Nonmonotonic Logics: An Overview

Leora Morgenstern
April 5, 1999
CS 4725
Columbia

What is a nonmonotonic logic/ nonmonotonic reasoning?

Nonmonotonic logic, in contrast to classical logic (propositional, first order, modal) has the property of nonmonotonicity.

Classical logics are <u>monotonic</u> in the set of assumptions, that is, the more assumptions you add, the more theorems you add. Formally, if Th(Γ) stands for the set of consequences (theorems) of a set of formulas Γ, then monotonicity can be stated as follows:

if Γ_1 is a subset of Γ_2, then Th(Γ_1) is a subset of Th(Γ_2)

"monotonic logic means never having to say you're sorry" **--*Benjamin Grosof***

In a nonmonotonic logic, as you add more conclusions you may have to retract some conclusions
So in general,
even if Γ_1 is a subset of Γ_2 it may not be the case that
Th(Γ_1) is a subset of Th(Γ_2)

Nonmonotonic reasoning is often called default or plausible reasoning because .....

Nonmonotonic logics are important for default, plausible reasoning because one often has to *retract* conclusions

From: Γ_1
Birds (typically) fly
Penguins are birds
Penguins don't fly
Tweety is a bird
Conclude: Tweety flies

From: Γ_2
Birds (typically) fly
Penguins are birds
Penguins don't fly
Tweety is a bird
Tweety is a penguin
Conclude: Tweety doesn't fly

If we know the facts in Γ_1, we will (plausibly) conclude that Tweety flies. If we find out one extra fact (Tweety is a penguin), we will have to retract that conclusion.

Because default or plausible reasoning is only plausible, and not definite, we need a mechanism for retracting conclusions.

That's what nonmonotonic logic provides.

General uses for nonmonotonic reasoning:

(variations on theme of plausible reasoning)

- **To represent and reason with plausible information**
  - **Birds typically fly**
  - **Most college students are adults**

- **To reason in the absence of complete information**
  - **you don’t know *what* type of bird Tweety is**

- **To avoid enumerating all conditions or exceptions to a rule (qualification problem)**

  Instead of writing

  If I turn the key in the ignition and the battery works and there’s gas in the tank and there’s no potato in the tailpipe and ... then the car will start

  write:

  Usually when I turn the key in the ignition, the car will start

Some limited forms of nonmonotonic reasoning

•Domain Closure Assumption

Assume that the theory specifies all the objects that there are

•Closed World Assumption

Assume that if you can’t infer something (positive), it’s false

Domain Closure Assumption

General idea: When you're stating a problem, you specify all the entities.

For example, in the missionaries and cannibals problem, if you're told that there are 3 missionaries, 3 cannibals, and a rowboat, common sense tells us that there are no more than 3 missionaries, no more than 3 cannibals, and just 1 rowboat. If you were to solve the problem by positing 2 rowboats, for example, you'd be cheating.

Formally: Given a theory T, assume that the only objects that exist are those named by <u>ground terms in the language of T,</u> that is, by constant symbols, and function symbols applied to constant symbols.

Why is domain closure nonmonotonic? Because if you add a new statement to the theory, naming a new object, your domain closure statement is weakened.

Example:
T1: On(A, Table) On(B,A)
Domain Closure "Assumption": (forall x)(x=Table v x=A v x=B)
T2: On (A,Table) On(B,A) On(C,A)
Domain Closure "Assumption": (forall x)(x=Table v x=A v x=B v x=C)

Note: T1 subset of T2, but T1's Domain Closure Assumption is stronger than T2's

Closed World Assumption

General Idea: If something is true, you say it or make sure it's derivable. So if you can't derive something, it's probably false.

Motivating example: an airline database, giving flight segments.
Assume the following database:
Flight(NYC,Boston), Flight(NYC, SF), Flight(NYC, LA), Flight(SF, LA), Flight(Boston, Toronto),Flight(Toronto, Phoenix), Flight(Pensacola, Atlanta)
and the rules: Flight(x,y) <==> Flight(y,x)
Connects(x,y) <==> Flight(x,y) v (exists z) (Connects(x,z) & Connects(z,y))
Then the answer to the question:
Is there a connection between Pensacola and Boston?
should be NO ---- even though this is not a true consequence of the theory.

Formally: Let T be a theory, and φ be a <u>ground atomic formula</u>, that is, a formula without variables or Boolean connectives, including negation. If φ cannot be inferred from T, nonmonotonically infer $\sim\varphi$.

