, s)) \equiv Holds(f, s)]. \quad (8)$$

Term models are used for this example to simplify the axiomatization. We will discuss this in another section. For the time being, think Herbrand models.

2.1.2 The Circumscription Policy

Let Γ be the axiom set described above, excluding the definitions for $Success$ and $Affects$. The circumscription policy proposed by Lifschitz is to minimize both $Causes$ and $Preconds$ and allow $Holds$ to vary. Since both $Success$ and $Affects$ are defined in terms of $Causes$ and $Preconds$, they are not included in the policy. Using the notation from the lecture that described parallel predicate circumscription, the following policy is suggested

$$Circ(\Gamma; \{Causes, Precond\}; \{Holds\})$$

Applying circumscription results in one minimal term model, M_0. Given that

$$S3 = result(shoot, result(wait, result(load, S0))),$$

It is not the case that the turkey is alive after loading, waiting and then shooting the gun.

$$Circ(\Gamma; \{Causes, Precond\}; \{Holds\}) \not\models Holds(alive, S3)$$

as desired.

Let's take a closer look at the axiomatization to see why this approach works for the YSP. Recall that the original problem was that global minimization of the Ab predicate resulted in two preferred models due to a conflict in minimizing at earlier and later instants of time. The chronological minimization approach solved this particular problem by incorporating a more *local* minimization policy, where one could express preference for minimizing at earlier points in time. So any indirect influences of the minimization at a point would be pushed forward chronologically. Minimizing at later points in time would rule out any indirect influences at chronologically earlier points. This approach to the YSP provides a solution at the expense of introducing a more complicated form of circumscription.

In the present approach proposed by Lifschitz, both $Cause$ and $Preconds$ are minimized globally, dispensing with the need to introduce a new form of circumscription. The trick is that in changing the ontology, the predicates minimized are decoupled from dependence on a temporal context. Observe that there are no "situation" arguments to $Cause$ and $Preconds$. Minimizing globally will not introduce conflict between earlier and later timepoints. In addition, the connection to temporal context is made indirectly via the definitions of $Success$ and $Affects$ which do include "situation" arguments. Perhaps minimization of $Cause$ and $Preconds$ can still lead to problems via some atemporal indirect affect due to the nature of the axioms. Careful observation shows that this can not happen either due to the fact that all uses of $Cause$ and $Preconds$ are as atomic formulas.

More specifically, minimizing $Causes$ indirectly maximizes the negative instances of $Affects$ via its definition in terms of $Causes$. By axiom (8), maximizing $\neg Affects$ also maximizes persistence. This is the proposal for solving the frame problem; the only changes to fluent values are those

which are explicitly caused by an action. Since cause is minimized, so is change, which means that persistence is maximized.

Minimizing $Preconds$ does two things. It maximizes $Success$ via the definition for success, where if a qualification is not explicitly represented as a precondition to an action, it is assumed not to be a precondition and an instance of the r.h.s of the definition for success is made true. Secondly, minimization of $Preconds$ assures that those qualifications explicitly mentioned must hold if success of the action is to be asserted. If the precondition is mentioned and it holds at the situation in question, then the instance of the r.h.s. of the definition is true. If not then the instance is false. Together, maximization of success assures the effect of the action via axiom (7).

In summary, the key intuition as to why this solution works for the YSP and earlier uses of global circumscription don't is summarized quite nicely by Lifschitz, where he states that

> "The main technical difference between this formulation and other attempts to apply circumscription to the situation calculus is that the predicates which we circumscribe, *causes* and *precond*, are not fluents, they do not have situation arguments. The predicate *success*, which does have a situation argument, is defined in terms of the minimized primitive *precond*. Since the minimized predicates have no situation arguments, the conflict between minimizing in earlier and later instants of time simply cannot arise." (p. 416, [**?**])

One other point: since we are mainly dealing with the frame problem and not the qualification problem, one can easily separate the two parts of the axiomatization from each other. This is achieved in a relatively straightforward manner by disregarding the definition of $success$, modifying the definition of $affects$ to be

$$affects(a, f, s) \equiv \exists v causes(a, f, v)$$

replacing axiom (7) with

$$\forall a \forall f \forall v.causes(a, f, v) \supset (holds(f, result(a, s)) \equiv v = true), \tag{9}$$

and substituting the new definition for $affects$ into axiom (8). With some minor manipulation, the following applies

$$\forall a \forall f \forall v \neg causes(a, f, v) \supset (holds(f, result(a, s)) \equiv holds(f, s)). \tag{10}$$

The modified circumscriptive policy would be to minimize *cause* and let *holds* vary.

