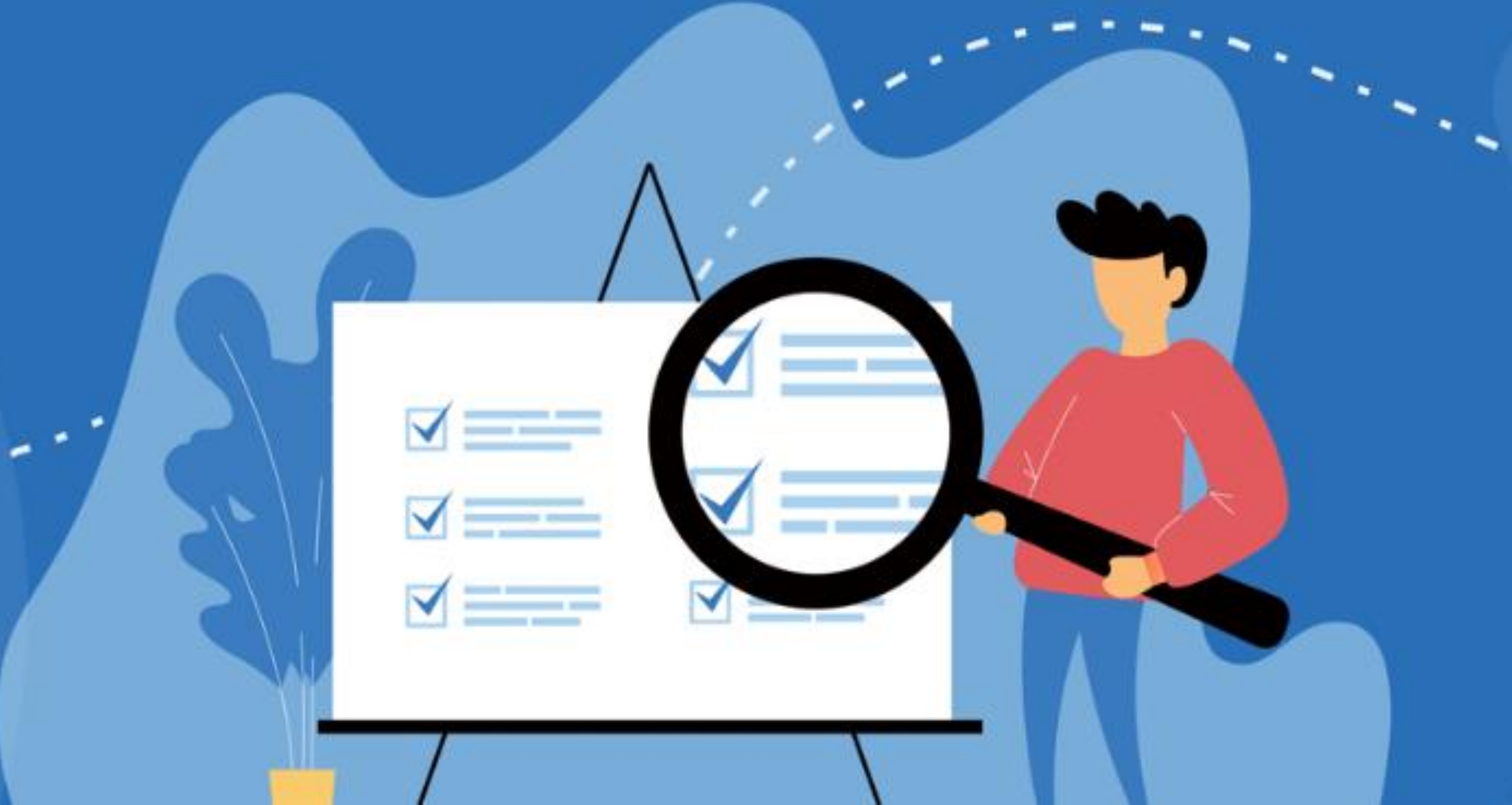
HOW TO PRESENT AN
ACTION PLAN

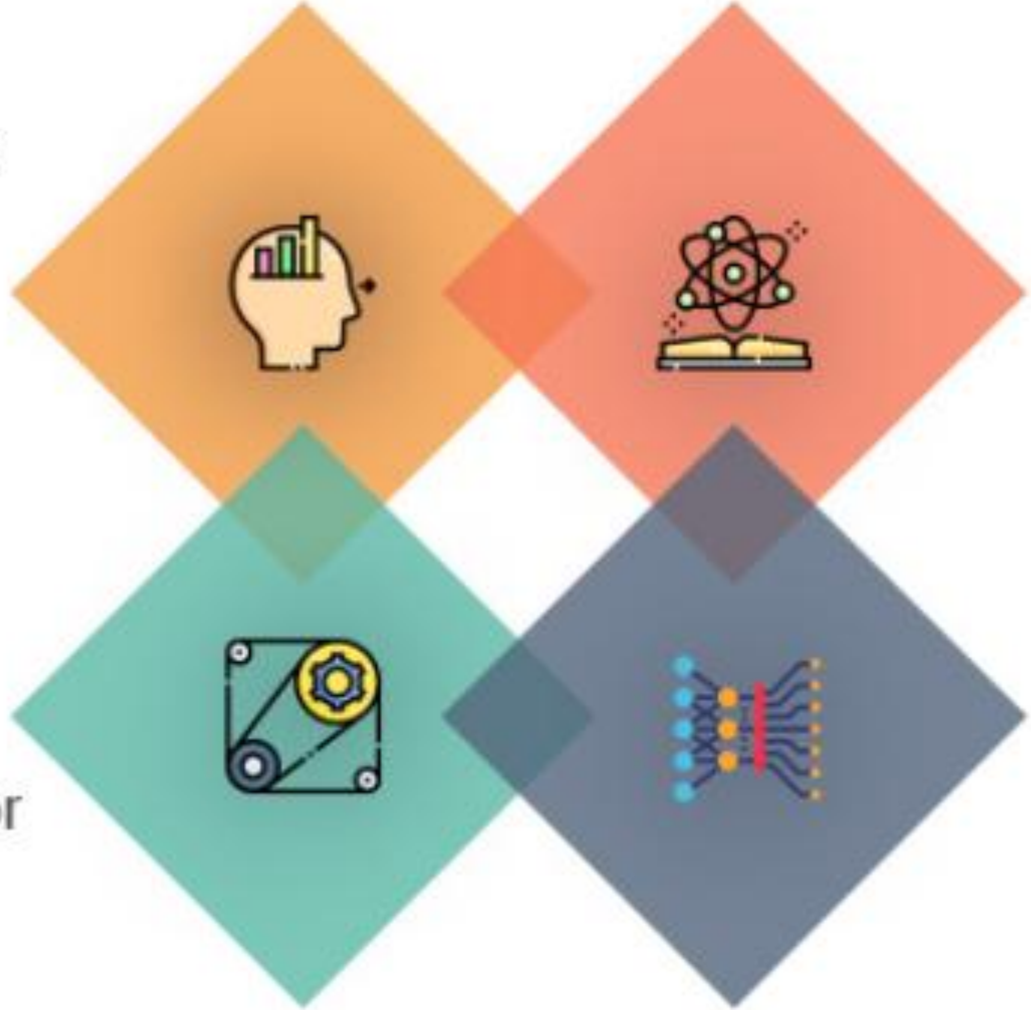


1. Perception

