Beliefs in a changing world – actions and planning

<u>Situation calculus</u> is a variant of FOL used to represent beliefs about a changing world.

Situations and actions are represented as objects in the domain.

-actions like put(r,x,y) – robot r puts object x on top of object y.

-situations – denote possible world histories

– there are two special symbols used:

-the constant S_0 – representing the initial situation

-the function symbol do(a,s) – represents the new situation resulting from performing action *a* in situation s.

Example:

do(pickup(b_2),do(pickup(b_1),S_0)) – the situation that results from picking up object b_1 in S_0 and then picking up object b_2.

Beliefs in a changing world – actions and planning

Def. Fluents are predicates and functions whose values may vary from situation to situation.

By convention, the last argument of a fluent is a situation.

Holding(r,x,s) – robot r holding object x in situation s

¬Holding(r,x,s) ∧ Holding(r,x,do(pickup(r,x),s))

Obs. There is no special symbol for the “current” situation. s can be different situations in the past, present or future.

A special predicate Poss(a,s) is used to state that action *a* can be performed in situation s:

Poss(pickup(r,x),S_0) – in the initial situation, the robot is able to pick up object x.

Beliefs in a changing world – actions and planning

Axioms (special formulas)

Actions have preconditions and effects.

Precondition axioms are conditions that need to be true for the action to occur.

Example: a robot can pick an object iff it is not holding anything, the object is not too heavy and the robot is next to the object:

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

Effect axioms are fluents that are changed as a result of performing the action.

$$Fragile(x) \supset Broken(x,do(drop(r,x),s))$$

$$\neg Broken(x,do(repair(r,x),s))$$

Effect axioms are called positive if they describe when a fluent becomes true, and negative otherwise.

Beliefs in a changing world – actions and planning

Frame axioms

If a fluent is not mentioned in an effect axiom for an action *a*, we don't know anything about it in the situation do(a,s).

Frame axioms limit (or frame) the effect of actions. It is necessary to know what fluents are unaffected by performing an action.

Example:

$$Color(x,c,s) \supset Color(x,c,do(drop(r,x),s))$$

$$\neg Broken(x,s) \wedge [x \neq y \vee \neg Fragile(x)] \supset \neg Broken(x,do(drop(r,y),s))$$

Beliefs in a changing world – actions and planning

For any given fluent, we expect that only a small number of actions affect the value of that fluent.

<u>The frame problem</u> – to reason effectively with an extremely large number of frame axioms.

Example: an object's color is unaffected by numerous actions: picking it up, moving it, opening a door, going to the beach…

All these require frame axioms.

Beliefs in a changing world – actions and planning

Solution

Suppose that the KB contains all the relevant effect axioms. That is, for each fluent $F(\vec{x},s)$, $\vec{x}=(x_1,\ldots,x_n)$ and action *a* that can cause the fluent to change, we have an effect axiom of the form:

$$\Phi(\vec{x},s) \supset (\neg)F(\vec{x},do(a,s)),$$

where $\Phi(\vec{x},s)$ is some condition on situation s.

We want a procedure to generate all the frame axioms from these effect axioms.

Beliefs in a changing world – actions and planning

First, put all effect axioms into a normal form.

For any fluent $F(\vec{x},s)$, we rewrite all the positive effect axioms as a single formula

positive effect conditions for F

$$\Pi_F(\vec{x},a,s) \supset F(\vec{x},do(a,s)), \qquad (1)$$

and all the negative effects axioms as

$$N_F(\vec{x},a,s) \supset \neg F(\vec{x},do(a,s)). \qquad (2)$$

negative effect conditions for F

Beliefs in a changing world – actions and planning

For example, for the two positive effect axioms for Broken:

$$\mathrm{Fragile}(x) \supset \mathrm{Broken}(x,do(drop(r,x),s)),$$

$$\mathrm{NextTo}(b,x,s) \supset \mathrm{Broken}(x,do(explode(b),s)),$$

we get a single formula

$$\exists r\{a=drop(r,x) \wedge \mathrm{Fragile}(x)\} \vee$$
$$\exists b\{a=explode(b) \wedge \mathrm{NextTo}(b,x,s)\} \supset \mathrm{Broken}(x, do(a,s)).$$

