CS 4649/7649 RIP
Robot Intelligence: Planning

Classical Planning

Sungmoon Joo

School of Interactive Computing
College of Computing
Georgia Institute of Technology

Course Info.

• Course Website: joosm.github.io/RIP2014
• Course Wiki: Teaming & Collaboration. Under construction
• Email me(sungmoon.joo@cc.gatech.edu) to introduce yourself
• Issues with registration?
What is Planning?

Planning: “devising a plan of action to achieve one’s goal” (Russel & Norvig)

Given:
- States
- Actions
- Initial State and Goal State
- Constraints

Task 1: Find a sequence of actions that take you from Init. to Goal
Task 2: Find actions that take you from any state to Goal
Task 3: Decide the best action to take now in order to improve your odds of reaching Goal
Task 4: Find a continuous path (in state space) that takes you from Init. to Goal

DARPA’s Assessment

...
State of the Art

First Representation: Predicate Logic

Unless, some how, we can describe the world, we cannot devise a plan.

Statement $\rightarrow$ Predicate

(Unary) Predicate: $P(x)$
Introduce a (functional) symbol($P$) for the predicate, and put the subject($x$) as an argument to the functional symbol.

N-ary predicate is defined similarly

Examples:
- $x$ is happy $\rightarrow$ Happy($x$): Unary predicate
- The suitcase contains a bomb $\rightarrow$ Contains(Suitcase, Bomb): Binary predicate
- $x$ is less than $y$ $\rightarrow$ LessThan($x,y$): Binary predicate
- $P(x_1,x_2,\ldots,x_n)$: n-ary predicate

*Proposition can be considered as a 0-nary predicate.
*Predicate Logic ~ First Order Logic(FOL)
First Representation: Predicate Logic

Objects (Constants): $a, 123, house, mike, robot, suitcase$

Variables: $x, y, z...$

Relations (Predicates): $LessThan, Contains, Parent, Happy$

Connectives: $\neg, \lor, \land, \Rightarrow$

Any expression is either true or false:

$LessThan(1, 2)$
$\neg Contains(Suitcase, Bomb)$

Truth Table

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$\neg A$</th>
<th>$A \land B$</th>
<th>$A \Rightarrow B$</th>
<th>$\neg A \lor B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Situation Calculus*

To represent and reason about dynamical worlds
To represent ‘change’, ‘state (implicitly time)’ is introduced.

- **Fluents** = Aspects of the world that change
  
  \[\text{Contains}(\text{Suitcase, Laptop, S0})\]
  
  \[\text{Working}(\text{Robot, S7})\]

- **Actions** are reified functions of constants. (They can be treated as constants themselves)
  
  \[\text{Put}(\text{Laptop, Suitcase})\]
  
  \[\text{Open}(\text{Car})\]
  
  \[\text{Lock}(\text{Car})\]

- **The do function:**
  
  \[do(\alpha, \sigma_0) \rightarrow \sigma_1\]
  
  \[\alpha = \text{action}\]
  
  \[\sigma = \text{state}\]

*Modern version is different from the original for clarity. Calculus = study about ‘change’.

---

**Situation Calculus**

- Effect axioms describe how a world changes by an action

- **Effect Axioms (Positive + Negative):** Specify the outcome of an action

  \[\text{In}(\text{Robot, Hall, s}) \land \text{Open}(\text{rm, s}) \Rightarrow \text{In}(\text{Robot, rm, do(enter(rm), s)})\]
  
  \[\text{In}(\text{Robot, Hall, s}) \land \text{Open}(\text{rm, s}) = \neg\text{In}(\text{Robot, Hall, do(enter(rm), s)})\]

\[
S1 = do(\text{enter(Class), S0})
\]

\[
\text{In}(\text{Robot, Class, S1}) \quad \neg\text{In}(\text{Robot, Hall, S1})
\]

\[
\text{In}(\text{Robot, Hall, S0})
\]

\[
\text{Open}(\text{Class, S0})
\]

\[
\text{Open}(\text{Office, S0})
\]

\[
S2 = do(\text{enter(Office), S0})
\]

\[
\text{In}(\text{Robot, Office, S2}) \quad \neg\text{In}(\text{Robot, Hall, S2})
\]

---
Motivational Detour: Implicit State Space

How many distinct states can we describe with \( n \) fluents?

\[
\begin{align*}
\text{In(Robot, Class, S1)} \\
\neg\text{In(Robot, Hall, S1)} \\
\text{Open(Door, Class, S1)} \\
\neg\text{In(Briefcase, Laptop, S1)}
\end{align*}
\]

S1

Implicit State Space Representation

How many distinct states can we describe with \( n \) fluents?

