Lecture 11 − Causal Minimization

1. Causal Minimization
   (a) Lifschitz's Theory of Action
   (b) Haugh's Causal Logic
      i. Omniscience Assumptions
      ii. Minimizing Unexplained Change
      iii. Minimizing Unjustified Change
2. Conclusions

Lifschitz's Logic of Action

– Lifschitz's initial solution involved use of pointwise circumscription. Disadvantage due to difficulty in computing preferred entailments.

– In the new approach, Lifschitz uses standard parallel predicate circumscription, but changes the ontology.

– The proposed solution deals with both the frame and qualification problems.

Basic Idea:

– Causes of changes to fluent values and preconditions to action descriptions are made explicit in terms of two new predicate constants: $Causes$ and $Preconds$.

– Intended models for a scenario description characterized by the fact that **all changes in the values of fluents are caused by actions**.

– In addition, **the only successful actions are those whose preconditions are true**. If an action is successful, it causes a change.

– Minimize $Causes$ and $Preconds$ in parallel.

Yale Shooting Problem

Preliminaries:

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

$$d(result) = \langle a \times s \to s \rangle.$$

3. $R = \{Holds, Causes, Precond, Success, Affects\}$ where*

$$d(Holds) = \langle f \times s \rangle,$$

$$d(Causes) = \langle a \times f \times v \rangle,$$

$$d(Precond) = \langle f \times a \rangle,$$

*Both $Success$ and $Affects$ are defined predicate constants and not primitive.

$$d(Success) = \langle a \times s \rangle,$$

$$d(Affects) = \langle a \times f \times s \rangle,$$

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.

– 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.

YSP Axiomatization

– 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:

$$Causes(load, loaded, true) \tag{3}$$

$$Causes(shoot, loaded, false) \tag{4}$$

$$Causes(shoot, alive, false) \tag{5}$$

Qualification:

– 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,

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

Inertia:

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

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

– 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,

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

Circumscription Policy

– Let Γ be the axiom set described above, excluding the definitions for $Success$ and $Affects$.

–The circumscription policy proposed by Lifschitz is

- **Minimize:** $Causes$ and $Preconds$
- **Vary:** $Holds$

–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.

<u>Intuitions</u>

In summary, the key intuition as to why this solution works for the YSP and earlier uses of global circumscription don't is summerized 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, [**?**])

Computing Preferred Models

– 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:

$$\mathcal{T}[Causes(\mathbf{a},\mathbf{f},\mathbf{v})] = \begin{array}{c}(\mathbf{a} = load \wedge \mathbf{f} = loaded \wedge \mathbf{v} = true)\\ (\mathbf{a} = shoot \wedge \mathbf{f} = loaded \wedge \mathbf{v} = false)\\ (\mathbf{a} = shoot \wedge \mathbf{f} = alive \wedge \mathbf{v} = false)\end{array}$$

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

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

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

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

where

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

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 $\mathcal{T}[\mathbf{A}]$.

Problems

Ramification:

– Suppose the following axioms are added to the YSP.

$$\forall s.Holds(dead, s) \equiv Holds\neg(alive, s) \tag{9}$$

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

$$\neg Holds(dead, S0) \tag{10}$$

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 (**??**), 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

$$Causes(shoot, dead, true). \tag{11}$$

– 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.

Two Minimal Models

– The first intended, the second unintended.

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

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

Another Problem: Context Dependent Change

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.

Haugh's Logic of Causality

– Haugh presents a logic of causality based on a reified temporal logic with a much richer ontology than Lifschitz.

– He tries to describe and avoid a number of epistemological assumptions about what an agent knows about the world. Such assumptions are implicit in many exisiting formalisms and Haugh claims that they are too strong.

– In order to lift the various assumptions, Haugh distinguishes syntactically between two types of knowledge, explicit and derived.

– In addition, he distinguishes between two types of justification for a fluent change:

1. Unexplained Change

2. Unjustified Change

– Minimization techniques are based on a combination of Clarke's completion and circumscription or just prioritized circumscription.