Similarly, a negative effect axiom like

$$\neg\mathrm{Broken}(x,do(repair(r,x),s))$$

can be rewritten as

$$\exists r\{a=repair(r,x)\} \supset \neg\mathrm{Broken}(x, do(a,s)).$$

Beliefs in a changing world – actions and planning

Explanation closure axioms

Assume that formulas 1) and 2) characterize all the conditions under which an action a changes the value of the fluent F (that is a completeness assumption – allows us to conclude that actions that are not mentioned explicitly in effect axioms leave the fluent invariant).

We can express these assumptions by using the explanation closure axioms:

$$\neg F(\vec{x},s) \wedge F(\vec{x},do(a,s))) \supset \Pi_F(\vec{x},a,s) \qquad (3)$$

(if F were false, and made true by doing action a, then condition Π_F must have been true)

$$F(\vec{x},s) \wedge \neg F(\vec{x},do(a,s))) \supset N_F(\vec{x},a,s) \qquad (4)$$

(if F were true, and made false by doing action a, then condition N_F must have been true)

Beliefs in a changing world – actions and planning

Explanation closure axioms

Assume that formulas 1) and 2) characterize all the conditions under which an action a changes the value of the fluent F (that is a completeness assumption – allows us to conclude that actions that are not mentioned explicitly in effect axioms leave the fluent invariant).

We can express these assumptions by using the explanation closure axioms:

$$\neg F(\vec{x},s) \wedge F(\vec{x},do(a,s))) \supset \Pi_F(\vec{x},a,s) \qquad (3)$$

(if F were false, and made true by doing action a, then condition Π_F must have been true)

$$F(\vec{x},s) \wedge \neg F(\vec{x},do(a,s))) \supset N_F(\vec{x},a,s) \qquad (4)$$

(if F were true, and made false by doing action a, then condition N_F must have been true)

These explanation closure axioms can be rewritten as frame axioms:

$$\neg F(\vec{x},s) \wedge \neg \Pi_F(\vec{x},a,s) \supset \neg F(\vec{x},do(a,s)))$$

$$F(\vec{x},s) \wedge \neg N_F(\vec{x},a,s) \supset F(\vec{x},do(a,s)))$$

F remains false after doing *a* when Π_F is false.

F remains true after doing *a* when N_F is false.

Beliefs in a changing world – actions and planning

Successor state axioms

We assume that the KB entails the following:

-integrity of the effect axioms for every fluent F:

$$\neg\exists\vec{x},a,s.\Pi_F(\vec{x},a,s) \wedge N_F(\vec{x},a,s)$$

-unique names for actions:

$$A(\vec{x})=A(\vec{y}) \supset (x_1=y_1) \wedge\ldots \wedge(x_n=y_n),\ \ \vec{x}=(x_1,\ldots,x_n) \text{ and } \vec{y}=(y_1,\ldots,y_n)$$

$A(\vec{x})\neq B(\vec{y})$, where A and B are distinct action names.

Prop. Under these two assumptions, it can be shown that for any fluent F,

$$KB \models \text{axioms 1), 2), 3) and 4)}$$

is logically equivalent to

$$KB \models F(\vec{x},(do(a,s)) \equiv \Pi_F(\vec{x},a,s) \vee (F(\vec{x},s) \wedge \neg N_F(\vec{x},a,s))$$

successor state axiom for the fluent F – it completely characterizes the value of fluent F in the successor state resulting from performing action *a* in situation s

Beliefs in a changing world – actions and planning

For example, for the fluent Broken we have:

$$\begin{aligned}\text{Broken}(x,do(a,s)) \equiv\ & \exists r\{a=drop(r,x) \wedge \text{Fragile}(x)\} \vee \\ & \exists b\{a=explode(b) \wedge \text{NextTo}(b,x,s)\} \vee \\ & (\text{Broken}(x,s) \wedge \forall r\{a \neq repair(r,x)\}).\end{aligned}$$

$$\left[\begin{array}{l}\Pi_F(\vec{x},a,s) \text{ is } \exists r\{a=drop(r,x) \wedge \text{Fragile}(x)\} \vee \exists b\{a=explode(b) \wedge \text{NextTo}(b,x,s)\} \\ N_F(\vec{x},a,s) \text{ is } \exists r\{a=repair(r,x)\}\end{array}\right.$$

The KB contains an axiom that entails all the necessary effect and frame axioms for the fluent in question.

