Knowledge Representation and Reasoning with First Order Logic

Module 4

Deepak Khemani

The Syllabus

Module 1 (2 hours): Syntax, Semantics, Entailment and Models, Proof Systems, Knowledge Representation.

Module 2 (2 hours): Skolemization, Unification, Deductive Retrieval, Forward Chaining, Backward Chaining

Module 3 (2 hours): Resolution Refutation in FOL, Horn Clauses and Logic Programming

Module 4 (2 hours): Variations on FOL

Text book

Deepak Khemani. A First Course in Artificial Intelligence (Chapters 12 & 13), McGraw Hill Education (India), 2013.

Terms of L(*P*,*F*,*C*)

The basic constituents of *FOL* expressions are *terms*. The set of terms $\Im$ of L(*P*,*F*,*C*) is defined as follows. The constants and the variables are terms by definition. More terms are defined using the function symbols.

If $t \in V$ then $t \in \Im$

If $t \in C$ then $t \in \Im$

If $t_1, t_2, \ldots, t_n \in \Im$ and $f \in F$ is an n-place function symbol then $f(t_1,t_2, \ldots,t_n) \in \Im$

Recap

Atomic Formulas of L(P,F,C)

The set of formulas is defined using terms and predicate symbols. By default the logical symbols "⊥" and "⊤" are also formulas. The set of well formed formulas *F* of L(*P*,*F*,*C*) is defined as follows.

Atomic formulas $\mathcal{A}$

$\bot \in \mathcal{A}$

$\top \in \mathcal{A}$

If $t_1, t_2 \in \Im$ then $(t_1=t_2) \in \mathcal{A}$

If $t_1, t_2, \ldots, t_n \in \Im$ and $P \in$ P is an n-place predicate symbol then $P(t_1,t_2,\ldots,t_n) \in \mathcal{A}$

Recap

Formulas of L(P,F,C)

The set of formulas of L(P,F,C) $\mathcal{F}$ is defined as follows

If $\alpha \in \mathcal{A}$ then $\alpha \in \mathcal{F}$

If $\alpha \in \mathcal{F}$ then $\sim\alpha \in \mathcal{F}$

If $\alpha, \beta \in \mathcal{F}$ then $(\alpha \wedge \beta) \in \mathcal{F}$

If $\alpha, \beta \in \mathcal{F}$ then $(\alpha \vee \beta) \in \mathcal{F}$

If $\alpha, \beta \in \mathcal{F}$ then $(\alpha \supset \beta) \in \mathcal{F}$

Recap

Universal and Existential Quantifiers

If $\alpha \in \mathcal{F}$ and $x \in V$ then $\forall x\ (\alpha) \in \mathcal{F}$

$\forall x\ (\alpha)$ is read as "for all x (α)"

If $\alpha \in \mathcal{F}$ and $x \in V$ then $\exists x\ (\alpha) \in \mathcal{F}$

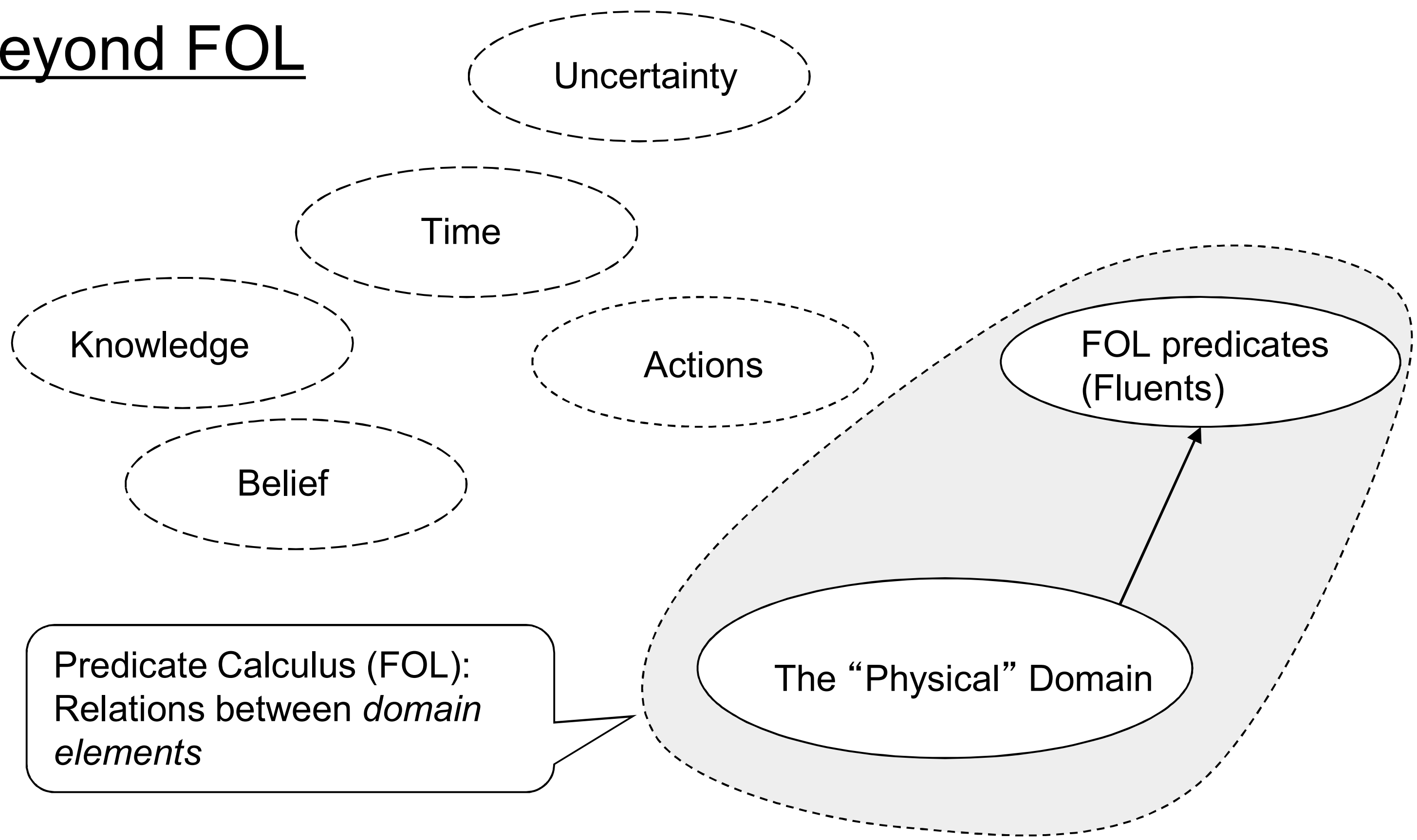
$\exists x\ (\alpha)$ is read as "there exists x (α)"