Omniscience Assumptions

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.

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:

- All Successful Actions are Known.

- All Fact Changes are Known.

- All Event Occurrences are Known.

- All Causal Laws are Known.

- All Intervening Events are Known.

- Sole Suspects are Culprits.

All Successful Actions are Known

– Assumption that all successful action occurrences are known implies that all true preconditions are known.

– If the truth of preconditions is unknown, most theories assume they are false and consequently the action occurrence is unsuccessful.

– Better way to interpret an action occurrence where the precondtions are unknown is to interpret it as successful and causing change in those models where the precondtions are satisfied, and to interpret it as unsuccessful in models where the preconditions are false.

– In this way one can distinguish *potential* change from actual change.

All Fact Changes are Known

– Many approaches use the following assumption:

All changes in the values of fluents are known.

– Obviously, there are many changes that occur around us without are explicit knowledge.

Several Approaches:

– Lifschitz distinguishes between *inertial* and *noninertial* fluents.

– Shoham restricts the assumption to one of *knowledge of changes* to persisting facts.

– Haugh distinguishes between *persistent* and *nonpersistent* fluents.

All Event Occurrences are Known

Another assumption is

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

– For example, unloading the gun during the wait period is an unmotivated action. Nothing has explicitly been stated about its occurrence.

– To avoid such actions, one can minimize the occurrences of actions and then state that the only way a fluent changes value is if there is a motivated or known action occurrence.

– Morganstern and Stein – Motivated Action Theory. (minimize occurrences).

– Lifschitz – Logic of action. (minimize types).

– Obviously unrealistic to assume that other actions, unexplained or unmotivated, do not occur in parallel with those actions we know about.

All Causal Laws are Known

Another assumption is

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

– 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,

1. the possible causes for a fact changing are represented and

2. an additional axiom states that these are the only causes that can change the value of a fact.

– Lifschitz's logic of action: Minimize cause.

– Explanation Closure (Reiter and Schubert).

All Intervening Events are Known

Another assumption is that

All intervening events are known.

– The situation calculus implicitly assumes that only one action occurrence can occur between any two situations:

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

– Starting at an initial situation and performing two different actions, must lead to two different situations.

– As stated, rules out concurrent events and the possibility of events occurring that we do not know about.

The Sole Culprits Problem

– The sole culprits problem is the result of a combination of the previous assumptions. It can be described with an example, the Stanford Murder Mystery:

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

– Since the shooting action is the only explanation (sole culprit) for the turkey expiring, the preferred models should entail that the gun was loaded and the shooting action caused the death.

– Most people generally agree with this result, but here is a similar scenario:

- I park my car.
- I leave it for two days.
- It is not there when I return.

– One explanation is that it was stolen sometime during the two days.

– Suppose, I only have a causal law for eating breakfast. By the sole culprit policy, since eating breakfast is the only explicit cause axiomatized, the explanation for the car disappearing might just be my eating breakfast.

Minimizing Unexplained Changes

–The following predicate constants represent **explict 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**:

$$T_x(p, s) \supset T(p, s) \tag{12}$$

$$T_x(p, s) \supset T(p, s) \tag{13}$$

$$Causes_x(p, e, f) \supset Causes(p, e, f) \tag{14}$$

– **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.

$$\begin{aligned} ExplainedClip(f, s1) \equiv [&Clipped(f, s1) \wedge \\ &(\exists e, p, s0)[Result_x(e, s0, s1) \wedge \\ &Causes_x(p, e, not(f)) \wedge T(p, s0)]] \end{aligned} \quad (15)$$

– Notice that explicit knowledge of preconditions is not necessary. 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.

$$\begin{aligned} &UnexplainedClip(f, s) \equiv \\ &\quad [Clipped(f, s) \wedge \neg ExplainedClip(f, s)] \end{aligned} \quad (16)$$

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

$$\begin{aligned} &Clipped(f, s) \equiv \\ &\quad (\exists e, s')[Result(e, s', s) \wedge T(f, s') \wedge \neg T(f, s)] \end{aligned} \quad (17)$$