2.1.3 Computing the Preferred Models

Although the solution provides a syntactic characterization in terms of the circumscription policy, in the spirit of Shoham and also to assure that we are convinced that there is only one preferred model, this can be generated in a straightforward way. This is due to the fact that we are using term models and that there is an initial situation from which to start and recursively generate the tuples in the extension of $Holds$ at each situation label. Recall the similarity with Shoham's direct approach to generating base sentences true in all c.m.i. preferred models where he applies the algorithm moving forward in time.

Since we are dealing only with term models, the universe of any such model will contain only the ground terms that can be constructed from the signature Λ associated with the axiomatization Γ. The ground terms include,

1. Constants:
   (a) The situation: $S0$.
   (b) The truth values: $true$ and $false$.

(c) The actions: *load*, *wait*, and *shoot*.

(d) The fluents: *loaded* and *alive*.

2. Functional terms:

   (a) Besides $S0$, infinitely many more situations constructed recursively from SO, the action constants and the *results* function.

Any model will be defined as the interpretations of the $Holds$, $Causes$, and $Preconds$ predicates, relative to this universe. Since this is equivalent to providing a truth mapping from the ground atoms to true and false, Lifschitz presents a boolean function $\mathcal{T}$ on the set of ground atoms via the following recursive equations:

$$\begin{aligned}\mathcal{T}[Causes(\mathbf{a},\mathbf{f},\mathbf{v})] = \quad & (\mathbf{a} = load \wedge \mathbf{f} = loaded \wedge \mathbf{v} = true) \\ & (\mathbf{a} = shoot \wedge \mathbf{f} = loaded \wedge \mathbf{v} = false) \\ & (\mathbf{a} = shoot \wedge \mathbf{f} = alive \wedge \mathbf{v} = false)\end{aligned} \tag{11}$$

$$\mathcal{T}[Precond(\mathbf{f},\mathbf{a})] = \mathbf{f} = loaded \wedge \mathbf{a} = shoot \tag{12}$$

$$\mathcal{T}[Holds(loaded, S0)] = false \tag{13}$$

$$\mathcal{T}[Holds(alive, S0)] = true \tag{14}$$

$$\mathcal{T}[Holds(\mathbf{f}, result(\mathbf{a},\mathbf{f}))] = \begin{cases} \mathbf{v}, & \text{if S[a,s]} \wedge \text{T[Causes(a, f, v)]}, \\ \mathcal{T}[Holds(\mathbf{f},\mathbf{s})], & otherwise, \end{cases} \tag{15}$$

where

$$\mathcal{S}[\mathbf{a},\mathbf{s}] = \bigwedge_{\mathbf{f}:\mathcal{T}[Precond(\mathbf{f},\mathbf{a})]} \mathcal{T}[Holds(\mathbf{f},\mathbf{s})] \tag{16}$$

Lifschitz shows that there is only one minimal term model M_0, and the value of any ground atom $\mathbf{A}$ in this model is $\mathcal{T}[\mathbf{A}]$.

2.2 Problems

Lifschitz's solution involves explicit representation of those fluents whose value changes as a result of an action. The ramification problem is similar to the qualification problem but instead involves practical representation of the consequences of an action. What consequences of actions should be explicitly represented and what consequences should be inferred indirectly as a result of the direct effects and possible domain constraints? Since Lifschitz's approach requires explicit representation of *all* fluents that change value as the result of an action, the representation of ramifications of an action might cause some problems. Let's look at an example.

Suppose the following axioms are added to the YSP.

$$\forall s.Holds(dead, s) \equiv Holds\neg(alive, s) \tag{17}$$

which is a domain constraint stating that if the turkey is not alive it is dead, and

$$\neg Holds(dead, S0) \tag{18}$$

which states that the turkey is not dead in the initial situation. Based on the previous discussion where the gun is loaded, there is a waiting period and then the gun is fired, the turkey is no longer alive in situation $S3$. Given the domain constraint (18), one ramification of the shoot action should be that the turkey is dead in situation $S3$. Since the turkey is not dead in situation $S0$ and we intuitively assume that the *load* and *wait* actions do not change the value of the *dead* fluent due to the minimization of $Cause$ and the fact that all changes must have a cause, it should be possible to derive that

$$Causes(shoot, dead, true). \tag{19}$$

Unfortunately, we are unable to derive this intuitive conclusion. There is another minimal model where rather than *Shoot* causing *dead* to be *true*, *load* causes *dead* to be *true* and *alive* to be $false$. Two of the minimal models, the first intended and the second unintended are shown below.

