Planning: situation calculus

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Representation of actions, situations, events

Propositional and first order logic are monotonic
• Once something is true it cannot become false

But, the world is dynamic:
• What is true now may not be true tomorrow
• Changes in the world may be triggered by our activities

Problems:
• How to represent the change in the FOL?
• How to represent actions we can use to change the world?
Planning

**Planning problem:**
- find a sequence of actions that achieves some goal
- an instance of a search problem
- the state description is typically *very complex and relies on a logic-based representation*

Methods for modeling and solving planning problems:
- State space search
- Situation calculus based on FOL
- STRIPS – state-space search algorithm based on restricted FOL
- Partial-order planning algorithms

Situation calculus

Provides a framework for representing change, actions and for reasoning about them

- **Situation calculus**
  - based on the *first-order logic*,
  - a *situation variable* models possible states of the world
  - properties and relations depend on different world states (situations)
  - *action objects* model activities
- **Inference:**
  - inference methods developed for FOL to do the reasoning
Situation calculus

Logic for reasoning about changes in the state of the world

The world dynamics is described by:
• Sequences of situations of the current state
• Changes from one situation to another are caused by actions

The situation calculus allows us to:
• Describe the initial state and the goal state
• Build the KB that describes the effect of actions (operators)
• Prove that the KB and the initial state can lead to the goal state
  – extracts a plan (sequence of actions) as side-effect of the proof

Situation calculus

The language is based on the First-order logic plus:
• Special variables: $s,a$ – objects of type situation and action
• Action functions: return actions (action objects).
  – E.g. $Move(A, \text{TABLE, B})$ represents a move action
  – $Move(x,y,z)$ represents an action schema
• Special function symbols of type situation
  – $s_0$ – initial situation
  – $DO(a,s)$ – represents the situation that is obtained after performing action $a$ in situation $s$
• Situation-dependent predicates, functions
  (also called fluents)
  – Relation: $On(x,y,s)$ – object $x$ is on object $y$ in situation $s$;
  – Function: $Above(x,s)$ – object that is above $x$ in situation $s$. 
Situation calculus. Blocks world example.

Initial state

- On(A, Table, s_0)
- On(B, Table, s_0)
- On(C, Table, s_0)
- Clear(A, s_0)
- Clear(B, s_0)
- Clear(C, s_0)
- Clear(Table, s_0)

Goal

Find a state (situation) s, such that

- On(A, B, s)
- On(B, C, s)
- On(C, Table, s)

Knowledge base: Axioms

Knowledge base is needed to support the reasoning:
- Must represent changes in the world due to actions.

Two types of axioms:
- **Effect axioms**
  - changes in situations that result from actions
- **Frame axioms**
  - things preserved from the previous situation

Example:
- blocks world with On, Clear predicates
- Move actions
Blocks world example. Effect axioms.

Effect axioms:
• represent the changes after the action is executed

Moving x from y to z. \( MOVE(x, y, z) \)

Effect of move changes on On relations
\[
On(x, y, s) \land Clear(x, s) \land Clear(z, s) \rightarrow On(x, z, DO(MOVE(x, y, z), s))
\]
\[
On(x, y, s) \land Clear(x, s) \land Clear(z, s) \rightarrow \neg On(x, y, DO(MOVE(x, y, z), s))
\]

Effect of move changes on Clear relations
\[
On(x, y, s) \land Clear(x, s) \land Clear(z, s) \rightarrow Clear(y, DO(MOVE(x, y, z), s))
\]
\[
On(x, y, s) \land Clear(x, s) \land Clear(z, s) \land (z \neq Table) \rightarrow \neg Clear(z, DO(MOVE(x, y, z), s))
\]

Blocks world example. Frame axioms.

• Frame axioms.
  – Represent relations/properties that remain unchanged by the executed action
  – Explicitly move the relations to the next situation (after the action)

On relations:
\[
On(u, v, s) \land (u \neq x) \land (v \neq y) \rightarrow On(u, v, DO(MOVE(x, y, z), s))
\]

Clear relations:
\[
Clear(u, s) \land (u \neq z) \rightarrow Clear(u, DO(MOVE(x, y, z), s))
\]
Planning in situation calculus

Planning problem:
• find a sequence of actions that lead to the goal

Planning in situation calculus is converted to the theorem proving problem

Goal state:
\( \exists s \text{ On}(A, B, s) \land \text{On}(B, C, s) \land \text{On}(C, \text{Table}, s) \)

• Possible inference approaches:
  – Inference rule approach
  – Conversion to SAT
• Plan (solution) is a byproduct of theorem proving
• Example: blocks world

Planning in the blocks world.