We will also use the notation (forall (x) (α)) and (exists (x) (α)) as given in the book Artificial Intelligence by Eugene Charniak and Drew McDermott.

Makes representation for use in programs simpler.

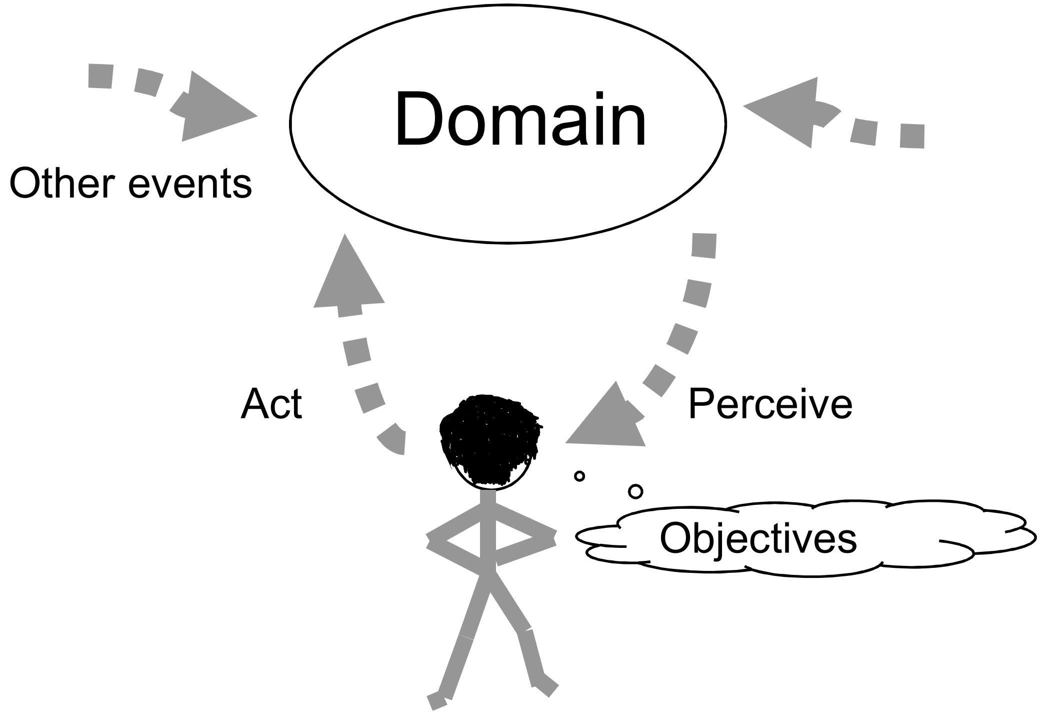
Recap

Beyond FOL



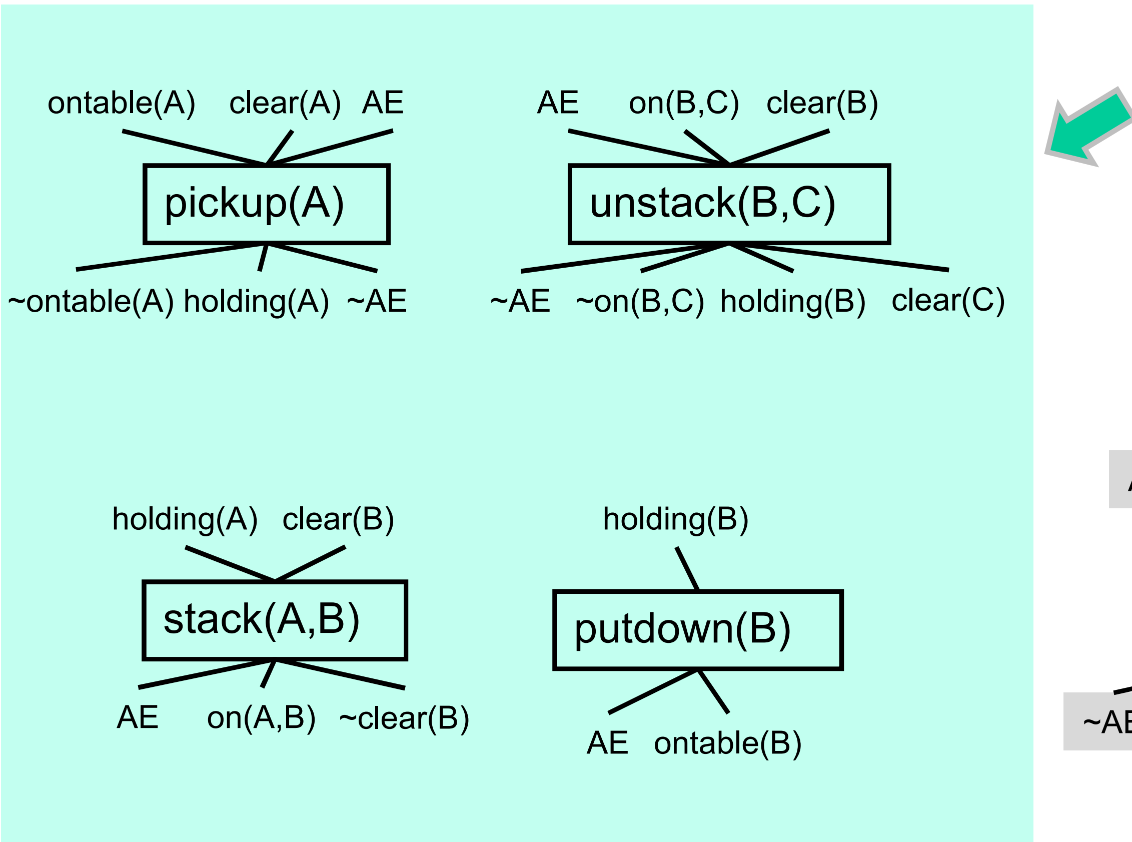
FOL cannot represent and reason about many things.

