Introduction to Cognitive Robotics

- Situation Calculus
  1. Fluents
  2. Effect Axioms
  3. Frame Problem
  4. Successor State Axioms
  5. Database Example

---

Review

We want to a uniform theoretical and implementation framework that integrates the reasoning, perception, and action of a robot.

We Adopt the K.R. Hypothesis – using 1st order logic as our representation language.

a. Sentences in 1st order logic will represent the knowledge of the agent about the world, actions, and effects.

b. These sentences play a causal role in engendering the behavior of the agent.
   This is not only explicit knowledge, but also implicit knowledge.

\[ \{ a | KB \models a \} \]

Problem: In our discussion last time of 1st order logic, can only represent knowledge at one point in time. How do we talk about change within logic?

---

Action Theories: Motivation

To perform non-trivial reasoning (like planning) an intelligent agent situated in a changing world needs:

- The knowledge of causal laws describing the actions or events that change the world.
- The ability to observe and record occurrences of these actions/events.
- A model of the state of the world.

Action theories are formalisms intended to represent and reason about actions and their effects.

---

The Situation Calculus (McCarthy)

- The world is conceived as being in some state \( s \); this state can change only in consequence of some agent performing an action.
- Often the constant \( S_0 \) is used to denote the initial situation.
- \( do(\alpha, s) \) denotes the successor state to \( s \) resulting from performing the action \( \alpha \).
- Actions may be parameterized:
  - \( put(x, y) \) might stand for the action of putting object \( x \) on object \( y \); 
  - \( do(put(block_1, block_2), s) \) denotes that state resulting from placing \( block_1 \) on \( block_2 \) when the world is in state \( s \).
Axiomatization of Initial Situation

\neg \text{Holding}(\text{robot}, \text{obj}_1, S_0)
\text{location}(\text{robot}, S_0) = 0
\text{location}(\text{obj}_1, S_0) = 1
\text{location}(\text{obj}_2, S_0) = 2
\text{color}(\text{obj}_1, \text{red}, S_0)

Can also say:
\text{Holding}(\text{robot}, \text{obj}_1, S_0) \lor
\text{Holding}(\text{robot}, \text{obj}_2, S_0)

The Situation Calculus (Continued)

- \text{Fluents} = \text{those relations whose truth values}
  \text{may vary from state to state.}
  - \text{Denoted by predicate symbols taking a}
    \text{state term as one of their arguments.}
    - In a world in which it is possible to paint
      \text{objects, we might have a fluent}
      \text{COLOR}(x, c, s), \text{meaning that the color of}
      \text{object } x \text{ is } c \text{ when the world is in state } s.
    - \text{Functional Fluents location}(x, s).

Preconditions of Actions

- \text{Actions have } \text{preconditions: necessary}
  \text{conditions which a world state must satisfy if}
  \text{the action can be performed in this state.}
  - If it is possible for a robot \( r \) to pick up an
    \text{object } x \text{ in the world state } s \text{ then the}
    \text{robot is not holding any object, it is next}
    \text{to } x, \text{ and } x \text{ is not heavy:}

    \[ \text{Poss} \left( \text{pickup}(r, x), s \right) \rightarrow \]
    \[ [\forall z] \neg \text{Holding}(r, z, s) \land \neg \text{Heavy}(x) \land \]
    \[ \text{Nexto}(r, x, s). \]

- Whenever it is possible for a robot to
  \text{repair an object, then the object must}
  \text{be broken, and there must be glue available:}

    \[ \text{Poss} \left( \text{repair}(r, x), s \right) \rightarrow \text{HasGlue}(r, s) \land \]
    \[ \neg \text{Broken}(x, s). \]

Effects of Actions

- \text{World dynamics are specified by } \text{effect axioms}
  \text{which specify the effect of a given action on}
  \text{the truth value of a given fluent.}
  - The effect on the fluent \text{Broken} of a
    \text{robot dropping an object:}

    \[ \text{Fragile}(x) \rightarrow \text{Broken}(x, \text{do(} \text{drop}(r, x), s)) \].