S0	load	S1	wait	S2	shoot	S3
alive		alive		alive		¬alive
¬load		load		load		¬load
¬dead		¬dead		¬dead		dead

Causes(load,loaded,true)

Causes(shoot,loaded,false)
Causes(shoot,alive,false)
Causes(shoot,dead,true)

S0	load	S1	wait	S2	shoot	S3
alive		¬alive		¬alive		¬alive
¬load		load		load		¬load
¬dead		dead		dead		dead

Causes(load,loaded,true)
Causes(load,alive,false)
Causes(load,dead,true)

Causes(shoot,loaded,false)
Causes(shoot,alive,false)

Lifschitz has proposed distinguishing between *primitive* and *non-primitive* fluents as a way for dealing with ramification, where only primitive fluents are legal arguments for the second argument place of the $Causes$ predicate and where the values of non-primitive fluents are derived by use of the domain constraint axioms. In this case, *alive* would be a primitive fluent and *dead* non-primitive, in the sense that *dead* is defined in terms of *alive*. This solution is not completely satisfactory and somewhat ad hoc. The main problem appears to be that one must have a very good idea about all the causal connections, both direct and indirect, related to a particular domain. In the long run, it is highly unlikely that the necessary causal information will be easily accessible.

Even if one could modify the theory so ramification is dealt with in a satisfactory manner, there is still the problem of context dependent change in the value of a fluent. Quite often, an action affects a fluent value differently, dependent on what context it is performed. For example, toggling a switch will result in it being on if it was off in the initial context and off if it was on in the initial context. Recall that one reason why Lifschitz's solution worked was because the $Causes$ predicate was context independent; there was no situation argument. This would appear to rule out the representation of context dependent causes. Lifschitz proposes replacing the value or third argument slot of $Causes$ with a fluent that returns different values relative to the situation in which another fluent that will relieve the value is changed. For example, toggling the switch might include the following cause

$$Causes(toggle, on, not(on))$$

where *not* is a function which transforms a boolean fluent (such as *on*) into its negation.

In summary, it is reasonable to conclude that although the idea of explicitly representing direct affects of actions provides some insight into the frame, qualification, and ramification problems, there is still some way to go before this approach, or others like it, will be judged satisfactory.

3 Haugh's Causal Logic

The article by Haugh provides a very interesting discussion on various assumptions of *omniscience* implicit in many of the formalisms proposed for solving the frame problem. The omniscience assumptions refer to knowledge concerning

1. occurrence of events,
2. causal laws or action-effect axioms, and
3. changes of certain types.

Haugh's point is that though these omniscience assumptions may be harmless for certain reasoning tasks such as temporal projection, in general they are not feasible. Haugh presents a logic of causation that loosens a number of the more common omniscience restrictions present in other formalisms.

3.1 Omniscience Assumptions

What is meant by an assumption of omniscience? In general, the idea is that any conception of the notion of temporal persistence includes implicit assumptions about its *knowledge* of the domain. Here are a few discussed by Haugh:

3.1.1 All Successful Actions are Known

Recall the previous lecture where logics of chronological minimization were studied. The idea was that one preferred abnormal change or clippings as late in a chronology as possible. These theories presume that

- all successful action-occurrences are known.

1. When an action occurrence is successful, it creates a change in the world.
2. Action occurrences whose preconditions are of unknown truth status are unsuccessful because the delay of clippings delays their effects indefinitely.

As a consequence of the initial presumption and this behavior, such theories also presume

- that all true preconditions are known.

As a consequence of these assumptions, given a description of an action-effect axiom and a statement of its occurrence at a point t, if all the preconditions do not hold at t then the action is assumed not to have any effects and any potential change will be delayed indefinitely until there is a new occurrence where the preconditions are satisfied.

In terms of models, all minimal models reflect the fact that the effects associated with the action have not taken effect and that the action is unsuccessful. A more realistic situation would be where some of the preferred models support the effects of the action where the preconditions are true and in others do not support the effects of actions where the preconditions are false. The net result is that the presumption that all successful actions are known is violated.

3.1.2 All Fact Changes are Known

Another common presumption is that

- all changes in the values of fluents are known.