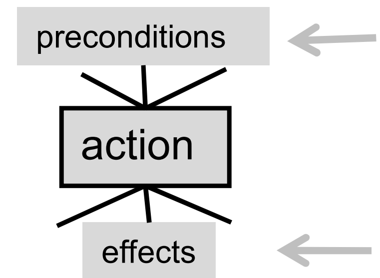
Planning: A Quick Introduction



A planning agent can perceive the world, and produces actions designed to achieve its objectives. In a static domain the agent is the only one who acts. A dynamic domain can be modeled by including other agencies that can change the world.

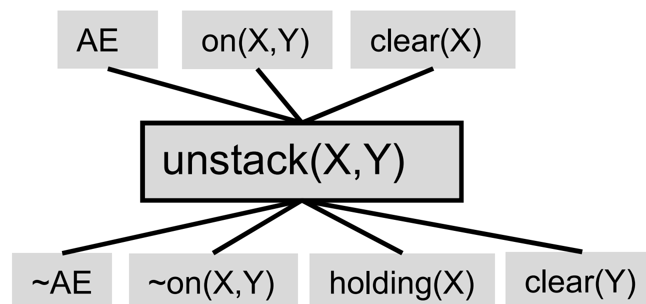
STRIPS Planning Domain: Blocks World



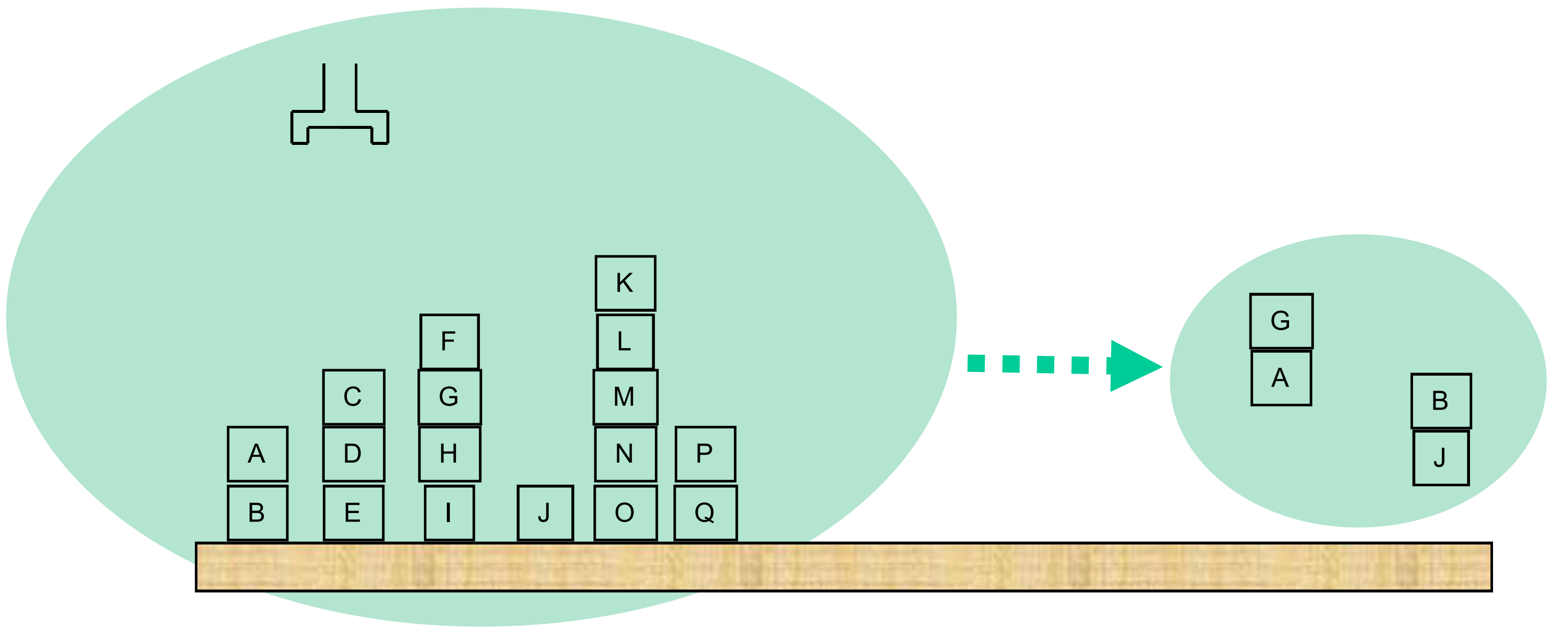


Predicates

ontable(X)
on(X,Y)
clear(X)
holding(X)
AE



State Space Planning



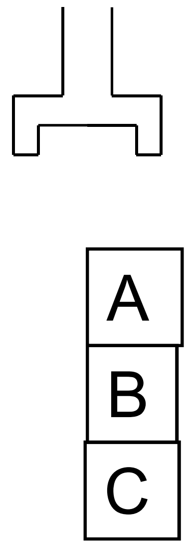
The Given State

The Goal = on(G,A) Λ on(B,J)

Note: The Goal is a partial state description.