Why is the closed world assumption nonmonotonic? Consider adding to the above database the sentence *Flight(Atlanta,NY)*. Then, the answer to the question: *is there a connection between Pensacola and Boston* is now YES.

More about the Closed World Assumption

The closed world assumption is non-computable (= non-decidable).
There is no general procedure to determine whether φ follows from T!

The closed world assumption can be inconsistent. This can happen in a variety of ways.

1. Disjunctions.
   Assume a theory T: P(A) v Q(A).
   Since P(A) is not derivable, we can infer ~P(A).
   Since P(B) is not derivable, we can infer ~P(B).
   But these 2 statements added to T give an inconsistent theory.

2. Multiple jointly-inconsistent predicates.
   Assume a theory T: (forall x) (P(x) <==> ~Q(x))
   P(A) is not derivable; so we can infer ~P(A).
   Q(A) is not derivable; so we can infer ~Q(A).
   But ~P(A) and ~Q(A) are inconsistent.
   MORAL: generally wish to apply CWA to only some predicates.

3. Closed world assumption restricted to the equality predicate is the nonmonotonic version of our old friend, the unique names assumption.

Major systems of nonmonotonic reasoning

- **Default logic**

Add new inference rule for each default rule

Typically birds fly == Bird(x): Flies(x)

Flies(x)

If x is a bird, and it's consistent that x flies, then conclude that x flies

- **Circumscription**

Stay in classical logic; add axiom schema or 2nd-order axiom that circumscribes or limits the extension of predicates.
E.g., can circumscribe the extension of non-flies.
(note: we will instead circumscribe "abnormality" predicates)

- **Autoepistemic logic**

Modal logic: Add new inference rule to let you reason from absence of information to the fact that you don't believe it's so.

L(Bird(x)) & ~L~(Flies(x)) ==> Flies(x)

From the absence of info about x flying, infer you don't believe x doesn't fly; then rule triggers.

If you believe x is a bird, and you don't believe that x doesn't fly, then conclude that x flies

- **Consequence relations**

Change the entailment relation |= and give it different properties; the most flexible but can be the hardest to use.

Note:
The major problem with nonmonotonic reasoning is that it's non-decidable.
There's generally an appeal to consistency, and that's not decidable.

Default Logic (Reiter, 1980)

Add new set of inference rules to classical logic

$$\frac{\alpha(x) : \beta(x)}{\gamma(x)}$$

If α(x) is true, and it's consistent that β(x), then conclude that γ(x) is true

E.g.,

$$\frac{\text{Bird}(x)\text{: Flies}(x)}{\text{Flies}(x)}$$

Note that in this case $\beta(x) = \gamma(x)$.

Defaults of the form $\dfrac{\alpha(x) : \beta(x)}{\beta(x)}$ **are known as <u>normal defaults</u>**

Defaults of the form $\dfrac{\alpha(x) : \beta(x) \ \& \ \gamma(x)}{\gamma(x)}$ **are <u>semi-normal defaults</u>**

$$\frac{\alpha(x) : \beta(x)}{\gamma(x)}$$

any default

$$\frac{\alpha(x) : \beta(x)}{\beta(x)}$$

normal default **great properties: semi-monotonicity**

$$\frac{\alpha(x) : \beta(x) \ \& \ \gamma(x)}{\gamma(x)}$$

semi-normal default

Reiter's original hope: all "natural" defaults could be written as normal defaults (turns out not to be the case)

Note: there is more than one way to write things

E.g., if I take flight 104, it will probably land at 10 am (as long as there's no bad weather, no delays, etc.)

regular default

Can represent as

$$\frac{\text{flight-104}(x)\text{: \textasciitilde Bad-weather}(x)\text{, Delay}(x)}{\text{Land-at-10am}(x)}$$

Or as:

$$\frac{\text{flight-104}(x)\text{: Land-at-10am}(x)}{\text{Land-at-10am}(x)}$$

normal default

Bad-weather(x) ==> ~Land-at-10am(x); Delay(x) ==> ~Land-at-10am(x)

The formal definition of a default theory:

Theory is a pair (W,D) where
W is a set of well-formed formulae
D is a set of defaults

Tweety example:

W:

Bird(Tweety)
(forall x)(Penguin(x) ==> Bird(x))
(forall x)(Penguin(x) ==> ~Flies(x))
(forall x)(Flies(x) ==> Winged(x))