If the theory doesn't support a change in value of a fact or fluent, then that fact or fluent hasn't changed value. There are obviously an infinite number of changes that occur without our knowledge of them occurring. There are a number of ways out of this; Shoham uses a knowledge operator and restricts the assumption to one of *knowledge of* changes to persisting facts. Lifschitz distinguishes between *inertial* and *non-inertial* fluents where only with the former set do we presume complete knowledge of changes. Haugh makes similar distinction by adding a *Persists* relation to fluent-situation pairs.

3.1.3 All Event Occurrences are Known

In some of the initial solutions to the YSP, one idea was to minimize event occurrences and then constrain change to be the result of only those event occurrences explicitly axiomatized or derivable from them. In this way, one could exclude the unmotivated action of unloading the gun during the wait period. Although this works for the YSP and temporal projection problems, it presumes that

- the known occurrences of events are the only ones that exist.

The consequences of this overly strong presumption are that no other events occur then those motivated events supported by the theory. It is easy to imagine all sorts of unexplained or unmotivated events occurring in parallel with those we know about.

3.1.4 All Causal Laws are Known

A number of recent approaches to the frame problem, invert the problem statement in the following way; rather than finding a means of representing the facts that do not change in the form of a general inertia law, the possible causes for a fact changing are represented instead. In addition, one axiomatizes the fact that these are the only causes that can change the value of a fact. The underlying presumption is that

- the known causal laws or laws of change are the only ones that exist.

Such presumptions underly Lifchitz's approach considered in an earlier lecture and a number of approaches we will consider in the next lecture that use what is commonly called *explanation closure*. Such a presumption is quite strong and more or less states that the only causal laws that exist anywhere and everywhere are those that we explicitly know about. Obviously, this is not correct. At the very least, one should accept additional causal laws that are unknown but would make the same change as those existing, or causal laws whose preconditions are not true, since they do not change anything.

3.1.5 All Intervening Events are Known

In an earlier lecture, we considered a number of different temporal logics and discussed their limitations as regards allowed temporal structures, expressibility etc. The situation calculus for example has a number of omniscience assumptions built into it solely by the nature of its temporal representation. Normally, one assumes the following axiom

$$result(A, s) = result(A', s') \supset A = A' \wedge s = s'$$

which is a unique names axiom for situations. A direct consequence of the axiom is that there can only be one action occurrence between any two situations. Starting in an initial situation and performing two different actions must lead to two unique situations. Consequently, the presumption that

- all intervening events are known

is built in to the situation calculus. This rules out concurrent events and the possibility of other events occurring that we do not know about.

3.1.6 Sole Suspects are Culprits

The "sole suspects are culprits" problem is a direct result of combinations of previous assumptions. It is also interesting in the sense that it highlights the "clash of intuitions" one might have about representative examples. The "Stanford Murder Mystery" is a variation on the YSP where it is known that

- the turkey is alive in the initial situation,
- the gun is shot, and
- the turkey is not alive after the shooting.

Since the shooting is the only explanation for the turkey not being alive, it is generally accepted that the intended models should entail that the shooting was the explicit reason for the death and consequently the gun was loaded. Haugh has a different point of view and reasons that just because there is only one possible explanation axiomatized, that it should not necessarily be chosen over other potential explanations that are unknown. Many of the current theories choose the sole suspect as culprit.

3.1.7 Comments

This paper is especially interesting in the context of Features and Fluents. One of Sandewall's major claims is that before we can discuss whether a specific logic provides the intended entailments for particular scenarios such as the SMM, one must first set the context by making explicit the underlying assumptions about the world. Sandewall discusses two classes of underlying assumptions, ontological and epistemological. The latter are very similar to the type of assumptions Haugh discusses. In the book, Sandewall has very little to say about epistemological assumptions and assumes complete knowledge by the agent of action occurrences, change, etc. But unlike other approaches, these assumptions are made explicit, and it remains to construct other logics which relax these restrictions. In fact, this is a good topic for future research and worth an exjob or PhD.

In the following, we consider Haugh's attempts at relaxing the omniscience assumptions present in other formalisms. He distinguish's between justified and explained changes. An unexplained change is one that has no causal explanation. An unjustified change is one that has no causal explanation or observation-based explanation. Both types of change are based on distinguishing between different types of information: explicit knowledge or derived knowledge.

3.2 Minimizing Unexplained Changes

The following predicate constants represent explicit knowledge of an agent:

- $T_x(p, s)$ – Proposition p is asserted to be true in situation s.
- $Result_x(a, s1, s2)$ – Action a in state $s1$ is asserted to result in state $s2$.