  - A robot repairing an object causes it not
    \text{to be broken:}

    \[ \neg \text{Broken}(x, \text{do(} \text{repair}(r, x), s)). \]
Nonmonotonic Reasoning

\[ \text{Bird(tweety)} \vdash \text{Flies(tweety)} \]

but

\[ \text{Bird(tweety)} \land \text{Penguin(tweety)} \vdash \neg \text{Flies(tweety)} \]

\[ \text{Bird}(x) \land \neg \text{Penguin}(x) \land \\
\neg \text{Ostrich}(x) \ldots \rightarrow \text{Flies}(x) \]

Qualification Problem (cont)

Here we will ignore the minor qualifications/exceptions.

• One approach: Assume that for each action \( A(\vec{x}) \), we have an axiom of the form

\[ \text{Poss}(A(\vec{x}), s) \leftrightarrow \Pi_A(\vec{x}, s), \]

where \( \Pi_A(\vec{x}, s) \) is a first order formula with free variables \( \vec{x}, s \) which does not mention \( \text{do} \). We shall call these action precondition axioms.

• Example:

\[ \text{Poss}(\text{Pickup}(r, x), s) \leftrightarrow \\
[ (\forall z) \neg \text{Holding}(r, z, s) ] \land \neg \text{Heavy}(x) \land \\
\text{Nextto}(r, x, s). \]
• Axioms other than effect axioms are required for formalizing dynamic worlds. These are called *frame axioms*, and they specify the action *invariants* of the domain, i.e., those fluents unaffected by the performance of an action.

- A positive frame axiom – dropping things does not affect an object’s color:
  \[ \text{COLOR}(y, c, s) \rightarrow \text{COLOR}(y, c, \text{do(DROP}(r, x), s)). \]

- A negative frame axiom – not breaking things:
  \[ \neg \text{BROKEN}(y, s) \wedge [y \neq x \vee \neg \text{FRAGILE}(y)] \rightarrow \neg \text{BROKEN}(y, \text{do(DROP}(r, x), s)). \]

---

**A Simple Solution to the FP (Reiter, Hass, Schubert)**

• **Example:** Suppose there are two positive effect axioms for the fluent *broken*:

\[ \text{Poss(DROP}(r, x), s) \wedge y = x \wedge \text{FRAGILE}(y) \rightarrow \text{BROKEN}(y, \text{do(DROP}(r, x), s)), \]

\[ \text{Poss(EXplode}(b), s) \wedge \text{Nexto}(b, y, s) \rightarrow \text{BROKEN}(y, \text{do(EXplode}(b), s)). \]

• And one negative effect axiom:

\[ \text{Poss(REPAIR}(r, x), s) \wedge y = x \rightarrow \neg \text{BROKEN}(y, \text{do(REPAIR}(r, x), s)). \]

• Now appeal to the following completeness assumption:

*The above axioms characterize all the conditions under which action a leads to y being broken.*
Successor State Axioms

- For the above example, the successor state axiom for Brocken is:

\[
[Broke(n(y, do(a, s))) \leftrightarrow \\
(\exists r)[a = \text{DROP}(r, y) \land \text{FRAGILE}(y)] \lor \\
(\exists b)[a = \text{EXPLODE}(b) \land \text{NEXTO}(b, y, s)] \lor \\
Broke(n(y, s) \land \neg(\exists r)a = \text{REPAIR}(r, y)]
\]

The Solution

- Reiter’s solution yields the following axioms:
  1. Successor state axioms: for each fluent \( R \),
     \[
     \text{Poss}(a, s) \rightarrow [R(do(a, s)) \leftrightarrow \\
     \gamma^+_R(a, s) \lor R(s) \land \neg \gamma^-_R(a, s)].
     \]
  2. For each action \( A \), a single action precondition axiom of the form:
     \[
     \text{Poss}(A(x), s) \leftrightarrow I_A(x, s).
     \]
  3. Unique names axioms for actions.

- Ignoring the unique names axioms (whose effects can be compiled), this axiomatization requires \( \mathcal{F} + \mathcal{A} \) axioms in total, compared with the \( 2 \times \mathcal{A} \times \mathcal{F} \) explicit frame axioms that would otherwise be required.