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

$$\begin{aligned} Before(s', s) \equiv (\exists e)[Result(e, s', s) \\ \vee(\exists s'')[Before(s', s'')\wedge Result(e, s'', s)]] \end{aligned} \tag{18}$$

$$Before(s', s)\supset\neg Before(s, s') \tag{19}$$

– Other basic facts about truths:

$$T(not(f), s)\supset\neg T(f, s) \tag{20}$$

$$f = not(not(f)) \tag{21}$$

$$T(True, s) \tag{22}$$

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

$$\begin{aligned} & Causes(preconds, event, effect)\supset \\ & \quad (\forall s, s')[[T(preconds, s)\wedge Result(event, s, s')]\supset \\ & \qquad T(effect, s')] \end{aligned} \tag{23}$$

Comments

–The axioms 12-23 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 (15).

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.

The Stanford Murder Mystery

The following is an axiomatization of the Stanford Murder Mystery.

$$T_x(alive, S0) \tag{24}$$

$$Result_x(shoot, S0, S1) \tag{25}$$

$$Result_x(wait, S1, S2) \tag{26}$$

$$T_x(not(alive), S2) \tag{27}$$

The general causal relations:

$$Causes_x(loaded, shoot, not(alive)) \tag{28}$$

– 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 $alive$.

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 ($TJustifiedClip(p, s)$).

$$JustifiedClip(p, s) \equiv \\ [ExplainedClip(p, s) \vee TJustifiedClip(p, s)] \quad (29)$$

– 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.

$$TJustifiedClip(p, s) \equiv [Clipped(p, s) \wedge \\ (\exists s2)[[Before(s, s2) \vee s = s2] \wedge T_x(not(p), s2) \wedge \\ (\forall s1)[[Before(s, s1) \wedge Before(s1, s2)] \supset T(not(p), s1)]]] \quad (30)$$

$$UnJustifiedClip(p, s) \equiv \\ [Clipped(p, s) \wedge \neg JustifiedClip(p, s)] \quad (31)$$

– 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 alterative explanations as to why the turkey is not alive. These explanations will be justified by the explicit knowledge of $not(alive)$ is $S2$.

Limited Persistence

– Haugh proposes narrowing 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.

$$
\begin{aligned}
&UnJustifiedClipP(p,s) \equiv \\
&\qquad [Clipped(p,s) \wedge \neg JustifiedClip(p,s) \\
&\qquad (\exists s0)[Persistes(p,S0) \wedge Before(s0,s) \wedge \\
&\qquad (\forall s1)[[Before(s0,s1) \wedge Before(s1,s)] \supset T(p,s1)]]] \quad (32)
\end{aligned}
$$

–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.

– The previous axiom (23), 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.

$$\begin{aligned}&Causes_x(preconds, event, effect) \supset \\ &\quad (\forall s, s')[[T(preconds, s) \wedge Result(event, s, s')] \supset \\ &\qquad Persists(effect, s')]\end{aligned} \tag{33}$$

– Several axioms are needed to relate persistence and truth.

$$Perists(p, s) \supset T(p, s) \tag{34}$$

$$Perists_x(p, s) \supset Persists(p, s) \tag{35}$$

$$Perists_x(p, s) \supset T_x(p, s) \tag{36}$$

The Yale Shooting Problem

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.

$$Persists_x(alive, S0) \tag{37}$$

$$Result_x(load, S0, S1) \tag{38}$$

$$Result_x(wait, S1, S2) \tag{39}$$

$$Result_x(shoot, S2, S3) \tag{40}$$

The general causal relations:

$$Causes_x(true, load, loaded) \tag{41}$$

$$Causes_x(loaded, shoot, not(alive)) \tag{42}$$

Unique Names Axioms:

$$\begin{aligned} load \neq wait \wedge load \neq shoot \wedge wait \neq shoot \\ \wedge loaded \neq true \wedge loaded \neq not(alive) \\ \wedge alive \neq loaded \wedge alive \neq true \wedge true \neq not(alive) \end{aligned} \tag{43}$$