- $Causes_x(p, c, f)$ – It is asserted that whenever cause c occurs and preconditions p hold, then result f holds in the resulting state.

Note that *Result* is now a relation and not a function. This solves the intervening events problem. More than one action may lead to the same resulting situation.

The following axioms relate explicit knowledge to the corresponding non-explicit knowledge:

$$T_x(p, s) \supset T(p, s) \tag{20}$$

$$T_x(p, s) \supset T(p, s) \tag{21}$$

$$Causes_x(p, e, f) \supset Causes(p, e, f) \tag{22}$$

Changes are represented as clippings $Clipped(f, s)$. An explained clip requires an explicitly known casual law ($Causes_x(p, e, not(f))$), whose triggering event e is explicitly known to occur ($Result_x(e, s0, s1)$), and whose preconditions p hold.

$$ExplainedClip(f, s1) \equiv [Clipped(f, s1) \wedge (\exists e, p, s0)[Result_x(e, s0, s1) \wedge Causes_x(p, e, not(f)) \wedge T(p, s0)]] \tag{23}$$

Notice that explicit knowledge of preconditions is not necessary. This weakening implies that all successful occurrences of an event e do not have to be known. In any model, when the preconds do hold, there is an explained clip, if unknown the clip is unexplained.

An unexplained clip is simply a clip that is not explained.

$$UnexplainedClip(f, s) \equiv [Clipped(f, s) \wedge \neg ExplainedClip(f, s)] \tag{24}$$

A clipping of f in s is defined as an occurrence of an event e in s' that changes f's value at s.

$$Clipped(f, s) \equiv (\exists e, s')[Result(e, s', s) \wedge T(f, s') \wedge \neg T(f, s)] \tag{25}$$

The temporal ordering of states is defined in terms of $Before$:

$$Before(s', s) \equiv (\exists e)[Result(e, s', s) \vee (\exists s'')[Before(s', s'') \wedge Result(e, s'', s)]] \tag{26}$$

$$Before(s', s) \supset \neg Before(s, s') \tag{27}$$

Other basic facts about truths:

$$T(not(f), s) \supset \neg T(f, s) \tag{28}$$

$$f = not(not(f)) \tag{29}$$

$$T(True, s) \tag{30}$$

The consequence of a generic causal relation whose preconditions hold and whose event occurs results in it's effects holding.

$$Causes(preconds, event, effect) \supset$$
$$(\forall s, s')[[T(preconds, s) \wedge Result(event, s, s')] \supset T(effect, s')] \quad (31)$$

The axioms T1-T14 provide a theory for minimizing *unexplained change* which avoids some of the omniscience problems present in previous causal based solutions to the frame problem.

1. Does not require explicit knowledge of the truth of preconditions (23).
2. All event occurrence need not be known. Any event can occur that does not introduce unexplained change. Of course, this restricts such events to making changes that are already made by other events that make the same change and have explanations.
3. All causal rules, both explicit and generic, need not be known. Any unknown causal rules may be admitted as long as the changes that would result from them already belong to those admitted by the theory.

The minimization policy associated with the theory would be to first minimize explicit relations (R_x) and then to minimize unexplained changes ($UnexplainedClip$). So, a form of prioritized minimization is necessary.

The trick is that one decouples cause and explanation by introducing $UnexplainedClip$ and defining it indirectly in terms of $Causes$ and $Results$. Minimizing $UnexplainedClip$ does not minimize $Cause$ or $Result$. Consequently, additional causal rules and event occurrences other than those explicitly axiomatized may exist as long as they do not introduce new unexplained clippings.

3.2.1 Example and Potential Problem

The following is an axiomatization of the Stanford Murder Mystery.

$$T_x(alive, S0) \quad (32)$$

$$Result_x(shoot, S0, S1) \quad (33)$$

$$Result_x(wait, S1, S2) \quad (34)$$

$$T_x(not(alive), S2) \quad (35)$$

The general causal relations:

$$Causes_x(loaded, shoot, not(alive)) \quad (36)$$

The present logic still suffers from the "sole culprit problem". Models in which the gun is loaded will contain an explained clip for *alive*, whereas those where the gun is unloaded will contain an unexplained clip. Upon minimization of unexplained clippings, we will be able to derive that the gun is loaded and therefore that the shooting action caused the clipping of *alive*. If we agree with Haugh that the potential causal explanations for the clipping are not the same as the actual causal explanations, then the present policy must be modified.