Beliefs in a changing world – actions and planning

For example, for the fluent Broken we have:

$$\begin{aligned} \mathrm{Broken}(x,do(a,s)) \equiv\ & \exists r\{a=drop(r,x) \wedge \mathrm{Fragile}(x)\} \vee \\ & \exists b\{a=explode(b) \wedge \mathrm{NextTo}(b,x,s)\} \vee \\ & (\mathrm{Broken}(x,s) \wedge \forall r\{a \neq repair(r,x)\}). \end{aligned}$$

$\Pi_F(\vec{x},a,s)$ is $\exists r\{a=drop(r,x) \wedge \mathrm{Fragile}(x)\} \vee \exists b\{a=explode(b) \wedge \mathrm{NextTo}(b,x,s)\}$

$N_F(\vec{x},a,s)$ is $\exists r\{a=repair(r,x)\}$

The KB contains an axiom that entails all the necessary effect and frame axioms for the fluent in question.

Obs. The solution depends on being able to put effect axioms in the normal form.

For actions with nondeterministic effects this is not possible.

For example:

$$\mathrm{Heads}(do(flipcoin,s)) \vee \mathrm{Tails}(do(flipcoin,s))$$

cannot be put into the normal form.

In general, we need to assume that every action *a* is deterministic.

Beliefs in a changing world – actions and planning

Complex actions

They are actions that have other actions as components:

- -conditional: if it rains I take the bus; otherwise I walk
- -iteration: while there are blocks on the table, pick one
- -nondeterministic choice: pick a red block from the table

For a complex action A, we use a formula called Do(A,s,s′) – when started in situation s, action A can terminate legally (i.e. through a sequence of actions) in situation s′.

Obs. Because of the nondeterministic nature, there may be more than one s′.

Beliefs in a changing world – actions and planning

The formula Do is defined recursively on the structure of the complex action as follows:

1. For any primitive action A, we have:

   $$Do(A,s,s') \stackrel{\text{def}}{=} Poss(A,s) \wedge (s'=do(A,s))$$

2. [A;B] sequential composition of complex actions A and B

   $$Do([A;B],s,s') \stackrel{\text{def}}{=} \exists s''.Do(A,s,s'') \wedge Do(B,s'',s')$$

3. [if Φ then A else B] – conditional

   $$Do([\text{if } \Phi \text{ then } A \text{ else } B],s,s') \stackrel{\text{def}}{=}$$
   $$[\Phi(s) \wedge Do(A,s,s')] \vee [\neg\Phi(s) \wedge Do(B,s,s')]$$

4. [Φ?] test action

   $$Do([\Phi?]\ s,s') \stackrel{\text{def}}{=} \Phi(s) \wedge s'=s$$

5. [A|B] – nondeterministic branch to A or B

   $$Do([A|B],s,s') \stackrel{\text{def}}{=} Do(A,s,s') \vee Do(B,s,s')$$

Beliefs in a changing world – actions and planning

6. [πx.A] – nondeterministic choice of a value for variable x

$$Do([\pi x.A],s,s') \stackrel{\text{def}}{=} \exists x.Do(A,s,s')$$

7. [while Φ do A] iteration

second-order quantification

$$Do([\text{while } \Phi \text{ do } A],s,s') \stackrel{\text{def}}{=} \forall P\{* \supset P(s,s')\},$$

where * is the conjunction of

interpreted as "smallest" relation P satisfying certain conditions

$\forall s_1.\ \neg\Phi(s_1) \supset P(s_1, s_1),$

$\forall s_1,s_2,s_3.\ \Phi(s_1) \wedge Do(A,s_1,s_2) \wedge P(s_2,s_3) \supset P(s_1, s_3)$