Gathering input from the environment

2. Reasoning and Planning

Using machine learning and LLMs to make decisions



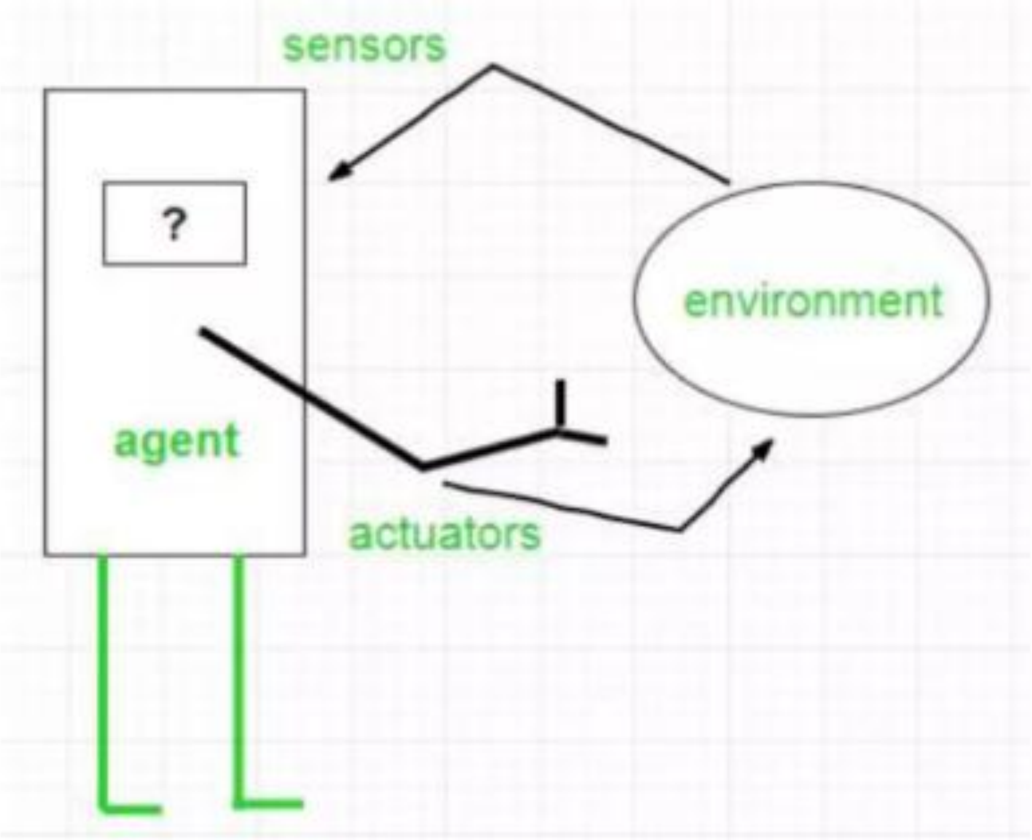