Planning

Find a sequence of actions that if performed in a world with an axiomatized initial situation, will lead to a situation in which some goal statement will be true.

\[
do(\text{PUTDOWN}, do(\text{MOVE}, do(\text{PICKUP}, S_0)))
\]

\[\text{[PICKUP, MOVE, PUTDOWN]}\]

\[
KB \models \exists sG(s) \\
KB \models G(s)\{s/do(\ldots)\}
\]

Can Use Answer Extraction and Resolution.

\[
KB \cup \{\neg G(s) \lor \text{ANS}(s)\}
\]

Shakey Project (SRI), Abandoned resolution theorem proving for planning and developed STRIPS planning.

Projection Problem

Does a particular sentence hold in the situation resulting from the execution of a particular sequence of actions?

One could use resolution, but note that the successor state axioms contain equivalences which when converted to clause form would create a very large space.

Can we avoid using a general purpose reasoning mechanism with the successor state axioms.
Regression

- Regression operators transform something of the form
  \[ G(\text{do}(A_1, \text{do}(\ldots, S_0) \ldots)) \]
  to
  \[ R(S_0) \]

- Theorem
  \[ \mathcal{F} \models G(\text{do}(A_1, \text{do}(\ldots, S_0) \ldots)) \]
  if
  \[ \mathcal{F} - \mathcal{F}_{s_0} \models R(S_0) \]

Education Database (Continued)

- Initial Database State
  These will be arbitrary first order sentences, the only restriction being that fluents mention only the initial state \(S_0\).

\[
\text{ENRLOLLED}(\text{SUE, C100, } S_0) \lor \text{ENRLOLLED}(\text{SUE, C200, } S_0) \\
(\exists c) \text{ENRLOLLED}(\text{BILL, c, } S_0), \\
(\forall p) \text{PRERQU}(p, \text{P300}) \leftrightarrow p = \text{P100} \lor p = \text{M100}, \\
(\forall p) \neg \text{PRERQU}(p, \text{C100}), \\
(\forall c) \text{ENRLOLLED}(\text{BILL, c, } S_0) \leftrightarrow c = \text{M100} \lor c = \text{C100} \lor c = \text{P200}, \\
\text{ENRLOLLED}(	ext{MARY, C100, } S_0), \\
\neg \text{ENRLOLLED}(	ext{JOHN, M200, } S_0), \ldots \\
\text{GRADE}(	ext{SUE, P300, } S_0), \\
\text{GRADE}(	ext{BILL, M200, 75, } S_0), \\
\text{GRADE}(	ext{BILL, M200, 70, } S_0), \ldots
\]

An Education Database

- Relations
  1. \text{ENRLOLLED}(st, course, s): st is enrolled in course when the database is in state s.
  2. \text{GRADE}(st, course, grade, s): The grade of st in course is grade when the database is in state s.
  3. \text{PRERQU}(pre, course): pre is a prerequisite course for course.

Example: Continued

- Database Transactions
  - Denote by function symbols.
  - Treat exactly like actions in situation calculus planning.

- For the example, there are three transactions:
  1. \text{REGISTER}(st, c),
  2. \text{CHANGE}(st, c, g),
  3. \text{DROP}(st, c).
Example: Continued

- Registration Prerequisites:

\[ \text{Poss}(\text{register}(st, c, s) \leftrightarrow (\forall p).\text{PREREQ}(p, c) \rightarrow (\exists g).\text{GRADE}(st, p, g, s) \land g \geq 50). \]

- It is possible to change a student’s grade iff he has a grade which is different than the new grade:

\[ \text{Poss}(\text{change}(st, c, g, s) \leftrightarrow (\exists g').\text{GRADE}(st, c, g', s) \land g' \neq g). \]

- A student may drop a course iff the student is currently enrolled in that course:

\[ \text{Poss}(\text{drop}(st, c, s) \leftrightarrow \text{ENROLLED}(st, c, s)). \]

- Update Specifications

\[ \text{[ENROLLED}(st, c, do(a, s)) \leftrightarrow a = \text{register}(st, c) \lor \text{ENROLLED}(st, c, s) \land a \neq \text{drop}(st, c)]. \]

---

Projection Problem

\[ KB \models \exists c \text{ ENROLLED} \text{(JOHN, c, do(MARY, C100), de(DROP(JOHN, C100), S0))).} \]

Is John enrolled in any courses after the sequence of actions

\[ \text{[DROP(JOHN, C100), REGISTER(MARY, C100)]} \]

drop has been executed?

References

