The task of coming up with a sequence of actions that will achieve a goal is called **planning**.

Simple approaches:
- Search-based
- Logic-based

**Representation** of states and actions become important issues.

---

**Example Domain: Wumpus World**

- Want to get to the gold and grab it.
- Want to avoid pits and the “wumpus”.
- Clues: breeze near pits and stench near the wumpus.
- Other sensors: wall (bump), gold (glitter), kill (scream)
- Actions: move, grab, or shoot.

**Performance measure**
- +1000: picking up gold
- -1000: fall in a pit, or get eaten by the wumpus
- -1: each action taken
- -10: each arrow used
Evolution of Knowledge in WW

- Move from [1,1] to [2,1].
- Based on the sensory data (breeze), we can mark [2,2] and [3,1] as potential pits, but not [1,1] since we came from there and we already know there's no pit there.

Inference in Wumpus World

- Knowledge Base: basic rules of the Wumpus World.
- Additional knowledge is added to the KB: facts you gather as you explore ([x,y] has stench, breeze, etc.)
- We can ask if a certain statement is a logical consequence of the KB: “There is a pit in [1,2]”

KB: basic rules, plus [1,1] and [2,1] explored.

- \( \alpha_1 \) = “There is no pit in [1,2]”
- \( \alpha_2 \) = “There is no pit in [2,2]”
- Only \( \alpha_1 \) follows from the KB

\[ \text{KB} \models \alpha_1 \iff \text{Model(KB)} \subseteq \text{Model(\alpha_1)}. \]
**Propositional-logic-based Agent**

- Query KB: Is there a Wumpus in [x,y]? Is there a pit in [x,y]?
- Add knowledge to KB (perceptual input): Breeze felt in [x,y], Stench detected in [x,y], etc.
- Decide which action to take (move where, etc.): Move to [x,y], grab gold, etc.

Note: here, there's only one goal, to grab the gold. Can we specify an arbitrary goal and derive a plan?

Problem: Propositions need to be explicit about location, e.g., \( \text{Breeze}_{x,y}, \text{Stench}_{x,y}, \neg \text{Wumpus}_{x,y} \).

**Situation Calculus**

Make propositional-logic-based planner scalable.

- Situations: logical terms indicating a state.
  - Example: In situation \( S_0 \) taking action \( a \) leads to situation \( S_1 \):
    \[
    S_1 = \text{Result}(a, S_0).
    \]

- Fluents: functions and predicates that vary from one situation to the next.
  - Example: \( \neg \text{Holding}(\text{Gold}_1, S_0), \text{Age}(\text{Wumpus}) \)

Other stuff: Atemporal/eternal predicates \( \text{Gold} (\text{Gold}_1) \), empty actions \( \text{Result}([], s) = s \), sequence of actions (\( seq \) followed by \( a \)) \( \text{Result}([a|seq], s) = \text{Result}(seq, \text{Result}(a, s)) \).

**Situation Calculus: Tasks**

- Projection:
  Deduce the outcome of a given sequence of actions

- Planning:
  Find a sequence of actions that achieves a desired effect.
  - Example: Wumpus world

  Initial: \( \text{At}(\text{Agent}, [1, 1], S_0) \land \text{At}(\text{G}_1, [1, 2], S_0), \ldots \)
  Goal: \( \exists seq \text{ At}(\text{G}_1, [1, 1], \text{Result}(seq, S_0)) \)

**Describing Actions in Situation Calculus**

Two axioms:

- Possibility axiom: when it is possible to execute an action
  \[
  \text{Preconditions} \rightarrow \text{Poss}(a, s)
  \]

- Effect axiom: What happens when a possible action is taken
  \[
  \text{Poss}(a, s) \rightarrow \text{Changes} \text{ that result}
  \]
Wumpus World: Axioms

• Possibility axioms: Move, grab, release

\[
\text{At} (\text{Agent}, x, s) \land \text{Adjacent} (x, y) \rightarrow \text{Poss} (\text{Go} (x, y), s)
\]

\[
\text{Gold} (g) \land \text{At} (\text{Agent}, x, s) \land \text{At} (g, x, s) \rightarrow \text{Poss} (\text{Grab} (g), s)
\]

\[
\text{Holding} (g, s) \rightarrow \text{Poss} (\text{Release} (g), s)
\]