Fluents, Events, and Time

Given state {(On A B), (On B C), (not (Maroon A)), (Maroon C)}



The FOL representation does not talk about time.

When is On(A,B) true?

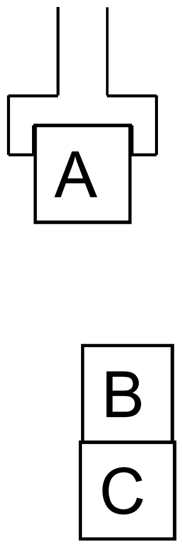
If the world is changing then how do we capture the statements that change in truth value?

Recap

Fluents, Events, and Time

After the *event* Unstack(A,B)

{(Holding A), (On B C), (not (Maroon A)), (Maroon C)}



The FOL representation does not have a notion of time. It is more suited for mathematics where a statement that is true remains true (for ever). For example the Pythagoras Theorem.

After the robot unstacks *A* from *B* On(A,B) is not longer true. We need to add time to statements to indicate when they are true.

We refer to statements (atomic predicates) whose truth value can change with time as *fluents.*

Interpretation 2

{(O A B), (O B C), (not (M A)), (M C)}

Domain: People

When was Anne looking at John?

Is *looking at* an event?

Anne is looking at John

Jack is looking at Anne

John is married

Anne ?

Jack is not married

Recap

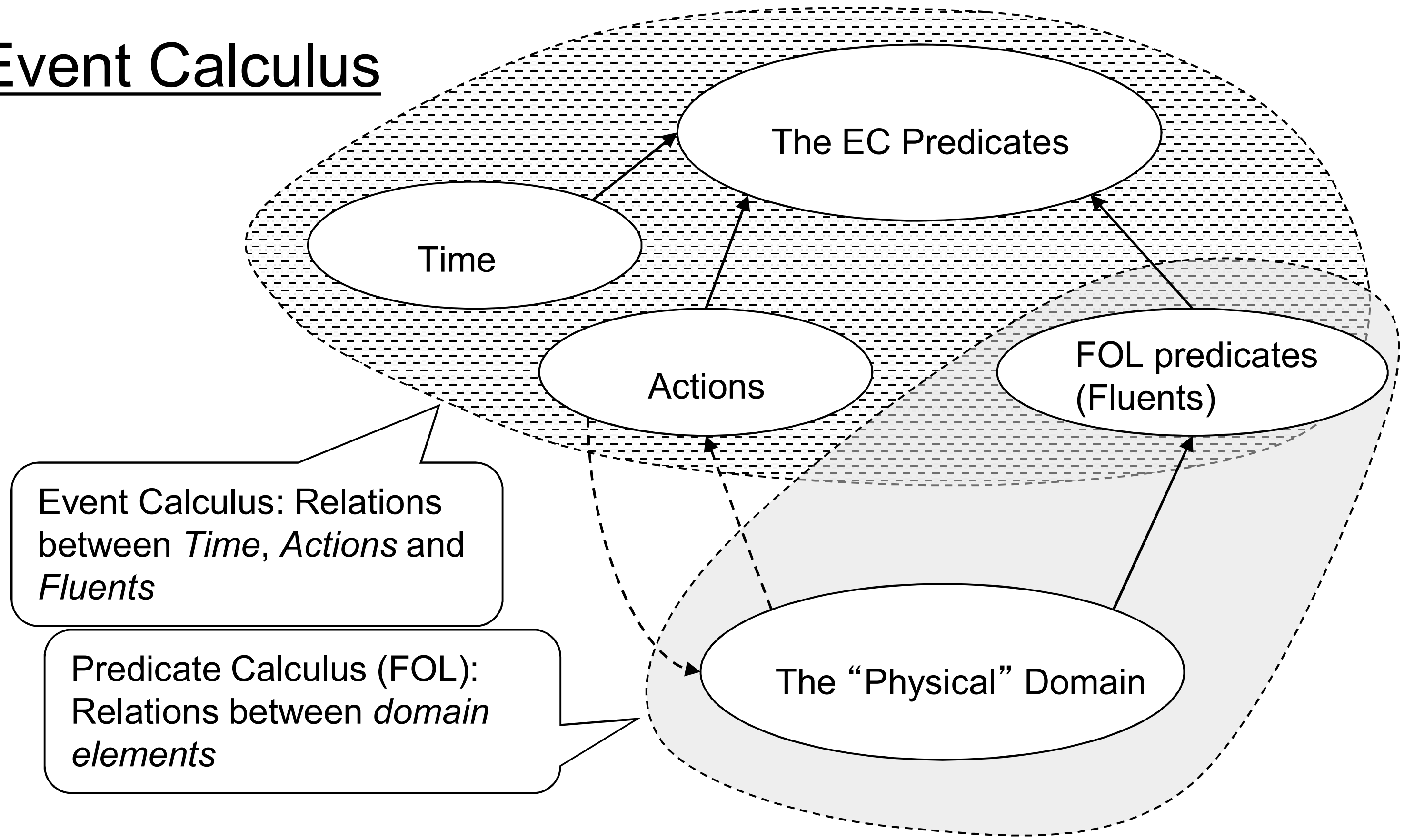
Reification

To reason *about* the effects of actions on the world then we need to be able to treat actions and fluents as *arguments* to predicates that represent relations between them. But arguments to predicates in *FOL* can *only* be terms, and terms are mapped to elements in the domain.

We can circumvent this problem by *extending* the domain to include instances of actions and fluents. That is, for the purpose of reasoning with them, we add the *symbolic representations* of predicates and actions to the domain or the universe of discourse.

We say that we have *reified* the actions and predicates.

Event Calculus



The domain for the *Event Calculus* is *Time*, *Actions* and *Fluents*.

<u>The Event Calculus</u>

Jogesh *made* a cup of tea and *left it* on the table. *Meanwhile* Smita *saw* the cup of tea and *drank it*. *When* Jogesh *came back* he *saw* that the cup *was empty*.