D:

$$\frac{\text{Bird}(x) : \text{Flies}(x)}{\text{Flies}(x)}$$

How to do inference in a default theory:
Keep applying as many defaults as possible

Inference in a default theory:

The formal definition:

Let Δ = (W,D) be a (closed)* default theory,
with defaults as previously defined.
For any set of (closed) wffs S, let Γ(S)
be the smallest set satisfying the following properties:

1. W is a subset of Γ(S)
2. Th(Γ(S)) = Γ(S)
3. If α(x) : β(x)/ γ(x) in D, and α(x) in Γ(S) and ~ β(x) not in S, then γ(x) in Γ(S)

A set E is an extension for Δ if Γ(E) = E; i.e. E is a fixed point of the operator Γ

W:

Bird(Tweety)
(forall x)(Penguin(x) ==> Bird(x))
(forall x)(Penguin(x) ==> ~Flies(x))
(forall x)(Flies(x) ==> Winged(x))

D:

$$\frac{\text{Bird(x): Flies(x)}}{\text{Flies(x)}}$$

---

* a closed default theory is one where for each default, α β γ etc. are closed wffs (sentences). In practice, most theories are not closed. The formal definition is usually given for closed theories; for open theories, the def. is similar, once we skolemize,

Inference in a default theory:

A different (more intuitive) characterization:

Let Δ = (W,D) be a (closed)* default theory, with defaults as previously defined.

Define $E_0 = W$
for i greater than 0,

$E_{i+1} = Th(E_i) \cup$

$\{\gamma \mid \alpha(x) : \beta(x)/ \gamma(x)$ in D, and $\alpha(x)$ in $\Gamma(S)$ and $\sim \beta(x)$ not in S$\}$

Then E is an extension for Δ if $E = \bigcup_{i=0 \text{ to inf.}} E_i$

i.e. keep applying defaults until you can't anymore.
Theorem: these two definitions of extension are equivalent.

In the Tweety example, get:
Tweety(flies); Winged(Tweety)

W:

Bird(Tweety)
(forall x)(Penguin(x) ==> Bird(x))
(forall x)(Penguin(x) ==> ~Flies(x))
(forall x)(Flies(x) ==> Winged(x))

D:

$$\frac{\text{Bird(x): Flies(x)}}{\text{Flies(x)}}$$

---

* a closed default theory is one where for each default, α β γ etc. are closed wffs (sentences). In practice, most theories are not closed. The formal definition is usually given for closed theories; for open theories, the def. is similar, once we skolemize,

Sometimes (usually) you get more than one extension for a theory

Nixon diamond:

W:
Quaker(Nixon)
Republican(Nixon)
D:

$$\frac{\text{Quaker}(x):\ \text{Pacifist}(x)}{\text{Pacifist}(x)} \qquad \frac{\text{Republican}(x):\ \sim\text{Pacifist}(x)}{\sim\text{Pacifist}(x)}$$

Two extensions:
EI contains Pacifist(Nixon) (from first default), but then can’t apply 2nd default
EII contains ~Pacifist(Nixon(from second default), but then can’t apply 1rst default

Credulous reasoning: accept statements in any one extension
Skeptical reasoning: accept statements true in <u>all</u> extensions (intersection)

Sometimes, defaults can interact in unexpected ways: (Reiter & Criscuolo)

Note how "typically" is not transitive:
Typically high school dropouts are adults
Typically adults are employed

But not: Typically high school dropouts are employed!

So, transitivity must be blocked.

That is, we don't always want to represent
Typically A's are B's
Typically B's are C's
as:

$$\frac{A(x): B(x)}{B(x)} \quad \text{and} \quad \frac{B(x): C(x)}{C(x)}$$

Because then we would in fact conclude from A(x) that C(x).
How we change the representation depends on the interaction of A, B, and C ...

In general, non-normal defaults are needed

Because "typically" is not transitive:

Typically high school dropouts are adults

Typically adults are employed

But not: Typically high school dropouts are employed!

We want to block transitivity.

We usually have to use non-normal defaults to do so.

How we block transitivity depends on the way A, B, & C interact. Some cases:

Case I: All we know is that it is not the case that A's are typically C's

A(x): B(x) / B(x)
B(x): ~(A(x) & C(x))/ C(x)

Case II: In addition to knowing that it is not the case that A's are typically C's, we also know that typically B's are not A's. For example, we know that typically, adults are not high school dropouts.