Beliefs in a changing world – actions and planning

6. [πx.A] – nondeterministic choice of a value for variable x

$$Do([\pi x.A],s,s') \stackrel{\text{def}}{=} \exists x.Do(A,s,s')$$

7. [while Φ do A] iteration

second-order quantification

$$Do([\text{while } \Phi \text{ do } A],s,s') \stackrel{\text{def}}{=} \forall P\{* \supset P(s,s')\},$$

where * is the conjunction of

interpreted as "smallest" relation

$\forall s_1.\ \neg\Phi(s_1) \supset P(s_1, s_1),$

P satisfying certain conditions

$\forall s_1,s_2,s_3.\ \Phi(s_1) \wedge Do(A,s_1,s_2) \wedge P(s_2,s_3) \supset P(s_1, s_3)$

<u>Obs</u>. Complex actions can be fully specified in the language of the situation calculus. The execution of a (GOLOG) program A consists of:

- finding a sequence of primitive actions $\vec{a}=<a_1,\ldots,a_n>$ such that

$$KB \models Do(A, S_0, do(\vec{a}, S_0))$$

- passing the sequence of actions $\vec{a}$ to a "robot" for execution in the world.

Beliefs in a changing world – actions and planning

Planning

It is the fundamental reasoning problem that requires to figure out what to do in order to make some arbitrary condition (or goal) true.

The sequence of actions that makes the goal true is called a plan.

Situation calculus is a natural candidate to formulate and support planning:

Given a formula, Goal(s), with a single free variable s, find a sequence of actions $\vec{a}$=<$a_1,\ldots,a_n$> such that

$$KB \models Goal(do(\vec{a},S_0)) \wedge Legal((do(\vec{a},S_0))$$

where $do(\vec{a},S_0)$ represents $do(a_n,do(a_{n-1},\ldots,do(a_1,S_0)\ldots))$

and $Legal((do(\vec{a},S_0))$ represents

$Poss(a_2,do(a_1,S_0)) \wedge Poss(a_3,do(<a_1,a_2>,S_0))\ldots \wedge Poss(a_n,do(<a_1,\ldots,a_{n-1}>,S_0))$.