3. Action

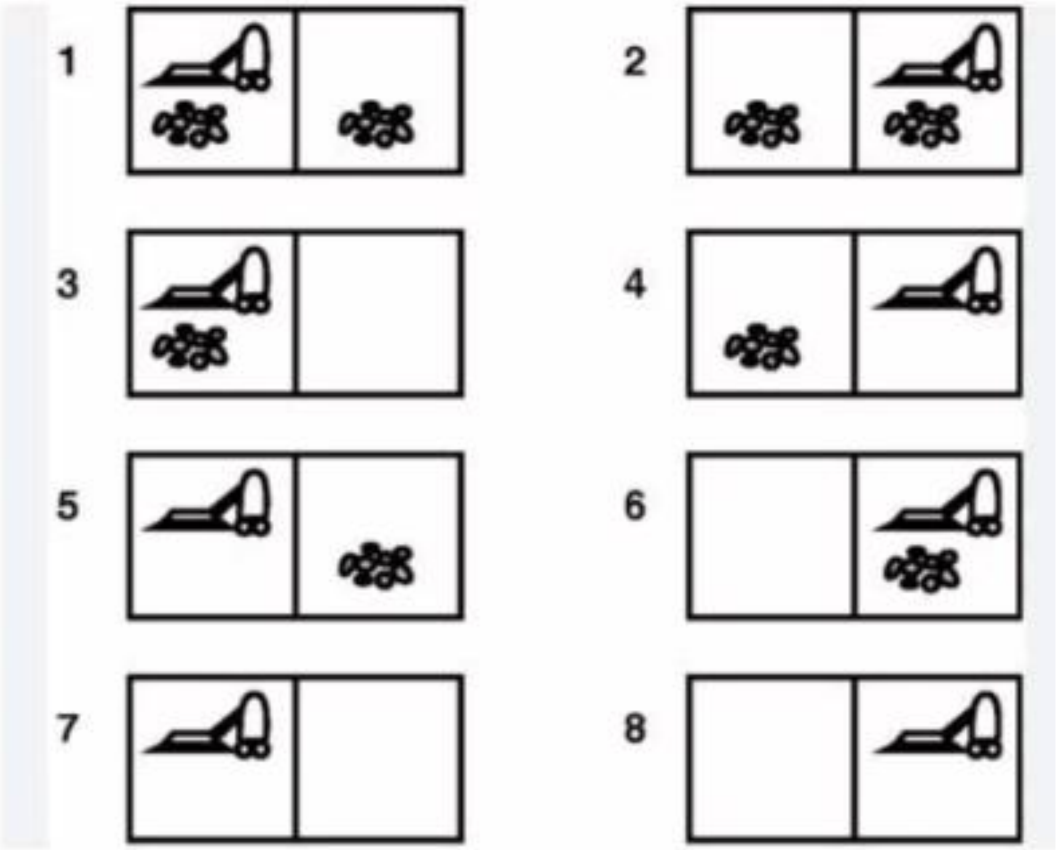
Executing tasks like updating database, or email responses autonomously