- One fluent: 2 states
- Two fluents: \( 2 \times 2 = 4 \) states
- \( n \) fluents = \( 2^n \) states

<table>
<thead>
<tr>
<th>Open(Door, s)</th>
<th>Hungry(Robot, s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Number of states is exponential in number of fluents.

Recall that fluents are themselves propositions over constants!
In(Robot, rm, s) = Educated(Students, rm, do(teach(rm), s))

\[ S1 = do(enter(Class), S0) \]
\[ S0 \]
\[ In(Robot, Class, S1) \]
\[ \neg In(Robot, Hall, S1) \]

\[ S1 \]
\[ S2 = do(enter(Office), S0) \]
\[ S0 \]
\[ In(Robot, Hall, S0) \]
\[ Open(Class, S0) \]
\[ Open(Office, S0) \]

\[ S2 \]
\[ S3 = do(enter(Office), S0) \]
\[ S0 \]
\[ In(Robot, Hall, S0) \]
\[ Open(Class, S0) \]
\[ Open(Office, S0) \]

\[ S1 \]
\[ S3 = do(teach, Class, do(enter, Class, S0)) \]
\[ Educated(Students, S3) \]

Warning: notations are abused...
Situation Calculus: Problems?

- Effect Axioms (Positive + Negative)
  \[ \text{In(Robot, Hall, s)} \land \text{Open(rm, s)} \Rightarrow \text{In(Robot, rm, do(enter(rm), s))} \]
  \[ \text{In(Robot, Hall, s)} \land \text{Open(rm, s)} \Rightarrow \neg\text{In(Robot, Hall, do(enter(rm), s))} \]

\[ S1 = \text{do(enter(Class), S0)} \]
\[ S0 \rightarrow S1 \]
\[ \neg\text{Open(Office, S0)} \]
\[ S2 = \text{do(enter, Office, S0)} \]

Warning: notations are abused...

Situation Calculus: Frame Problem*

- Effect Axioms (Positive + Negative)
  \[ \text{In(Robot, Hall, s)} \land \text{Open(rm, s)} \Rightarrow \text{In(Robot, rm, do(enter(rm), s))} \]
  \[ \text{In(Robot, Hall, s)} \land \text{Open(rm, s)} \Rightarrow \neg\text{In(Robot, Hall, do(enter(rm), s))} \]

\[ S1 = \text{do(enter, Class, S0)} \]
\[ S0 \rightarrow S1 \]
\[ \neg\text{Open(Office, S0)} \]
\[ \text{Color(Sky, Blue)} \]
\[ S2 = \text{do(enter, Office, S0)} \]

\[ \neg\text{In(Robot, Hall, S1)} \]

*First recognized by McCarthy & Hayes (1969)

Specifying only which conditions are changed by the actions do not allow, in logic, to conclude that all other conditions are not changed.
**Frame Axioms**

Effect Axioms (Positive + Negative)

\[ \text{In(Robot, Hall, s)} \land \text{Open(rm, s)} = \text{In(Robot, rm, do(enter(rm), s))} \]
\[ \text{In(Robot, Hall, s)} \land \text{Open(rm, s)} = \neg \text{In(Robot, Hall, do(enter(rm), s))} \]

For each unchanged fluent we add:

\[ \text{Open(Office, s)} = \text{Open(Office, do(enter(rm), s))} \]
\[ \neg \text{Open(Office, s)} = \neg \text{Open(Office, do(enter(rm), s))} \]
\[ \text{Color(Sky, Blue, s)} = \text{Color(Sky, Blue, do(enter(rm), s))} \]
\[ \neg \text{Color(Sky, Blue, s)} = \neg \text{Color(Sky, Blue, do(enter(rm), s))} \]

How many in total? (for \( n \) distinct fluents and \( m \) distinct actions)

\[ 2nm \]

(Not exponential – but often not practical)

Explicitly specify that all conditions not affected by actions are not changed while executing that action.

---

**Frame Problem**

- (Representational) Frame Problem
  
  Significant because the real world has very many fluents
  
  Size of axioms: \( O(mn) \)

- Inferential Frame Problem
  
  Problem of projecting forward the results of a \( t \) step of actions in time \( O(nt) \)

- Origin of the Name: “Frame Problem”
  
  (i) Frame of Reference in Physics, which is assumed stationary wrt which motion is measured.
  
  (ii) Frames of a movie, in which normally most of the background stays constant while changes occur in the foreground
Ramification Problem

What are the ramifications of an action?
- The robot entered the room.
- Its sensors are now in the room.
- The object it was carrying is now in the room.
- The robot is visible to you.
  ➔ the indirect consequences of an action
Do we really want to say all that in an effect axiom?

How does this relate to the Frame Problem?
  ➔ Ramification Problem = The frame problem in the context of actions with indirect effects

A solution: STRIPS

Shakey @ Stanford

https://www.youtube.com/watch?v=qXdn6ynwpil