# Planning and search

## FOL and situation calculus

# Outline

◇ FOL continued

◇ Situation calculus

◇ Logic and planning

# Equality

$term_1 = term_2$ is true under a given interpretation
if and only if $term_1$ and $term_2$ refer to the same object

E.g., $1 = 2$ and $\forall x \;\; \times(Sqrt(x), Sqrt(x)) = x$ are satisfiable
$2 = 2$ is valid

E.g., definition of (full) $Sibling$ in terms of $Parent$:

$$\forall x, y \;\; Sibling(x,y) \;\Leftrightarrow\; [\neg(x = y) \land \exists m, f \;\; \neg(m = f) \land Parent(m,x) \land Parent(f,x) \land Parent(m,y) \land Parent(f,y)]$$

# Interacting with FOL KBs

Suppose a wumpus-world agent is using an FOL KB
and perceives a smell and a breeze (but no glitter) at $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$
$Ask(KB, \exists\, a \;\; BestAction(a, 5))$

I.e., does $KB$ entail any particular actions at $t = 5$? For example,

Answer: $Yes,\; \{a/Shoot\}$ $\quad\leftarrow$ substitution (binding list)

Given a sentence $S$ and a substitution $\sigma$,
$S\sigma$ denotes the result of plugging $\sigma$ into $S$; e.g.,
$S = Smarter(x, y)$
$\sigma = \{x/Hillary, y/Bill\}$
$S\sigma = Smarter(Hillary, Bill)$

$Ask(KB, S)$ returns some/all $\sigma$ such that $KB \models S\sigma$

# Knowledge base for the wumpus world

"Perception"
$\forall\, b, g, t \;\; Percept([Smell, b, g], t) \Rightarrow Smelt(t)$
$\forall\, s, b, t \;\; Percept([s, b, Glitter], t) \Rightarrow AtGold(t)$

Reflex: $\forall\, t \;\; AtGold(t) \Rightarrow Action(Grab, t)$

Reflex with internal state: do we have the gold already?
$\forall\, t \;\; AtGold(t) \land \neg Holding(Gold, t) \Rightarrow Action(Grab, t)$

$Holding(Gold, t)$ cannot be observed
$\Rightarrow$ keeping track of change is essential

# Deducing hidden properties

Properties of locations:
$\forall x, t \;\; At(Agent, x, t) \land Smelt(t) \;\Rightarrow\; Smelly(x)$
$\forall x, t \;\; At(Agent, x, t) \land Breeze(t) \;\Rightarrow\; Breezy(x)$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

$$\forall y \;\; Breezy(y) \;\Rightarrow\; \exists x \;\; Pit(x) \land Adjacent(x, y)$$

Causal rule—infer effect from cause

$$\forall x, y \;\; Pit(x) \land Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Neither of these is complete—e.g., the causal rule doesn't say whether squares far away from pits can be breezy

Definition for the $Breezy$ predicate:

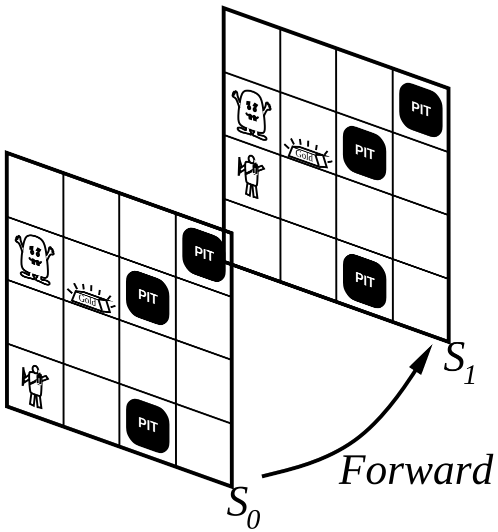
$$\forall y \;\; Breezy(y) \;\Leftrightarrow\; [\exists x \;\; Pit(x) \land Adjacent(x, y)]$$

# Keeping track of change

Facts hold in situations, rather than eternally
E.g., $Holding(Gold, Now)$ rather than just $Holding(Gold)$

Situation calculus is one way to represent change in FOL:
Adds a situation argument to each non-eternal predicate
E.g., $Now$ in $Holding(Gold, Now)$ denotes a situation

Situations are connected by the $Result$ function
$Result(a, s)$ is the situation that results from doing $a$ in $s$



# Time points vs situations

$\forall x \forall y \forall t \forall x_1 \forall y_1 \forall z$

$At(x, y, z, t) \wedge MovedTo(x_1, y_1, z, t) \;\Rightarrow\; At(x_1, y_1, z, t+1))$

(if some $z$ is located at coordinates ($x$, $y$) at time $t$, and at $t$ it is moved to ($x_1$,$y_1$), then at the next moment it is located at ($x_1, y_1$)).