4. Learning

Continuously improving through experience

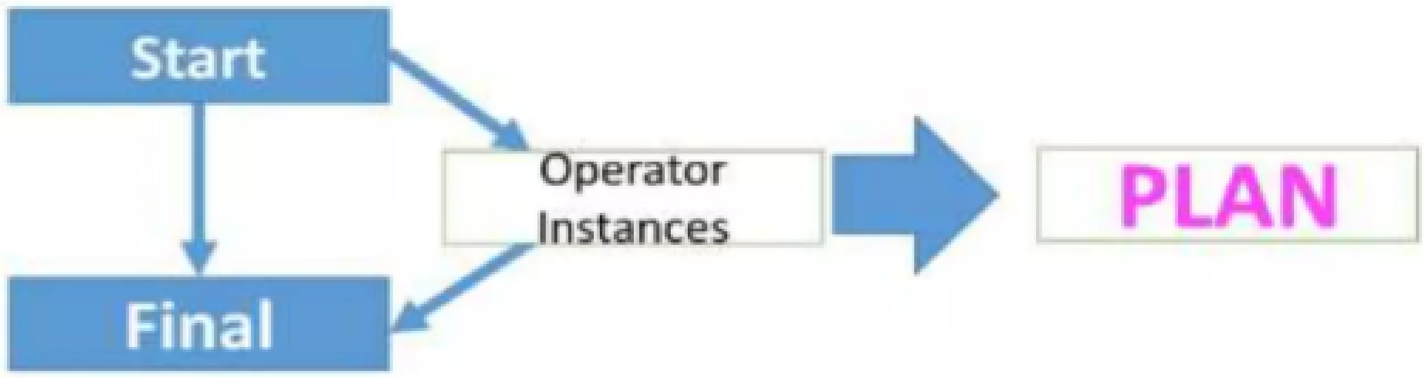
Planning Agent Eg: vacuum cleaner





Purpose of Planning

- The purpose of planning is to **find a sequence of actions** that achieves a given **goal** when performed starting in a given state.
- In other words, given a set of operator instances (defining the possible primitive actions by the agent), an initial state description, and a goal state description or predicate, the planning agent computes a plan.
- Start Final Operator Instances PLAN.



