CSEN 901 - Introduction to Artificial Intelligence

Lecture 8 - Classical Planning

Dr. Nourhan Ehab

nourhan.ehab@guc.edu.eg

Recall: What We Did So Far

- We studied search agents and knowledge-based agents.

Recall: What We Did So Far

- We studied search agents and knowledge-based agents.
- Both need to represent an exponentially large search space.

Outline

1. Classical Planning
2. State Space Planning
3. Recap

Planning Agents

- Planning is the process of deliberation to find a sequence of actions from an initial state to a goal.

Planning Agents

- Planning is the process of deliberation to find a sequence of actions from an initial state to a goal.
- It is a combination of both search and knowledge-based agents.

Classical Representations of Planning

Representations

- States: a set of ground atoms (closed world assumption).
  - If p holds in s, then $p \in s$.
  - If p does not hold in s, then $p \notin s$.
- Operators: an operator o is $\langle name(o), preconds(o), effects(o) \rangle$.
- Actions: ground instances of operators.
- Goal: a set of literals.

Classical Representations of Planning

Representations

- States: a set of ground atoms (closed world assumption).
  - If p holds in s, then $p \in s$.
  - If p does not hold in s, then $p \notin s$.
- Operators: an operator o is $\langle name(o), preconds(o), effects(o) \rangle$.
- Actions: ground instances of operators.
- Goal: a set of literals.
- A planning problem is defined as $\langle O, s_0, g \rangle$ with:
  1. O is a set of operators.
  2. s_0 is the initial state.
  3. g is the goal.

Back to the Blocks World

GUC German University in Cairo

Example

Assume a planning problem (O, s_0, g) where O:

- PickUp(x)
  - **Preconditions:** $\texttt{Clear}(x), \texttt{OnTable}(x), \texttt{EH}$
  - **Effects:** $\neg\texttt{OnTable}(x), \neg\texttt{Clear}(x), \neg\texttt{EH}, \texttt{Holding}(x)$
- PutDown(x)
  - **Preconditions:** $\texttt{Holding}(x)$
  - **Effects:** $\texttt{OnTable}(x), \texttt{Clear}(x), \texttt{EH}, \neg\texttt{Holding}(x)$
- Stack(x, y)
  - **Preconditions:** $\texttt{Holding}(x), \texttt{Clear}(y)$
  - **Effects:** $\texttt{On}(x, y), \texttt{Clear}(x), \texttt{EH}, \neg\texttt{Clear}(y), \neg\texttt{Holding}(x)$
- Unstack(x, y)
  - **Preconditions:** $\texttt{On}(x, y), \texttt{Clear}(x), \texttt{EH}$
  - **Effects:** $\texttt{Holding}(x), \texttt{Clear}(y), \neg\texttt{On}(x, y), \neg\texttt{Clear}(x), \neg\texttt{EH}$

Back to Blocks World

GUC German University in Cairo

Example

- $s_0 = \{\texttt{On(A,B)},\texttt{On(B,C)},\texttt{OnTable(C)},\texttt{Clear(A)},\texttt{EH}\}$
- $g = \{\texttt{Clear(C)}, \neg\texttt{Holding(x)}\}$

Goal Satisfaction

Definition

A state s satisfies a goal g (denoted $s \models g$) if there is a substitution θ such that:

1. For every positive literal $p \in g$, $SUBST(\theta, p) \in s$.
2. For every negative literal $\neg n \in g$, $SUBST(\theta, n) \notin s$.

Example

Let $s = \{P(A, B), Q(B, C)\}$ and $g = \{P(x, B), \neg Q(x, C)\}$.

Goal Satisfaction

Definition

A state s satisfies a goal g (denoted $s \models g$) if there is a substitution θ such that:

1. For every positive literal $p \in g$, $SUBST(\theta, p) \in s$.
2. For every negative literal $\neg n \in g$, $SUBST(\theta, n) \notin s$.

Example

Let $s = \{P(A, B), Q(B, C)\}$ and $g = \{P(x, B), \neg Q(x, C)\}$.
$s \models g$ under $\theta = \{A/x\}$

Outline

1. Classical Planning
2. State Space Planning
3. Recap

Progression and Regression Planning

- Our approach will be reducing the planning problem to a search problem.

Progression and Regression Planning

- Our approach will be reducing the planning problem to a search problem.
- Two ways of doing this:
  1. Start from the initial state and apply all possible operators to compute next states till a goal is reached $\rightarrow$ Progression Planning.

Progression and Regression Planning

- Our approach will be reducing the planning problem to a search problem.
- Two ways of doing this:
  1. Start from the initial state and apply all possible operators to compute next states till a goal is reached → Progression Planning.
  2. Start from the goal and compute possible previous states till the initial state is reached → Regression Planning.

Progression Planning

	Progression
Initial State	s_0
Operators	applicable actions
Transition fn $\gamma(s,a)$	$s \cup effects^+(a) - effects^-(a)$
Goal test	$s \models g$

An action a and substitution θ are applicable in state s if:

1. every positive precondition p of a, $SUBST(\theta, p) \in s$.
2. every negative precondition $\neg n$ of a, $SUBST(\theta, n) \notin s$

Progression Planning Algorithm

function PROGRESSION-PLANNER(O, s_0, g) **returns** a plan
  $s \longleftarrow s_0$
  $\pi = ()$
  loop
    if s satisfies g, **then return** π
    $app = \{a | a$ is a ground instance of some $o \in O$ which is applicable to $s\}$
    if $app = \varnothing$ **then return** *failure*
    nondeterministically choose $a \in app$
    $s \longleftarrow \gamma(s, a)$
    $\pi \longleftarrow \pi \cdot a$

Regression Planning

	Regression
Initial State	g
Operators	relevant operators (has effects in s and does not negate anything in s)
γ^{-1}(s,a)	$s - effects(a) \cup preconds(a)$
Goal test	$s_0 \models s$

An action a and substitution θ are relevant for g if:

1. $SUBST(\theta, g) \cap effects(SUBST(\theta, a)) \neq \varnothing$;
2. $SUBST(\theta, g^+) \cap effects^-(SUBST(\theta, a)) = \varnothing$; and
3. $SUBST(\theta, g^-) \cap effects^+(SUBST(\theta, a)) = \varnothing$;

Regression Planning Algorithm

function REGRESSION-PLANNER(O, s_0, g) **returns** a plan
  $\pi = ()$
  loop
    if s_0 satisfies g with substitution σ, **then return** SUBST(σ, π)
    $rel = \{(o, \sigma) | (o, \sigma)$ is relevant for $g\}$
    if $rel = \varnothing$ **then return** *failure*
    nondeterministically choose $(o, \sigma) \in rel$
    $g \longleftarrow \gamma^{-1}(g, (o, \sigma))$
    $\pi \longleftarrow$ SUBST(σ, o) $\cdot$ SUBST(σ, π)

Back to Blocks World

GUC German University in Cairo

Example

Trace Progression and Regression planning on the previously defined planning problem.

Outline

Points to Take Home

1. Classical Planning Representations.
2. Progression Planning.
3. Regression Planning.
4. Reading Material:
   - R&N, Chapter 11, Section 11.1, 11.2.1, 11.2.2.

Next Lecture: Partial Order Planning!

Due Credits

The presented material is based on previous editions of the course at the GUC due to Prof. Haythem Ismail.