A(x): B(x)/ B(x), B(x): ~A(x)/ ~A(x),
B(x) & ~A(x) : C(x) / C(x)

Case III: We know that typically A's are not C's and that typically B's are not A's. (Example: Let A be big-time lottery winners; Let B be millionaires; Let C be people who graduated from private colleges.)

A(x): B(x) / B(x); B(x): ~A(x)/~A(x),
B(x) & ~A(x): C(x))/ C(x)
A(x): ~C(x) / ~ C(x)

Note: normal defaults suffice if you know that typically B's are not A's.

Circumscription (McCarthy, 1980; 1986)

Basic idea: Restrict the set of objects satisfying a predicate to the objects it <u>must</u> satisfy

Example:

(forall x) (Bird(x) ==> Winged(x))
Bird(Tweety)
Penguin(Opus)

(forall x)(Penguin(x) ==> Bird(x))
Winged(Dracula)
Mortal(Socrates)

Intuitively, we would expect that only 2 objects (Tweety, Opus) are birds; only 3 objects (Tweety, Opus, Dracula) are winged. But that's not inherent in the theory; it's consistent that Dracula is a bird; that Socrates is winged.

Achieve what we want by <u>CIRCUMSCRIBING</u> over the predicates *bird* and *winged*.

How do we achieve this? To circumscribe predicate P in a theory T
--- We add an axiom schema (or 2nd order axiom):
[T[P/α] & (forall x) α(x) ==> P(x)] ==> (forall x) P(x) ==> α(x)]
--- which is the same as inducing a <u>preference relation</u> among models of the theory.

This second characterization is easier to understand and work with.

Circumscription as a preference relation among models *(i.e. a wayof specifying which models are best)*

Consider the theory

(forall x) (Bird(x) ==> Winged(x))
Bird(Tweety)
Penguin(Opus)

(forall x)(Penguin(x) ==> Bird(x))
Winged(Dracula)
Mortal(Socrates)

There are (typically infinitely) many models satisfying this theory.
Here are some:

M1: Bird(Tweety), Penguin(Opus), Winged(Dracula), Mortal(Socrates), Bird(Opus), Winged(Tweety), Winged(Opus)

M2: Bird(Tweety), Penguin(Opus), Winged(Dracula), Mortal(Socrates), Bird(Opus), Winged(Tweety), Winged(Opus), Bird(Dracula)

M3: Bird(Tweety), Penguin(Opus), Winged(Dracula), Mortal(Socrates), Bird(Opus), Winged(Tweety), Winged(Opus), Bird(Dracula), Winged(Socrates)

M4: Bird(Tweety), Penguin(Opus), Winged(Dracula), Mortal(Socrates), Bird(Opus), Winged(Tweety), Winged(Opus), Bird(Socrates), Winged(Socrates)

Note that M1 has just the amount of info we want; no extraneous birds or winged creatures. In fact, M1 seems preferable to M2, which seems preferable to both M3 and M4 (M3 and M4 are in a sense incomparable)

Circumscription as a preference relation among models (continued):

How can we formalize this?
We say that M1 is preferred to M2 if M1 and M2 agree on all predicates except for P (the predicate we're interested in circumscribing); and the extension of P in M1 is a subset of the extension of P in M2:

M1 $<_{pref}$ M2 iff
for all predicates Q such that Q not equal to P,
forall x, M1 |= Q(x) iff M2 |= Q(x)
and
for all x M1 |= P(x) ==> M2 |= P(x)
but there exists some c such that M2 |= P(c)
but it's not the case that M1 |= P(c)

The $<_{pref}$ relation induces a partial order on models.

We choose the minimal models in the partial order.
These are the "preferred" models.

Circumscription: Working through the axiom schema

So, what is this axiom schema supposed to mean?