Initial state \((s_0)\)

\[
\begin{align*}
\text{On}(A, \text{Table}, s_0) & \quad \text{Clear}(A, s_0) & \quad \text{Clear}(\text{Table}, s_0) \\
\text{On}(B, \text{Table}, s_0) & \quad \text{Clear}(B, s_0) \\
\text{On}(C, \text{Table}, s_0) & \quad \text{Clear}(C, s_0)
\end{align*}
\]

Action: \( \text{MOVE}(B, \text{Table}, C) \)

\( s_1 = \text{DO(MOVE}(B, \text{Table}, C), s_0) \)

\[
\begin{align*}
\text{On}(A, \text{Table}, s_1) & \quad \text{Clear}(A, s_1) & \quad \text{Clear}(\text{Table}, s_1) \\
\text{On}(B, C, s_1) & \quad \text{Clear}(B, s_1) \\
\neg\text{On}(B, \text{Table}, s_1) & \quad \text{Clear}(C, s_1) \\
\text{On}(C, \text{Table}, s_1) & \quad \neg\text{Clear}(C, s_1)
\end{align*}
\]
Planning in the blocks world.

Initial state (s0)  s1  s2

\[
\begin{align*}
\text{Initial state (s0)}: & \quad \text{A} \quad \text{B} \quad \text{C} \\
\text{s1:} & \quad \text{A} \quad \text{B} \quad \text{C} \\
\text{s2:} & \quad \text{A} \quad \text{B} \quad \text{C}
\end{align*}
\]

\[s_1 = \text{DO(MOVE(B, Table, C), s_0)}\]
\[\text{On}(A, \text{Table}, s_1) \quad \text{Clear}(A, s_1) \quad \text{Clear}(\text{Table}, s_1)\]
\[\text{On}(B, C, s_1) \quad \text{Clear}(B, s_1) \quad \text{Clear}(\text{Table}, s_1)\]
\[\text{On}(C, \text{Table}, s_1) \quad \neg\text{Clear}(C, s_1)\]

\[\text{Action: MOVE(A, Table, B)}\]
\[s_2 = \text{DO(MOVE(A, Table, B), s_1)}\]
\[= \text{DO(MOVE(A, Table, B), DO(MOVE(B, Table, C), s_0))}\]
\[\text{On}(A, B, s_2) \quad \neg\text{On}(A, \text{Table}, s_2) \quad \neg\text{Clear}(B, s_2)\]
\[\text{On}(B, C, s_2) \quad \neg\text{On}(B, \text{Table}, s_2) \quad \neg\text{Clear}(C, s_2)\]
\[\text{On}(C, \text{Table}, s_2) \quad \text{Clear}(A, s_2) \quad \text{Clear}(\text{Table}, s_2)\]

M. Hauskrecht
Planning in the blocks world.

Initial state (s0)  s1  s2

\( s_1 = DO(MOVE(B, Table, C), s_0) \)
\( On(A, Table, s_1) \)
\( On(B, C, s_1) \)
\( \neg On(B, Table, s_1) \)
\( On(C, Table, s_1) \)

\( s_2 = DO(MOVE(A, Table, B), DO(MOVE(B, Table, C), s_0)) \)
\( On(A, B, s_2) \)
\( On(B, C, s_2) \)
\( On(C, Table, s_2) \)

\( Clear(A, s_1) \)
\( Clear(B, s_1) \)
\( \neg Clear(C, s_1) \)

\( Clear(A, s_2) \)
\( Clear(B, s_2) \)
\( \neg Clear(C, s_2) \)

\( Clear(Table, s_1) \)

DO functions capture the plan

Situation calculus: problems

Frame problem refers to:
• The need to represent a large number of frame axioms

Solution: combine positive and negative effects in one rule

\[ On(u, v, DO(MOVE(x, y, z), s)) \iff (\neg ((u = x) \land (v = y)) \land On(u, v, s)) \lor \]
\[ \lor (((u = x) \land (v = z)) \land On(x, y, s) \land Clear(x, s) \land Clear(z, s)) \]

Inferential frame problem:
  – We still need to derive properties that remain unchanged

Other problems:
• Qualification problem – enumeration of all possibilities under which an action holds
• Ramification problem – enumeration of all inferences that follow from some facts