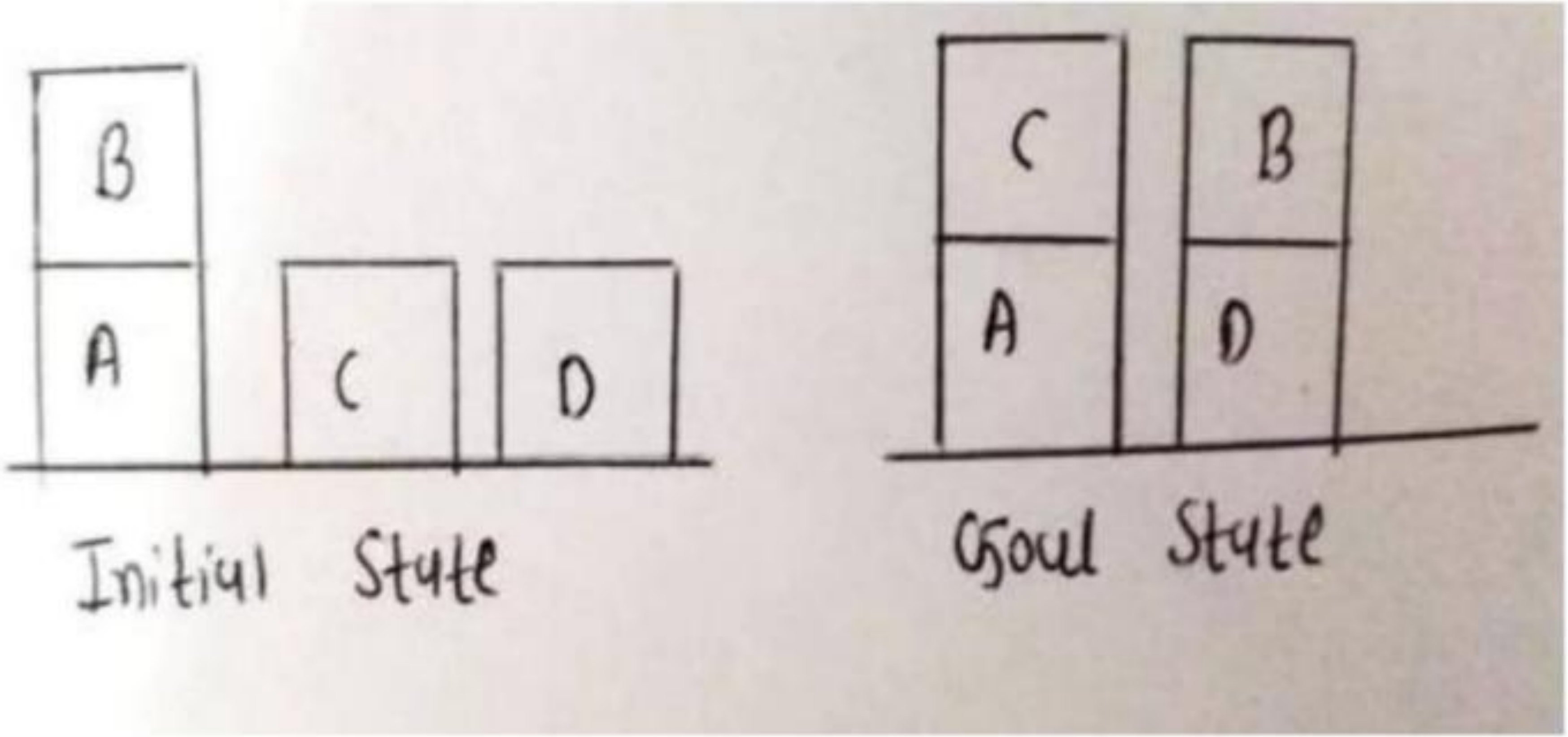
Type of Environments

- **fully observable**
  - we see everything that matters
- **deterministic**
  - the effects of actions are known exactly
- **static**
  - no changes to environment other than those caused by agent actions
- **discrete**
  - changes in time and space occur in quantum amounts
- **single agent**
  - no competition or cooperation to account for

What is plan?

- The task of coming up with a **sequence of actions that will achieve a goal** is called Planning.
- Planning Problems so far:
  - **search-based problem solving**
  - **logical planning**
- Consider only environment that are **fully observable, deterministic, finite, static (change happens only when the agent acts), and discrete (in time, action, objects and effects).** These are called **Classical Planning.**