[T[P/α] & (forall x) α(x) ==> P(x)] ==> (forall x) P(x) ==> α(x)]

**Actually, it just forces the extension of P to be as small as possible.
What we're saying is:
Take any predicate α at all.
The circumscriptive axiom schema says that
if it is true in the theory whenever P is (i.e., you could substitute α for P
in T), and the extension of α is smaller than P, then P is smaller than α,
i.e., P is exactly the same size as α. That is, you can't pick anything smaller than
P which satisfies the theory .
That is, P is as small as we can get.**

Circumscription: Working through the axiom schema

[T[P/α] & (forall x) α(x) ==> P(x)] ==> (forall x) P(x) ==> α(x)]

Here is how this schema is derived and used, working through an example of Davis (1990):

Assume a theory:

(forall x) (kangaroo(x) ==> marsupial(x))
(forall x) (opossum(x) ==> marsupial(x))
(forall x) (cat(x) ==> ~(kangaroo(x) v opossum(x))

We're interested in cirumscribing marsupial; i.e., forcing marsupial to be as small as possible. That is, suppose we take a predicate α(x) which satisfies the axioms on marsupial. That is, suppose
(forall x)(kangaroo(x) ==> α(x)); (forall x)(opossum(x) ==> α(x))
Certainly, we don't want some marsupials to not be α's; because if so, we could just reduce the predicate marsupial to α ! We can write this as:
(forall x)(kangaroo(x) ==> α(x)) & (forall x)(opossum(x) ==> α(x)) ==>
~(forall x)(α(x) ==> marsupial(x) & (exists x)(marsupial(x) & ~ α(x))

Which is equivalent to:
[(forall x)(kangaroo(x) ==> α(x)) & (forall x)(opossum(x) ==> α(x)) &
(forall x)(α(x) ==> marsupial(x))] ==> (forall x)(marsupial(x) ==> α(x))

Now pick α(x) to be the formula kangaroo(x) v opossum(x)

Substituting in the axiom schema, we get:

[(forall x)(kangaroo(x) ==> kangaroo(x) v opossum(x)) &
(forall x)(opossum(x) ==> kangaroo(x) v opossum(x)) &
(forall x)kangaroo(x) v opossum(x) ==> marsupial(x))] ==>
(forall x) (marsupial(x) ==> kangaroo(x) v opossum(x))

What does this formula give us? Let's look at it clause by clause

[(forall x)(kangaroo(x) ==> kangaroo(x) v opossum(x)) & *trivially true*
(forall x)(opossum(x) ==> kangaroo(x) v opossum(x)) & *trivially true*
(forall x)kangaroo(x) v opossum(x) ==> marsupial(x))] ==> *true by axioms of theory*
(forall x) (marsupial(x) ==> kangaroo(x) v opossum(x)) TRUE because antecedents are

That is, we get
(forall x)(marsupial(x) ==> kangaroo(x) v opossum(x))

That is, the only marsupials are kangaroos or opossums; just what we wanted.

Using Circumscription: the Tweety example

We usually formalize default rules by using abnormality predicates

Here's how one might formalize a limited theory of ornithology:
(all variables assumed to be universally quantified unless otherwise specified)

Typically objects don't fly:
object(x) & ~ab1(x) ==> ~flies(x)

But birds usually do fly:
bird(x) ==> object(x); bird(x) ==> ab1(x)
bird(x) & ~ab2(x) ==> flies(x)

Except if they're penguins:
penguin(x) ==> bird(x); penguin(x) ==> ab2(x)
penguin(x) & ~ab3(x) ==> ~flies(x)

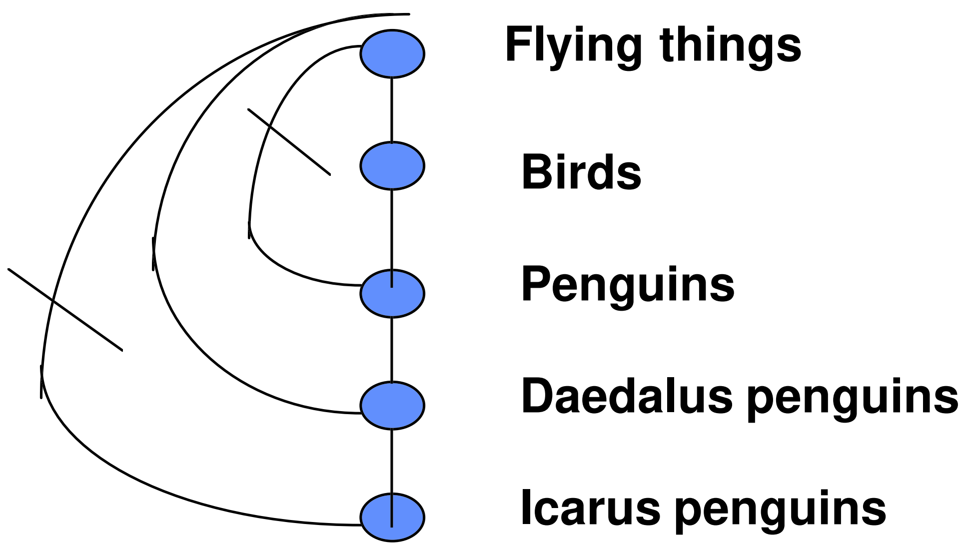
But even penguins can fly if they tie on wings:
Daedalus-penguin(x) ==> penguin(x); Daedalus-penguin(x) ==> ab3(x)
Deadalus-penguin(x) & ~ab4(x) ==> flies(x)