Reasoning about time, action, and change.

Recap

Epistemic reasoning

Jogesh made a cup of tea and left it on the table. Meanwhile Smita saw the cup of tea and drank it. When Jogesh came back he saw that the cup was empty.

He concluded that Smita had polished off his cup of tea.

Smita knew that Jogesh knew that she drank the tea.

Knowledge and belief of agents

Recap

Event Calculus: Variables and Constants

The variables and constants of the *EC* belong to one of the following sorts.

•*an event sort*, with variables {e, e_1, e_2, …}. Includes actions and exogenous events in the specific domain, like *unstack(block2, block6), walk(home22, ramesh23, office34), wakeUp(kumbhakaran1), cyclone(nisha, 2008).*

•*a fluent sort*, with variables {f, f_1, f_2, …}. The predicates from the domain, like *holding(block2), loc(ramesh23, office34), awake(kumbhakaran1).*

•*a timepoint sort* with variables {t, t_1, t_2, …}. In the Continuous Event Calculus (CEC) they may be real numbers, and in Discrete Event Calculus (DEC) they are integers.

The Event Calculus: Predicates-1

- *Happens(e,t_1,t_2)*: Event *e* starts at t_1 and ends at t_2. Observe that the event has a duration. For example, *Happens(Eclipse321,t_5,t_7)* says that a particular eclipse happened between time points t_5 *and* t_7.

- An instantaneous version of Happens can be defined as,
  Happens(e,t) $\stackrel{\text{def}}{=}$ *Happens(e,t,t)*

- *HoldsAt(f,t)*: Fluent *f* is *true* at time point *t.* For example, *HoldsAt(Form(Glacier17, Solid), t_1)* says that at time point t_1 *Glacier17* is in solid form. One may also define a predicate

- *Initally(f)* to assert that fluent *f* is *true* initially.
  Initally(f)$\stackrel{\text{def}}{=}$ *HoldsAt(f,t_0)*

The Event Calculus: Predicates-2

- *Initiates(e,f,t)*: Event *e* occurs at time *t* and results in the fluent *f* becoming *true after t.* For example the event of waking up initiates the fluent of being awake to be *true*, by
  Initiates(wakeup(kumbhakaran1), awake(kumbhakaran1), t_3*).*

  In the DEC it means that the fluent *f* is *true* at time *(t+1)* and later.

- *Terminates(e,f,t):* Event *e* occurs at time *t* and results in the fluent *f* becoming *false after t.*

- For the durative version of the action one can define the fluent to become *true* or *false* at either endpoint. For example, for *Walk(Home, Actor, Office)*, the fluent *AtHome(Actor)* becomes *false* (is terminated) at the start of the walk event, while the fluent *AtOffice(Actor)* becomes *true* (is initiated) at the end.

The Event Calculus: Predicates-3

- *ReleasedAt(f,t)*: The fluent *f* is released from the *commonsense law of inertia* at time *t.*

- The commonsense law of inertia states that a fluent's truth value will not change unless affected - initiated or terminated – by an event. If a fluent is *released* from the commonsense law then it can fluctuate, and one cannot deduce its state.

- Releasing a fluent from the commonsense law is a mechanism to deal with certain kinds of uncertainty. For example, if you toss a coin then you release the fluent *Heads(Coin)* from the commonsense law, and it could take any value.

- *Releases(e,f,t)*: Event *e* occurs at time *t*, after which the fluent *f* is released from the commonsense law of inertia.

The Event Calculus: Predicates-4

- If fluent f_1 is initiated by an event that occurs at time t_1 then fluent f_2 will be *true* at time (t_1+t_2).
- *Trajectory*(f_1,t_1,f_2,t_2): This allows one to capture a causal relation between two fluents. For example, if *On(Stove*) is initiated by *Light(Stove)* at t_1 then the fluent *Temp(Water,SomeIncreasingFn*(t_2)*)* is *true* at time (t_1+t_2).
- *AntiTrajectory*(f_1,t_1,f_2,t_2): If fluent f_1 is terminated by an event that occurs at time t_1 then fluent f_2 will be *true* at time (t_1+t_2).

The Event Calculus: Derived Predicates

Clipped(t_1,f,t_2): A fluent f that was *true* is made *false* sometime after or at time point t_1 and before t_2. This is equivalent to the longer formula,

$$\exists e,t(Happens(e,t) \wedge (t_1 \leq t < t_2) \wedge Terminates(e,f,t))$$

Declipped(t_1,f,t_2): A fluent f that was *false* is made *true* sometime after or at time point t_1 and before t_2. This is equivalent to the longer formula,

$$\exists e,t(Happens(e,t) \wedge (t_1 \leq t < t_2) \wedge Initiates(e,f,t))$$

PersistsBetween(t_1,f,t_2): The fluent f is not released from the commonsense law of inertia after time point t_1 and up to and including time point t_2. That is, it retains its truth value during the interval. This is a short form for,

$$\neg\exists t(ReleasedAt(f,t) \wedge (t_1 < t \leq t_2))$$

EC: The Effect of Events on Fluents

Note: All formulas are universally quantified

EC_1: $(Happens(e,t) \wedge Initiates(e,f,t)) \supset HoldsAt(f,t)$

Likewise if an event happens that terminates a fluent the fluent ceases to hold when it happens.

EC_2: $(Happens(e,t) \wedge Terminates(e,f,t)) \supset \neg HoldsAt(f,t)$

Events may release a fluent from the commonsense law of inertia, or they may terminate their released status.

EC_3: $(Happens(e,t) \wedge Releases(e,f,t)) \supset ReleasedAt(f,t)$

EC_4: $(Happens(e,t) \wedge (Initiates(e,f,t) \vee Terminates(e,f,t)) \supset \neg ReleasedAt(f,t)$

The Inertia Axiom (IA)

Using the above definitions we can infer that the value of a fluent remains the same if it remains under the commonsense law of inertia *and* is not clipped by some event.