We can use the Resolution with answer extraction for the query

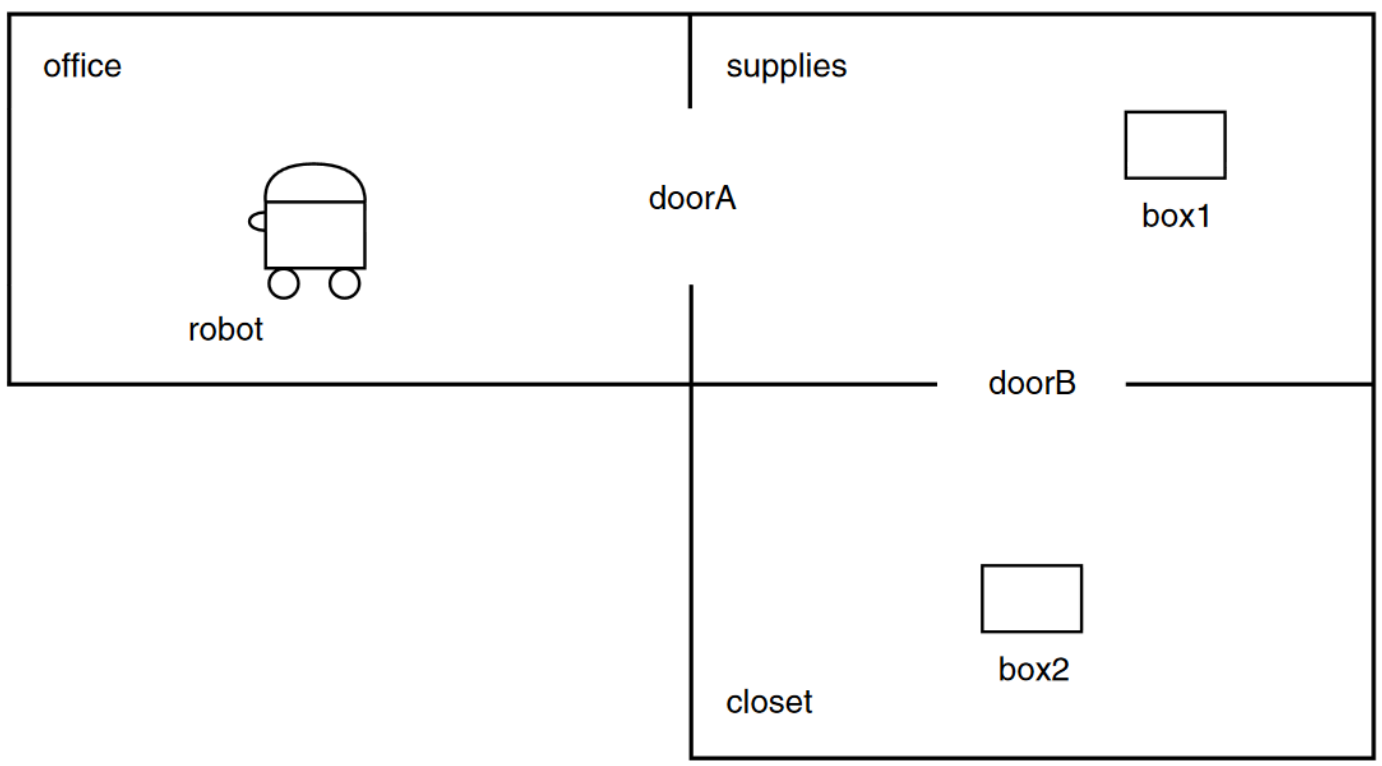
$$KB \models \exists s.Goal(s) \wedge Legal(s)$$

If the extracted answer is of the form $do(\vec{a},S_0)$, then $\vec{a}$ is a correct plan.

Beliefs in a changing world – actions and planning

Example – a simple robot world

Task: the robot has to get some box into the office



Beliefs in a changing world – actions and planning

The precondition axioms:

$Poss(goThru(d,r_1,r_2),s) \equiv Connected(d,r_1,r_2) \wedge InRoom(robot,r_1,s)$

$Poss(pushThru(x,d,r_1,r_2),s) \equiv Connected(d,r_1,r_2) \wedge InRoom(robot,r_1,s) \wedge InRoom(x,r_1,s)$

The successor state axiom:

$InRoom(x,r,do(a,s)) \equiv \Pi(x,a,r) \vee (InRoom(x,r,s) \wedge \neg\exists r'.(r{\neq}r') \wedge \Pi(x,a,r'))$,

where $\Pi(x,a,r)$ is

$x{=}robot \wedge \exists d\exists r_1.a{=}goThru(d,r_1,r) \vee$

$x{=}robot \wedge \exists d\exists r_1\exists y.a{=}pushThru(y,d,r_1,r) \vee$

$\exists d\exists r_1.a{=}pushThru(x,d,r_1,r)$

Facts about the initial situation:

$Connected(doorA,office,supplies)$

$Connected(doorB,supplies,closet)$

$InRoom(robot,office,S_0)$

$InRoom(box1,supplies,S_0)$

$InRoom(box2,closet,S_0)$

$Box(box1)$

$Box(box2)$

Beliefs in a changing world – actions and planning

The goal:

$\text{Goal}(s) \equiv \exists x.\ \text{Box}(x) \wedge \text{InRoom}(x,\text{office},s)$

First, the KB must be converted to CNF.

The formula to be proven, that is

$$\exists s_1 \exists x.\ \text{Box}(x) \wedge \text{InRoom}(x,\text{office}, s_1) \wedge \text{Legal}(s_1)$$

is negated.

Now the Resolution (including an answer predicate) is used to get the plan

do(pushThru(box1,doorA,supplies,office),

do(goThru(doorA,office,supplies),S_0))