Unless they fly too close to the sun:
Icarus-penguin(x) ==> Daedalus-penguin(s); Icarus-penguin(x) ==> ab4(x)
Icarus-penguin(x) & ~ab5(x) ==> ~flies(x)

And so on ...

The idea is to circumscribe the abnormality predicates

The Tweety example is just a case of inheritance reasoning ...



In fact, inheritance is one of the major applications of nonmonotonic reasoning.
The other major application is nonmonotonic temporal reasoning ...
that is, reasoning about time and action.
But major problems arise.

Nonmonotonic Reasoning and Temporal Reasoning

Goal: Solve problems like
the frame problem (determining which things stay the same when actions are performed)
the ramification problem (determining which things change)
the qualification problem (avoiding the explicit mention of all conditions in a causal rule)
using nonmonotonic logic.

The hope and belief in the late 70s and early 80s:
nonmonotonic logic would make temporal reasoning much easier.

The idea (for the frame problem): Rather than write down lots of frame axioms (or explanation closure axioms, write down one general principe of *inertia*:

Holds(s,fluent) & ~ab(fluent,action,s) ==> Holds(Result(s,action),fluent)

That is, if a fluent holds in a situation, and an action happens that isn't abnormal with respect to that fluent, the fluent still holds in the resulting situation.

Now (we hope) all we have to do is specify when actions are abnormal wrt (i.e. affect) fluents, and we're done! Unfortunately, ...

The Yale Shooting Problem

The Yale Shooting Problem, discovered by Hanks & McDermott in 1985/1986, shows that a straighforward nonmonotonic solution to the frame problem won't work.

Suppose we have the following theory: (With ab circumscribed)

Holds(s,fluent) & ~ab(fluent,action,s) ==> Holds(Result(s,action),fluent) ***Inertia***

Holds(S0,Alive) ***Fred (the turkey) is alive at S0***
Holds(Result(s,load),loaded) ***When you load a gun, it's loaded as a result***
Holds(s,loaded) ==> ab(alive,shoot,s) & Holds(Result(s,shoot),~alive) ***If you shoot a loaded gun at a turkey, it's dead as a result***

Also suppose we define situations as follows:

S1=Result(S0,load)
S2=Result(S1,wait)
S3=Result(S2,shoot)

The question is: is Fred dead or alive in S3?
We expect Fred to be dead, because he's been shot with a loaded gun.
But actually, it turns out that there's some ambiguity.

The Yale Shooting Problem: A Case of Multiple Extensions

Recall the Nixon diamond: Nixon is a Republican and a Quaker; Quakers are typically pacifists; Republicans are typically non-pacifists. We can't conclude anything about Nixon because of the ambiguity. A similar thing happens in the Yale Shooting Problem.

We have two minimal models (where ab is circumscribed)

M1: (the expected model)

S1=Result(S0,load)
S2=Result(S1,wait)
S3=Result(S2,shoot)
Holds(S0,alive)

Holds(S1,loaded)
Holds(S2,loaded)
<u>ab(alive,shoot,S2)</u>
Holds(S3,~alive)

M2: (the unexpected model)

S1=Result(S0,load)
S2=Result(S1,wait)
S3=Result(S2,shoot)
Holds(S0,alive)

Holds(S1,loaded)
<u>ab(loaded,wait,S1)</u>
Holds(S2, ~loaded)
Holds(S3,alive)

In M1, everything goes as planned. In M2, the abnormality happens earlier; somehow, the wait is abnormal to the gun being loaded. Which means that the shoot in S2 isn't abnormal for the turkey being alive.
But note that both models have exactly one ab statement!
Models aren't comparable; neither is a subset of the other; <u>they're both minimal!</u>

The Yale Shooting Problem pointed out that formalizing nonmonotonic temporal reasoning was tricky.
The second, unexpected model seems to be unintuitive, because it goes against our understanding of causation.
How can we fix it? Stay tuned... (next class, set of slides)