State and Goal: Classical planning



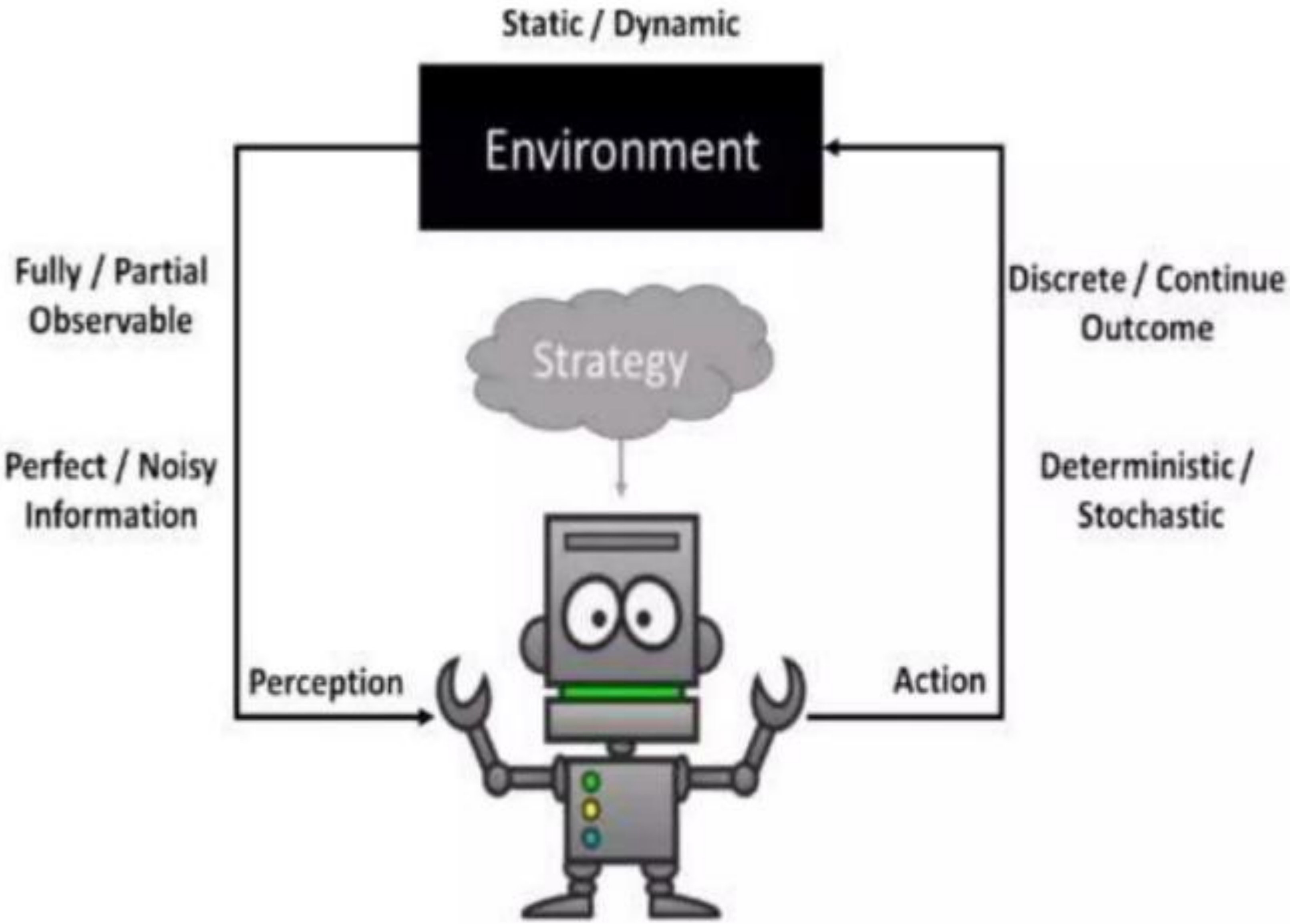
Non classical planning

- **Non classical** planning is for partially observable or stochastic environments and involves a different set of algorithms and agent designs.

Problem Solving Agent + Knowledge Based Agent = Planning Agent

Why we need planning?

- **Intelligent** agents must operate in the world.
  - Take intelligent actions
  - Compose actions together to achieve complex goals
- Change the world to suit the needs. Agents need to reason about what the world will be like after executing a sequence of actions
  - **Need to reason about dynamic environment**


Static / Dynamic
Environment
Fully / Partial
Observable
Strategy
Discrete / Continue
Outcome
Perfect / Noisy
Information
Deterministic /
Stochastic
Perception
Action

Planning algorithm

- **Generate** a goal to achieve
- Construct a plan to achieve goal from current state
- Execute plan until finished
- Begin again with new goal

The language planning problems

- Planning algorithms should take advantage of the logical structure of the problem.
- The key is to find a language that is **expressive enough to describe a wide variety problems,** but restrictive enough to allow efficient algorithms to operate over it.
- The problem should be expressed in a suitable logical language.
- Planning is considered different from problem solving because of the difference in the way they **represent states, goals, actions, and the differences in the way they construct action sequences.**

Representation of states

- Planners decompose the world into logical conditions and represent a state as conjunction of positive literals
- **Example;** prepositional logic P V Q

Representation of goals

- A goal is partially specified state, represented as a conjunction of positive ground literals, such as **P** ∧Q.
- A propositional state s satisfies a goal g if s contains all the atoms in g.
- **Example;** P∧ ***Q*** ∧R satisfies the goal **P** ∧Q