IA: $(HoldsAt(f,t_1) \wedge (t_1<t_2) \wedge PersistsBetween(t_1,f,t_2) \wedge \neg Clipped(t_1,f,t_2)) \supset HoldsAt(f,t_2)$

Note: All formulas are universally quantified

The Frame Problem

https://plato.stanford.edu/entries/frame-problem/

The frame problem is the challenge of representing the effects of action in logic without having to represent explicitly a large number of intuitively obvious non-effects.

Initiates(*Paint*(*x*, *c*), *Colour*(*x*, *c*), *t*)
Initiates(*Move*(*x*, *p*), *Position*(*x*, *p*), *t*)
HoldsAt(*Colour*(*A*, *Red*), *1)*
HoldsAt(*Position*(*A*, *House*),*1)*
Happens*(Paint*(*A*, *Blue*), 2)
Happens*(Move*(*A*, *Garden*), 2)

What is true at *time* 4? Does Move affect colour?

The Yale Shooting Problem (Hanks & McDermott, 1987)

There are three types of action — *Load, Sneeze, Shoot*
Three fluents — *Loaded, Alive, Dead*

Initiates(Load,Loaded,t)	*(Y1.1)*
Initiates(Shoot,Dead,t) $\leftarrow$ *HoldsAt(Loaded,t)*	*(Y1.2)*
Terminates(Shoot,Alive,t) $\leftarrow$ *HoldsAt(Loaded,t)*	*(Y1.3)*
InitiallyP(Alive)	*(Y2.1)*
Happens(Load,T1)	*(Y2.2)*
Happens(Sneeze,T2)	*(Y2.3)*
Happens(Shoot,T3)	*(Y2.4)*
$T1 < T2$	*(Y2.5)*
$T2 < T3$	*(Y2.6)*
$T3 < T4$	*(Y2.7)*

Now let Σ be the conjunction of (Y1.1) to (Y1.3),
and let Δ be the conjunction of (Y2.1) to (Y2.7

The Yale Shooting Problem (Hanks & McDermott, 1987)

Now let Σ be the conjunction of (Y1.1) to (Y1.3),
and let Δ be the conjunction of (Y2.1) to (Y2.7)

Let SC be a Simple Calculus

$$HoldsAt(f,t) \quad InitiallyP(f) \wedge \neg\, Clipped(0,f,t) \qquad (SC1)$$

$$HoldsAt(f,t2) \leftarrow Happens(a,t1) \wedge Initiates(a,f,t1) \wedge t1 < t2 \wedge \neg Clipped(t1,f,t2) \qquad (SC2)$$

$$Clipped(t1,f,t2) \leftrightarrow \exists\, a,t\ [Happens(a,t) \wedge t1 < t < t2 \wedge Terminates(a,f,t)] \qquad (SC3)$$

Does the following hold?

$$\Sigma \wedge \Delta \wedge SC \vDash HoldsAt(Dead,T4).$$

The Yale Shooting Problem (Hanks & McDermott, 1987)

Does the following hold?

$$\Sigma \wedge \Delta \wedge SC \vDash \mathit{HoldsAt(Dead,T4)}.$$

Unfortunately this sequent is not valid. We have not described explicitly the non-effects of actions. In particular, we haven't said that the Sneeze action doesn't unload the gun. So there are, for example, models of SC ∧ Σ ∧ Δ in which Terminates(Sneeze,Loaded,T2) is true, Holds(Alive,T4) is true, and HoldsAt(Dead,T4) is false.

In addition to describing the non-effects of actions, we must describe the non-occurrence of actions. And, more trivially, we must include formulae that rule out the possibility that, say, the Sneeze action and the Shoot action are identical.

UNA[Load, Sneeze, Shoot] (Y3.1)
UNA[Loaded, Alive, Dead] (Y3.2)

A Circumscriptive Solution to the Frame Problem

The idea of circumscription is to minimise the extensions of certain named predicates. That is to say, the circumscription of a formula Φ yields a theory in which these predicates have the smallest extension allowable according to Φ. The circumscription of Φ minimising the predicate ρ is written, CIRC[Φ ; ρ].

This is equivalent to the following second-order formula.

$$\Phi \wedge \neg \exists q \, [\Phi(q) \wedge q < \rho] \text{ where,}$$

- $q = \rho$ means $\forall x \, [q(x) \leftrightarrow \rho(x)]$,
- $q \leq \rho$ means $\forall x \, [q(x) \rightarrow \rho(x)]$,
- $q < \rho$ means $[q \leq \rho] \wedge \neg [q = \rho]$, and
- $\Phi(q)$ is the formula obtained by replacing all occurrences of ρ in Φ by q.

Circumscription on the Yale Shooting problem

Given,

- a conjunction Σ of Initiates and Terminates formulae,
- a conjunction ∆ of InitiallyP, Happens and temporal ordering formulae, and
- a conjunction Ω of uniqueness-of-names axioms for actions and fluents,

we're interested in, CIRC[Σ ; Initiates, Terminates] ∧ CIRC[∆ ; Happens] ∧ SC ∧ Ω.

The minimisation of Initiates and Terminates corresponds to the default assumption that actions have no unexpected effects, and the minimisation of Happens corresponds to the default assumption that there are no unexpected event occurrences.

Let Σ be the conjunction of (Y1.1) to (Y1.3), and let ∆ be the conjunction of (Y2.1) to (Y2.7). We have,

CIRC[Σ ; Initiates, Terminates] ∧ CIRC[∆ ; Happens] ∧ SC ∧ Ω ⊨ HoldsAt(Dead,T4).