3.3 Minimizing Unjustified Changes

Haugh generalizes the notion of justified change to include not only change justified via causal explanation, but also to change justified via direct observation or via explicit knowledge of change independent of whether there is a causal justification. As an example, recall the SMM. It should suffice as an explanation for change that we observe the turkey being alive in the initial situation and not alive in a later situation. In this case, a causal law should not necessarily be required to provide an explanation for the change.

Justifying Changes via Explicit Truths:

A justified clip is either one that has a causal explanation ($ExplainedClip(p,s)$), or one that is explained via explicit truths ($TJustifiedClip(p,s)$).

$$JustifiedClip(p,s) \equiv [ExplainedClip(p,s) \vee TJustifiedClip(p,s)] \tag{37}$$

A truth justified clipping for p at s is a clipping at a situation s where $not(p)$ is true at s, there is explicit knowledge of $not(p)$ being true at a situation $s2$ after s, and for all situations between s and $s2$, $not(p)$ is true.

$$\begin{aligned} TJustifiedClip(p,s) \equiv [&Clipped(p,s) \wedge \\ &(\exists s2)[[Before(s,s2) \vee s = s2] \wedge T_x(not(p),s2) \wedge \\ &(\forall s1)[[Before(s,s1) \wedge Before(s1,s2)] \supset T(not(p),s1)]]] \end{aligned} \tag{38}$$

$$UnJustifiedClip(p,s) \equiv [Clipped(p,s) \wedge \neg JustifiedClip(p,s)] \tag{39}$$

As before, one would first minimize explicit knowledge and then $UnjustifiedClip$.

If this new policy is applied to to the SMM, then there will be alternative explanations as to why the turkey is not alive. These explanations will be justified by the explicit knowledge of $not(alive)$ is $S2$.

3.3.1 Limited Persistence

One final modification that Haugh proposes is to narrow the scope of our persistence assumptions. Rather than assume that all facts or fluents persist indefinitely, he introduces a $Persists$ predicate applied to a fact and situation. One can then specify which facts are subject to the minimization policy. Only persisting facts will need a justification when changing value. Since non-persisting facts can change without justification, there changes play no role in preferring one model to another. One minimizes unjustified clippings only when they clip a persisting fact.

$$\begin{aligned} UnJustifiedClipP(p,s) \equiv [&Clipped(p,s) \wedge \neg JustifiedClip(p,s) \\ &(\exists s0)[Persistes(p,S0) \wedge Before(s0,s) \wedge \\ &(\forall s1)[[Before(s0,s1) \wedge Before(s1,s)] \supset T(p,s1)]]] \end{aligned} \tag{40}$$

The existential part states that there is a situation before the clipping where the fact persists and it stays true in all situations between that point and the clipping point.

Our previous axiom (31), has to be modified. The effect of a causal rule now has to be stated to persist rather than be true. Truth of a persisting fact is derived indirectly via another axiom.

$$\begin{aligned} &Causes_x(preconds, event, effect) \supset \\ &(\forall s, s')[[T(preconds,s) \wedge Result(event,s,s')] \supset Persists(effect,s')] \end{aligned} \tag{41}$$

Several axioms are needed to relate persistence and truth.

$$Perists(p,s) \supset T(p,s) \tag{42}$$

$$Perists_x(p,s) \supset Persists(p,s) \tag{43}$$

$$Perists_x(p,s) \supset T_x(p,s) \tag{44}$$

3.3.2 Example2: The YSP

The following is an axiomatization of the Yale Shooting Problem based on the policy of minimizing unjustified change and limiting the minimization to persisting fluents.

$$Persists_x(alive, S0) \tag{45}$$

$$Result_x(load, S0, S1) \tag{46}$$

$$Result_x(wait, S1, S2) \tag{47}$$

$$Result_x(shoot, S2, S3) \tag{48}$$

The general causal relations:

$$Causes_x(true, load, loaded) \tag{49}$$

$$Causes_x(loaded, shoot, not(alive)) \tag{50}$$

Unique Names Axioms:

$$\begin{aligned} load \neq wait \wedge load \neq shoot \wedge wait \neq shoot \\ \wedge loaded \neq true \wedge loaded \neq not(alive) \\ \wedge alive \neq loaded \wedge alive \neq true \wedge true \neq not(alive) \end{aligned} \tag{51}$$

3.3.3 Computing Preferred Models

In the appendix of the article, Haugh goes through the YSP example in detail, using a combination of predicate completion for circumscribing explicit knowledge and variable circumscription for minimizing $UnjustifiedClip$.