However this does not distinguish between actions and static properties of the state, and identifies a 'state of the world' with a time point.

# Situation calculus

Situation calculus is a dialect of FOL where situations (static states of the world) and actions are basic terms:

variables over situations are denoted $s, s_1, s_2, \ldots,$.

a distinguished initial situation is denoted by a constant $S_0$.

actions are terms like $move(x, y, z)$ (move thing $z$ to coordinates $x, y$) etc. Note that actions are also terms, not formulas: they denote an 'action' and are not true or false.

a special function $Result$ takes an action and a situation and returns a new situation: $Result(a, s)$ denotes a new situation which results from performing an action $a$ in a situation $s$.

# Fluents

Predicates and functions whose values vary from situation to situation are called fluents.

Last argument in a fluent is a situation:

$\neg Holding(r, x, s) \land Holding(r, x, Result(pickup(r, x), s))$

(a robot $r$ is not holding $x$ in $s$ but is holding it in the situation resulting from $s$ by picking $x$ up).

A distinguished fluent $Poss$ says which actions are possible in a given situation:

$Poss(pickup(r, blockA), S_0)$

# Preconditions of actions

A precondition is a condition which makes an action possible.

It can be expressed by a formula, sometimes called precondition axiom.

For example:

$$\forall r \forall x \forall s\, Poss(pickup(r,x),s) \;\; \Leftrightarrow \;\; \forall z(\neg Holding(r,z,s) \wedge \neg Heavy(x) \wedge NextTo(r,x,s))$$

(a robot can pick $x$ up if it is not holding anything else, $x$ is not heavy, and the robot is next to it).

# Postconditions or effects of actions

A postcondition or effect of an action is a change resulting from executing the action.

Formulas expressing postconditions are sometimes called effect axioms.

For example, $\forall x \forall s \forall r (Fragile(x) \Rightarrow Broken(x, Result(drop(r, x), s)))$

Effect axioms for fluents which become true as a result of an action are called positive, and those where the fluent becomes false are called negative.

# Frame axioms

We are considering a very simple world: actions have clearly predefined effects (no non-determinism), and the world changes only as a result of clearly specified actions.

For every action, we can also say which fluents it **does not** affect.

The formulas which specify which properties are not changed as a result of an action are called frame axioms.

For example,

$\forall x \forall y \forall s \forall r$

$\neg Broken(x, s) \land (x \neq y \lor \neg Fragile(x)) \Rightarrow \neg Broken(x, Result(drop(r, y), s))$

Frame axioms do not logically follow from precondition and effect axioms.

They are called frame axioms because they limit or frame the effects of actions.

# Why do we need frame axioms

A typical kind of task in reasoning about actions is to check whether

(1) a certain sequence of actions $a_1, \ldots, a_n$ will succeed (bring about some desired state of the world)

(2) a certain sequence of actions is possible

In both cases, some relevant information about $S_0$ is given (which fluents hold in $S_0$).

The precondition and effects of actions are used to determine which fluents will be true in $Result(a_n, Result(a_{n-1}, \ldots Result(a_1, S_0) \ldots)$.

Some fluent may be a precondition of some action $a_i$ which is true in $S_0$ and is unchanged by $a_1, \ldots, a_{i-1}$.

However we cannot derive that it is unchanged from just the precondition and effect axioms for $a_1, \ldots, a_{i-1}$: need to also have explicit frame axioms.

# Frame problem

Frame problem is the problem of representing frame conditions coincisely (**not** with an axiom for each pair of action and fluent!).

# Solution to the frame problem

For each fluent $F(\bar{x}, s)$ (where $\bar{x}$ are all the free variables of the fluent) we collect together all positive effect axioms. For example, if $Broken(x, s)$ has two positive effect axioms:

$$\forall x \forall s\ Fragile(x) \Rightarrow Broken(x, Result(drop(x), s))$$

$$\forall x \forall s\ Broken(x, Result(break(x), s))$$

then together they can be written as:

$$\forall x \forall a \forall s (Fragile(x) \wedge a = drop(x)) \vee (a = break(x)) \Rightarrow Broken(x, Result(a, s))$$

In general, have an expression $\forall \bar{x} \forall a \forall s (\Pi_F(\bar{x}, a, s) \Rightarrow F(\bar{x}, Result(a, s)))$

# Solution to the frame problem continued

Same for the negative effect axioms:

$\forall \bar{x} \forall a \forall s (N_F(\bar{x}, a, s) \Rightarrow \neg F(\bar{x}, Result(a, s)))$

For example:

$\forall \bar{x} \forall a \forall s (a = fix(x) \Rightarrow \neg Broken(x, Result(a, s)))$

# Solution to the frame problem continued

Once we have a single formula $\Pi_F$ for all actions which make $F(x, s)$ true and a single formula $N_F$ for all actions which make $F$ false, we can write explanation closure axioms:

$$\forall \bar{x} \forall a \forall s (\neg F(\bar{x}, s) \wedge F(\bar{x}, Result(a, s)) \Rightarrow \Pi_F(\bar{x}, a, s))$$

$$\forall \bar{x} \forall a \forall s (F(\bar{x}, s) \wedge \neg F(\bar{x}, Result(a, s)) \Rightarrow N_F(\bar{x}, a, s))$$

They **replace all frame axioms** by saying that

$F$ only becomes true if $\Pi_F$ holds (only certain actions in certain circumstances make $F$ true)

$F$ only becomes false if $N_F$ is true

$\Pi_F$ and $N_F$ are short, and explanation axioms entail all the frame axioms (so the frame problem solved - for this simple case anyway).

One of the assumptions is that all actions are deterministic.

# Successor state axioms

If some additional assumptions hold, namely:

(1) no action has both a positive and negative effect on a fluent $F$,

(2) action terms can only be equal if they are the same action name applied to the same arguments

then explanation closure axioms can be combined into a successor state axiom for a fluent: $\forall \bar{x} \forall a \forall s (F(\bar{x}, Result(a, s)) \Leftrightarrow \Pi_F(\bar{x}, a, s) \vee (F(\bar{x}, s) \wedge \neg N_F(\bar{x}, a, s)))$

Under those assumptions, all that is needed to solve the frame problem and describe the actions and fluents completely are: precondition axioms and successor state axioms

# Summary: Describing actions I

"Effect" axiom—describe changes due to action

$\forall\, s \;\; AtGold(s) \;\Rightarrow\; Holding(Gold, Result(Grab, s))$

"Frame" axiom—describe **non-changes** due to action

$\forall\, s \;\; HaveArrow(s) \;\Rightarrow\; HaveArrow(Result(Grab, s))$

Frame problem: find an elegant way to handle non-change

(a) representation—avoid frame axioms

(b) inference—avoid repeated "copy-overs" to keep track of state

Qualification problem: true descriptions of real actions require endless caveats—what if gold is slippery or nailed down or . . .

Ramification problem: real actions have many secondary consequences—what about the dust on the gold, wear and tear on gloves, . . .

# Summary: Describing actions II

Successor-state axioms solve the representational frame problem

Each axiom is "about" a **predicate** (not an action per se):

P true afterwards $\Leftrightarrow$ [an action made P true
$\vee$ P true already and no action made P false]

For holding the gold:

$$\forall\, a, s \;\; Holding(Gold, Result(a, s)) \;\Leftrightarrow$$
$$[(a = Grab \wedge AtGold(s))$$
$$\vee\, (Holding(Gold, s) \wedge a \neq Release)]$$

# Making plans

Initial condition in KB:

$At(Agent, [1,1], S_0)$
$At(Gold, [1,2], S_0)$

Query: $Ask(KB, \exists\, s\ \ Holding(Gold, s))$
i.e., in what situation will I be holding the gold?

Answer: $\{s/Result(Grab, Result(Forward, S_0))\}$
i.e., go forward and then grab the gold

# Making plans: A better way

Represent plans as action sequences $[a_1, a_2, \ldots, a_n]$

$PlanResult(p, s)$ is the result of executing $p$ in $s$

Then the query $Ask(KB, \exists\, p\ \ Holding(Gold, PlanResult(p, S_0)))$
has the solution $\{p/[Forward, Grab]\}$

Definition of $PlanResult$ in terms of $Result$:

$$\forall\, s\ \ PlanResult([\,], s) = s$$
$$\forall\, a, p, s\ \ PlanResult([a|p], s) = PlanResult(p, Result(a, s))$$

Planning systems are special-purpose reasoners designed to do this type of inference more efficiently than a general-purpose reasoner

# Summary

First-order logic:
- objects and relations are semantic primitives
- syntax: constants, functions, predicates, equality, quantifiers

Increased expressive power: sufficient to define wumpus world

Situation calculus:
- conventions for describing actions and change in FOL
- can formulate planning as inference on a situation calculus KB

Next lecture: Planning. Russell and Norvig, 3rd ed., Chapter 10.1-10.2.