Problems with the Universal Quantifier

One facet of reasoning under uncertainty is known as default reasoning. This involves making inferences that are *plausible* or *likely* but not necessarily entailed by the knowledge base. The need for default reasoning arises because of our desire to generalize connections between categories, to express them in a succinct manner. A universal statement like the one below is simply not adequate.

$$\forall x\ (Bird(x) \supset Flies(x))$$

The moment we come up with an exception for example a bird called Peppy who cannot fly being a penguin,

$$Bird(peppy) \wedge Penguin(peppy) \wedge \neg Flies(peppy)$$

our knowledge base becomes unsatisfiable

Default Reasoning

Instead we would like to have a mechanism by which given a knowledge base we can make the set of *plausible inferences*, with the caveat that if the knowledge base grows then some of the inferences may not hold.

This implies that the set of inferences that we can make does not grow monotonically with what we know, and could in fact become smaller when we add more facts.

This form of reasoning is called *non-monotonic reasoning*, because the set of inferred sentences does not grow monotonically with the set of known facts.

Circumscription

Circumscription, devised by John McCarthy, is an approach that aims to minimize the extent of only some predicates (McCarthy, 1980; 1986), (Lifschitz, 1985; 1994). Traditionally these predicates characterize *abnormality* with respect to the intended default inference, but circumscription itself can be done over any set of specified predicates.

The solution to this problem as proposed by McCarthy adds another clause to the antecedent saying that in addition to be being birds the individual should not be abnormal. This clause is intended to catch the abnormal cases.

$$\forall x\ (Bird(x) \land \neg Ab(x) \supset Flies(x))$$

Default reasoning with circumscription aims to minimize the extent of the abnormality predicates.

Circumscription

Let $\vartheta_1(D, I_1)$ and $\vartheta_2(D, I_2)$ be two interpretations that agree on all constants and functions of the language. We define the relation $\leq$ as follows,

$\vartheta_1 \leq \vartheta_2$ iff for every predicate *P* being circumscribed $I_1(P) \subseteq I_2(P)$

And, $\vartheta_1 < \vartheta_2$ iff $\vartheta_1 \leq \vartheta_2$ and $\vartheta_2 \not\leq \vartheta_1$.

We can now define entailment $\models_{\leq}$ under circumscription as,

KB $\models_{\leq} \alpha$ iff for every interpretation ϑ such that $\vartheta \models$KB either KB $\models \alpha$ or there is an interpretation ϑ' such that $\vartheta' < \vartheta$ and $\vartheta' \models$KB.

If the predicate being circumscribed is *Ab* then we can also say equivalently that,

KB $\models_{\leq} \alpha$ iff Circ[KB; Ab] $\models \alpha$

An illustrative example

Consider the knowledge base ,

$$KB = \{\forall x\ (Bird(x) \land \neg Ab(x) \supset Flies(x)),\ Bird(tweety),\ Bird(chilly),\ chilly \neq tweety,\ \neg Flies(chilly)\}$$

Is it reasonable to conclude that Tweety can fly?

$$KB \vDash_{\leq} Flies(tweety)?$$

In Circumscription we minimize *Ab* predicate, and then consider normal entailment in the minimal model.

An illustrative example

Observe that ,

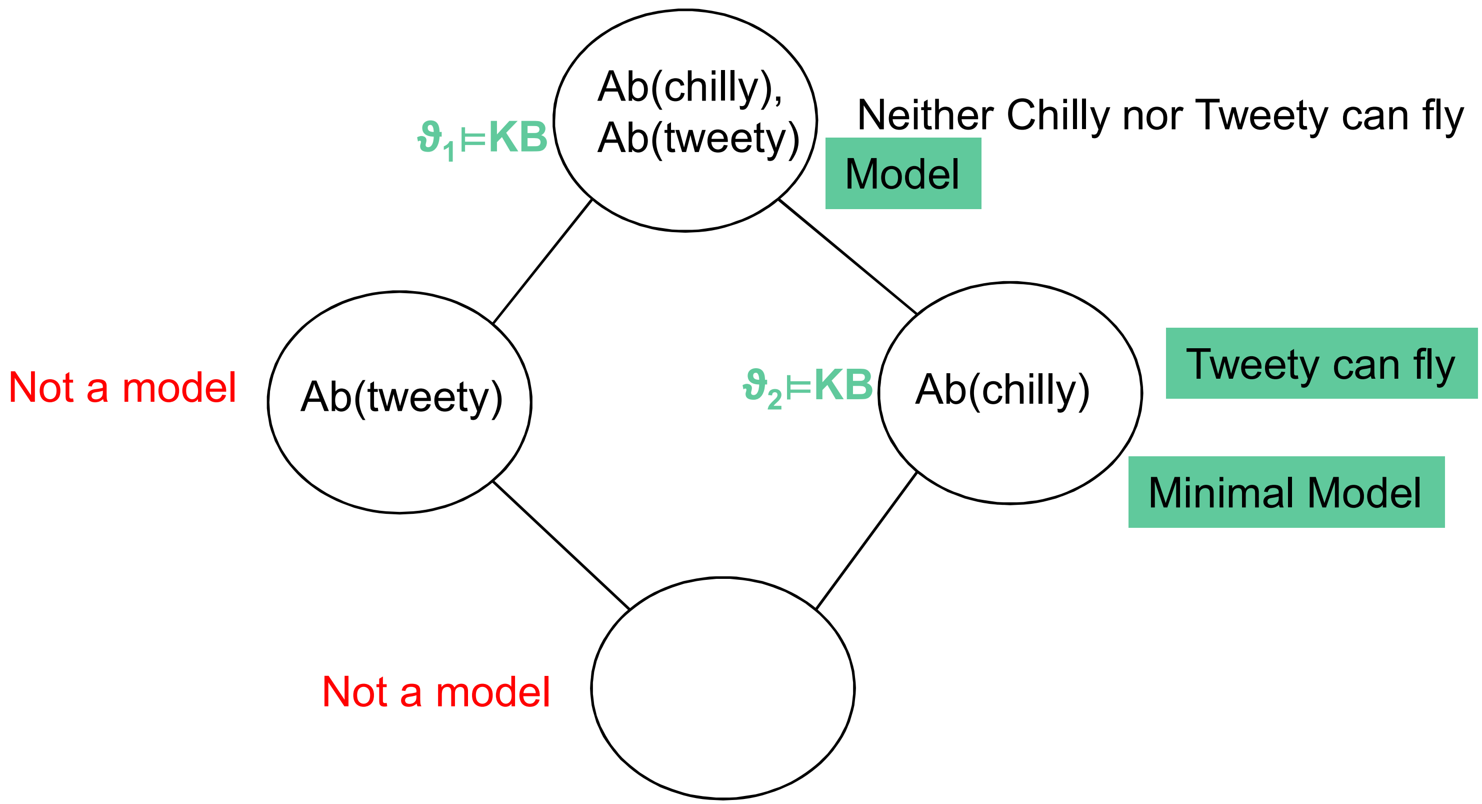
$\forall x\ (Bird(x) \wedge \neg Ab(x) \supset Flies(x)) \equiv \forall x\ (Bird(x) \wedge \neg Flies(x) \supset Ab(x))$