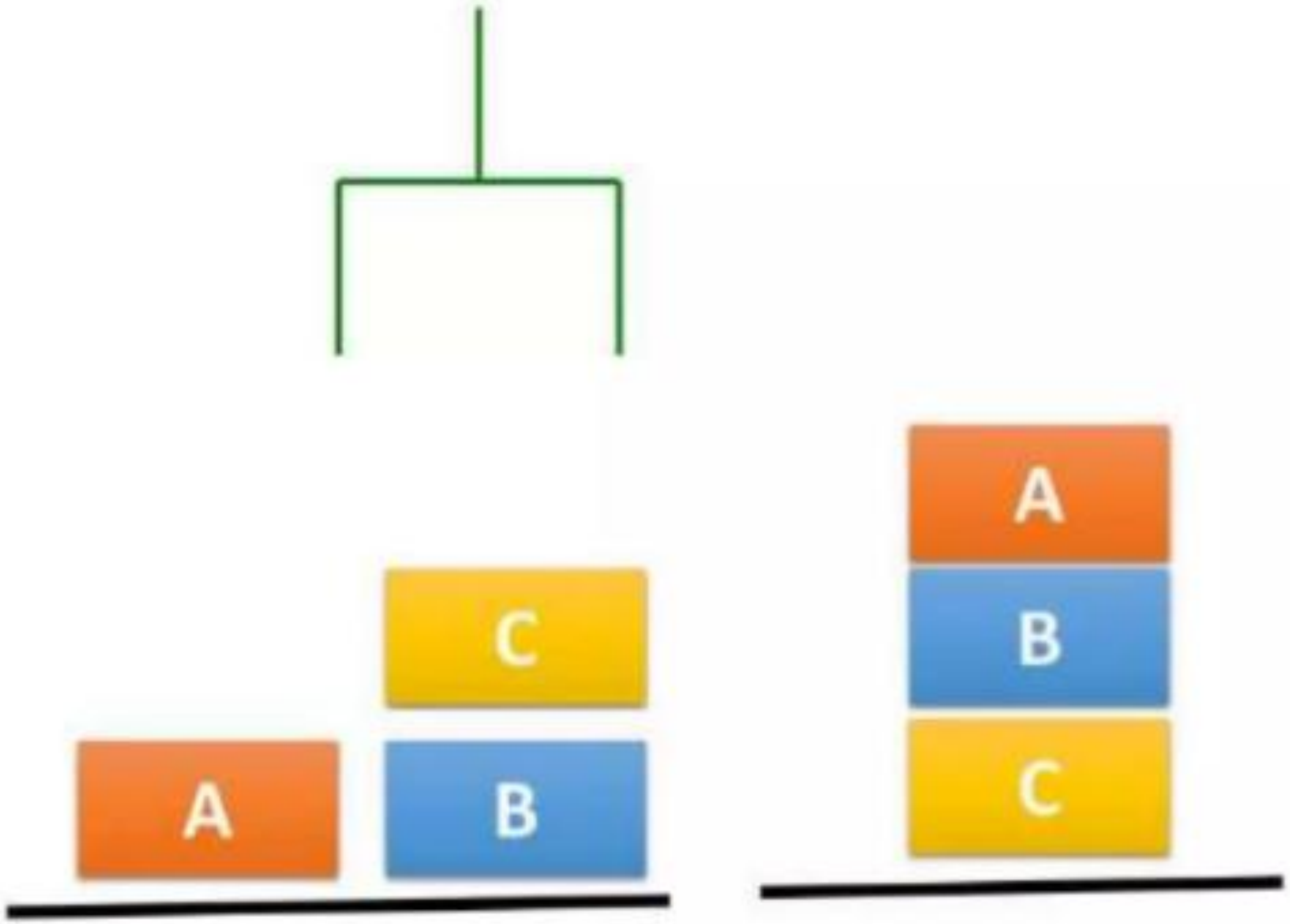
Representation of actions

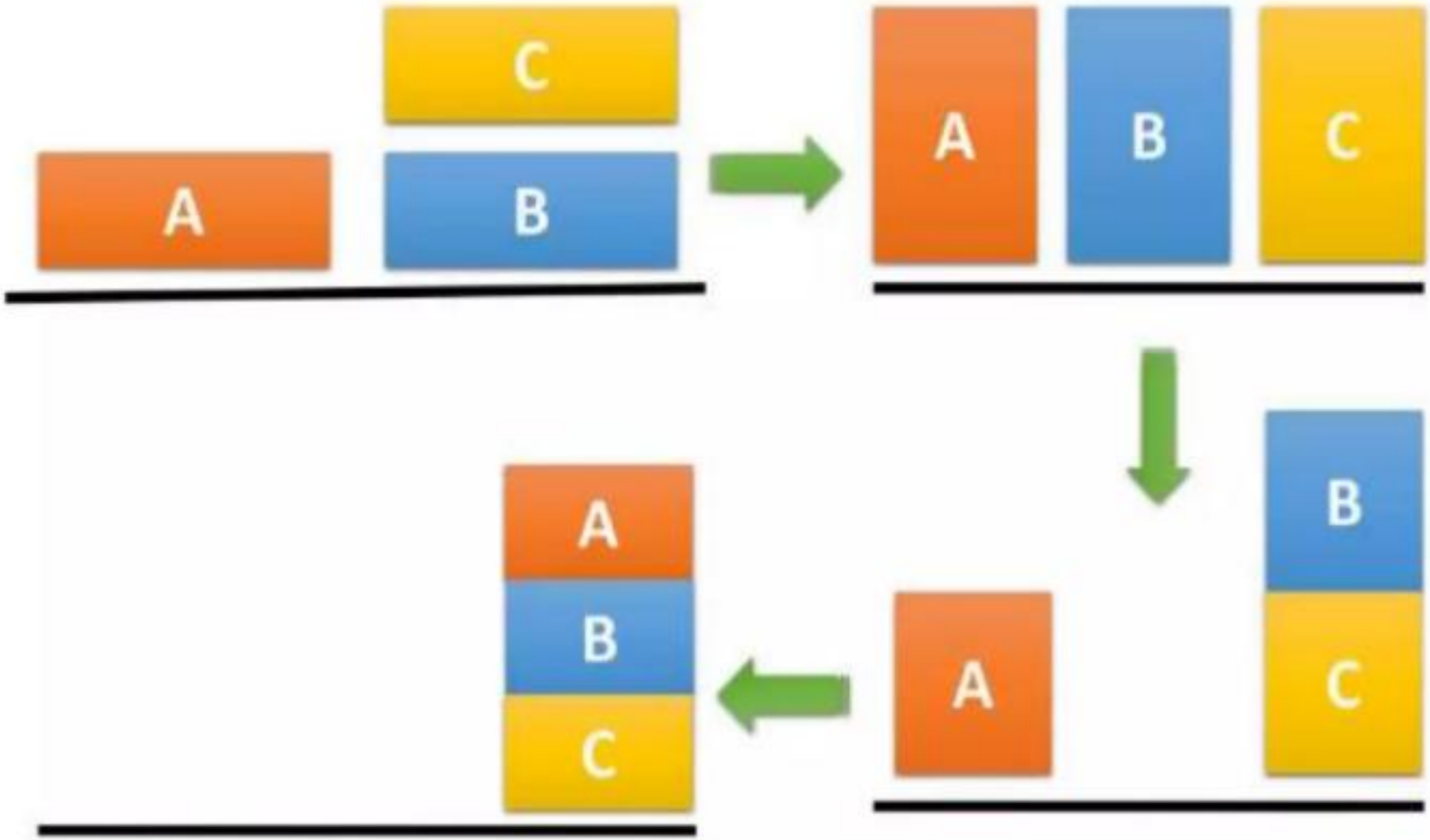
- An action is specified in terms of the pre conditions that must hold before it can be executed and the effects that ensue when it is executed.
- **Example:** Action(Fly(p,From,To),
- PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to), EFFECT: ~At(p,from) ∧ At(p,to))

Action schema

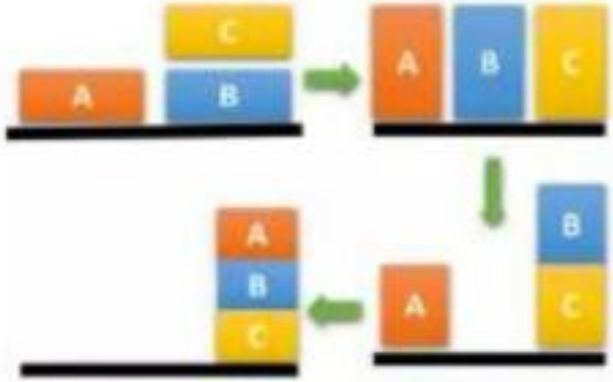
- Action schema, meaning that **it represents a number of different actions that can be derived by instantiating the variables p, from and to** different constants. In general action schema consists of three parts:
  - The **action name and parameter list**
  - The **precondition** is a conjunction of function-free positive literals stating what must be true in state before the actions can be executed.
  - The **effect** is a conjunction of function-free literals describing how **the state changes when the action is executed.**

Example : Box world




C
A
B
A
B
C
B
A
C
A
B
C

Operations



•Op{Action: unstack(C,B)}

•Op{Action: pickup(B)}

•Op{Action: stack(B,C)}

•Op{Action: pickup(A)}

•Op{Action: stack(A,B)}