• Effect axioms: Move, Grab, Release

\[
\text{Poss} (\text{Go} (x, y), s) \rightarrow \text{At} (\text{Agent}, y, \text{Result} (\text{Go} (x, y), s))
\]

\[
\text{Poss} (\text{Grab} (g), s) \rightarrow \text{Holding} (g, \text{Result} (\text{Grab} (g), s))
\]

\[
\text{Poss} (\text{Release} (g), s) \rightarrow \neg \text{Holding} (g, \text{Result} (\text{Release} (g), s))
\]

13

Frame Problem

• Representing all things that stay the same: Frame problem.

• In the previous slide, we cannot deduce if the following can be proven (\(G_1\) represents a particular lump of gold):

\[
\text{At} (G_1, [1, 1], \text{Result} (\text{Go} ([1, 1], [1, 2]), \text{Grab} (G_1), [1, 1], s_0))
\]

• It is because the effect axioms say only what should change, but not what does not change when actions are taken.

\[
\text{At}(G, s) \land (\neg \text{Agent} \land \neg \text{Holding}(G, s))
\]

\[
\rightarrow \text{At}(G, s) \land \text{Result} (\text{Grab} (G, [2, 1], s))
\]

\[
\rightarrow \text{Poss} (\text{Gold} (G), s)
\]

13

Two Frame Problems

• Representational frame problem: Explained in the previous slide

• Inferential frame problem: To project results of a t-step sequence of actions in time \(O(E_t)\) rather than \(O(F_t)\) or \(O(AE_t)\).

\(E_t\) is the number of actions, typically much less than \(F_t\), the number of fluent predicates.

14

Solving the Representational Frame Problem

Consider how each fluent predicate evolves over time:

Successor-state axioms: Action is possible

Fluent predicate is true in result state \(\leftrightarrow\) Action’s effect made it true

Successor state: Action left it alone

Example:

\[
\text{Poss}(a, s) \rightarrow (\text{At}(\text{Agent}, y, \text{Result}(a, s)) \leftrightarrow a = \text{Go}(c, y))
\]

\(\lor (\text{At}(\text{Agent}, y, s) \land a \neq \text{Go}(y, z))\)

It was true before and action left it alone.

Problem is that you need \(O(AF)\) such axioms for all (action, fluent) pair (\(A\): num of actions, \(F\): num of fluent predicates).

15

Remaining issues: implicit effect (moving while holding something moves that something as well) – ramification problem. Can solve by using a more general successor-state axiom.
Solving the Inferential Frame Problem

- Given a $t$-step plan $p (S_t = Result(p, S_0))$, decide which fluents are true in $S_t$.

- We need to consider each of the $F$ frame axiom of each time step $t$.

- Axioms have an average size of $AE/F$, we have an $O(AEt)$ inferential work. Most of the work is done copying unchanged fluents from time step to time step.

- Solutions: use fluent calculus rather than situation calculus, or make the process more efficient.

Other Formalisms

- Event calculus: Fluents hold at different time points, not situations. Reasoning is done over time.

- Other constructs: generalized events (spatiotemporal), process, intervals, etc.

- Formal theory of belief: propositional attitude, reification, etc.

Truth Maintenance Systems

New facts inferred from the KB can turn out to be incorrect.

- Let's say $P$ was derived in the KB and later it was found that $\neg P$.

- Adding $\neg P$ to the KB will invalidate the entire KB, so $P$ should be removed ($Retract(KB, P)$).

- Care needs to be taken since other facts in the KB may have been derived from $P$, etc.

- Truth maintenance systems are designed to handle these complications.
Planning Approaches

- State-space search: forward or backward.
- Heuristic search: subgoal independence assumption.
- Partial-order planning: utilize problem decomposition. Can place two actions into a plan without specifying the order. Several different total order plans can be constructed from partial order plans.