Then given that Bird(chilly) $\wedge$ $\neg$Flies(chilly) we can conclude Ab(chilly).

Therefore in the minimal model Ab(chilly) must be present
and therefore $\neg$Ab(tweety) is true in the minimal model.

Therefore, KB $\vDash_{\leq}$ Flies(tweety?

$KB \vDash_{\leq} Flies(tweety)$



Another illustrative example

Consider the knowledge base ,

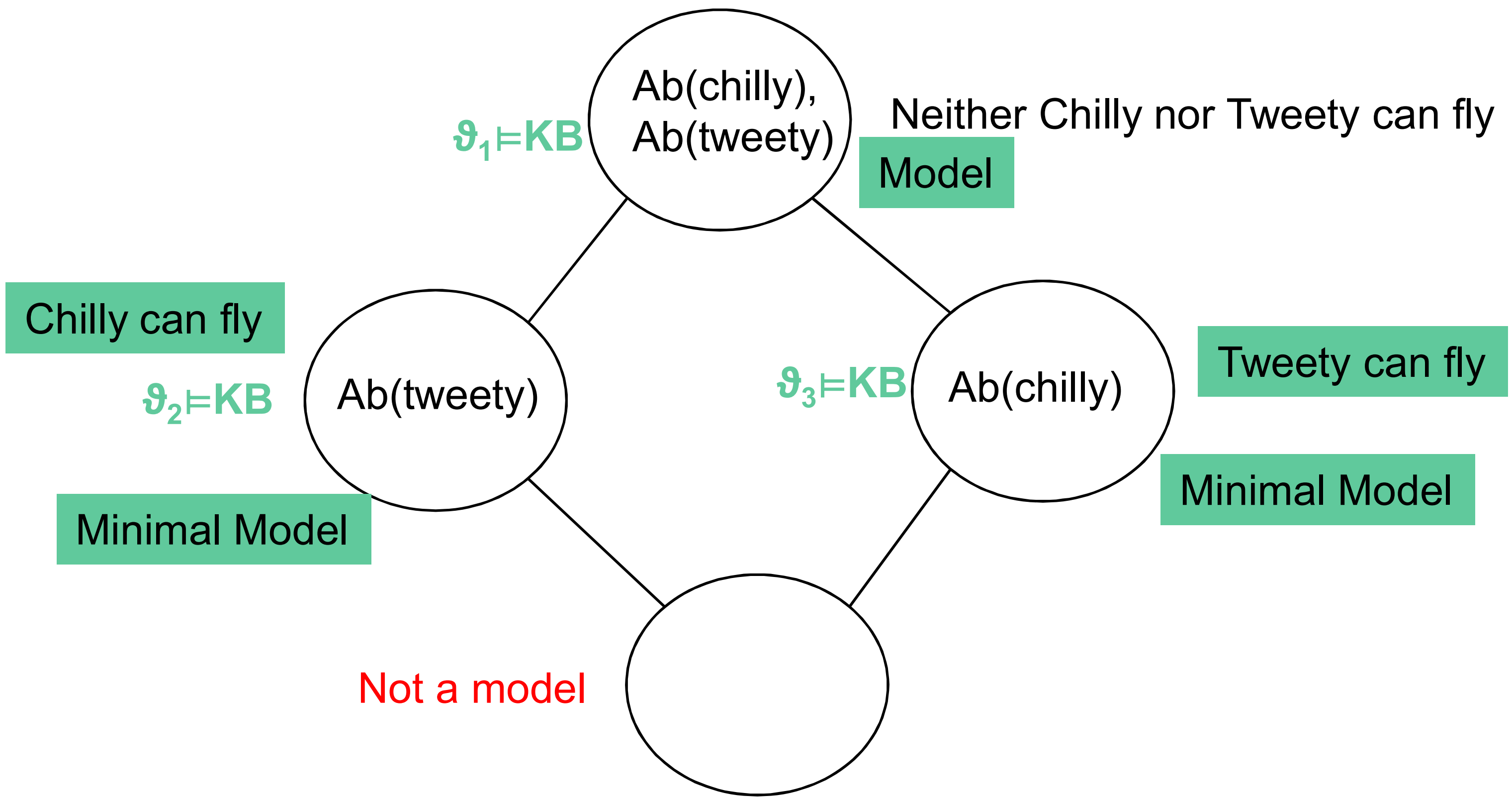
$$KB = \{\forall x\ (Bird(x) \wedge \neg Ab(x) \supset Flies(x)),\ Bird(tweety),\ Bird(chilly),$$
$$chilly \neq tweety,\ (\neg Flies(tweety) \vee \neg Flies(chilly))\}$$

Is it reasonable to conclude that Tweety or Chily can fly?

$$KB \models_{\leq} (Flies(tweety) \vee Flies(chilly))\ ?$$

In Circumscription we minimize *Ab* predicate, and then consider normal entailment in the minimal model.

$KB \models_{\leq} (Flies(tweety) \lor Flies(chilly))$



End of Module 4