The Planning Problem

- ***Problem : Get Books from store.***
- ***Given:***
  a) ***Initial State – The agent is at home without book.***
  b) ***Output State - The agent is at home with book.***
- ***Predicate Calculus Convert***
  a) ***States : At(x)***
  b) ***Actions: Go(y), Buy (z)***

Key Ingredients of Planning

Planning is in a form of

- **Problem->Language->Planner->Solution**

Models for defining , classifying, & understanding problems.

a) **What is a planning problem**

b) **What is a solution(plan)**

c) **What is optimal solution**

Languages – for representing problems.

Algorithm – for solving them.

What is planning in artificial intelligence?

- The planning in Artificial Intelligence is about the decision making tasks performed by the robots or computer programs to achieve a specific goal.
- The execution of planning is about choosing a sequence of actions with a high likelihood to complete the specific task.
- In Other way,
- Actions are given as logical descriptions of Actions are given as logical descriptions of preconditions and effects.

What are the different types of planning in AI?

- There are two main approaches to planning:
- state-space planning and
- plan-space planning, also called partial-order planning (POP)

Properties of Planning Algorithm

a) Complete - A planning algorithm is said to be complete, if a solution can be found whenever one actually exist.

b) Optimal – If the order in which solutions are found is consistent with some measure of plan quality.

c) Strictly complete – If ALL the solutions are included in search space.

Linear v/s Non-linear Planning

- Linear Planning is planning or scheduling of project management tasks where distance is a significant factor in the project. Examples of projects include roads, rail, pipeline and transmission lines. Linear planning considers not only the time factors of a task but also the location factors.
- Non-linear planners are able to maintain the emerging plan as a partially- ordered network of actions. Unnecessary ordering (or linearisation) of the actions is avoided. Only when there are conflicts between parallel branches of the plan (such as the inability to determine the answer to a query) is an ordering imposed.

Classical Planning **(EXPECTED JUNE 2020)

- There is complete knowledge about the initial state.
- Actions are deterministic with exactly one outcome.
- The solution is a linear plan (a sequence of actions).
- Search "off-line", then execute with "eyes closed".

Partial-Order Planning or Non-linear Planner

- Partial-order planning is an approach to automated planning that maintains a partial ordering between actions and only commits ordering between actions when forced to i.e., ordering of actions is partial. Also this planning doesn't specify which action will come out first when two actions are processed.
- It contrasts with total-order planning, which produces an exact ordering of actions.

Partial order planning builds on these ideas

- **adopts a plan space representation** – states in the state space correspond to partial plans.
- **relaxes the requirement that solutions are constructed sequentially** – steps can be added to the plan in any order.
- adopts the **principle of least commitment** – decisions about step ordering and variable bindings are delayed as long as possible.
- returns a plan in which some steps are ordered and other steps are unordered

Least commitment

- ✓ choices about plan step ordering and operator variable binding are only be made when necessary.
- ✓ reduces backtracking while planning.
- ✓ allows the agent to execute actions in the plan in parallel if it can.

Partial-Order Planning

- Idea: (Non-linear planner)

a) works on several subgoals independently.

b) solves them with subplans.

c) combines the subplans.

d) flexibility in ordering the subplans.

e) least commitment strategy:

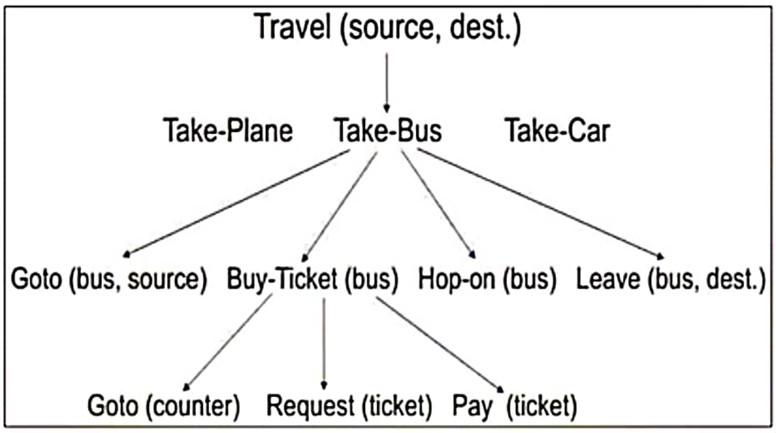
  - delaying a choice during search
  - Example, leave actions unordered, unless they must be sequential

Total-Order Planning or Linear Planner

- Forward/backward state-space searches are forms of totally ordered plan search.
- explore only strictly linear sequences of actions directly connected to the start or goal.
- cannot take advantages of problem decomposition.

HIERARCHICAL PLANNING

Hierarchical planning is a planning method based on Hierarchical Task Network (HTN) or HTN planning.



Hierarchical (HTN) Planning

- In order to solve hard problems, a problem solver may have to generate long plans,
- It is important to be able to eliminate some of the details of the problem until a solution that addresses the main aspect is found.
- Then an attempt can be made to fill the appropriate details.
- Early attempts to do this involved the use of macro operators.
- But in this approach, no details were eliminated from actual descriptions of the operators.

Structure of Hierarchical (HTN) Planning

- Hierarchical Task Networks (from here on abbreviated as HTN) is a hierarchical planning technique where actions are divided into different levels, or hierarchies.
- There are different types of actions residing within these hierarchies, one of them being High Level Actions (from here on abbreviated as HLA).
- These HLAs could be any action that you can divide into smaller actions called refinements. These refinements may be sequences of actions or even other HLAs.
- Refinements in turn are broken down into implementati (also called primitive actions) which refers to any act that has no further refinements or implementations.

Indeterministic Domain

- ✓ But in an **uncertain environment**, on the other hand, an agent must **use its percepts** to discover what is happening.

- ✓ While the plan is being executed and possibly **modify or replace** the plan if something **unexpected happens.**

- ✓ Agents have to deal with both **incomplete** and **incorrect** information.

- ✓ **Incompleteness arises** because the world is partially observable, nondeterministic, or both.

Cont…

- ✓ The possibility of having **complete or correct** knowledge depends on how much **indeterminacy** there is in the world.

- ✓ With **bounded indeterminacy,** actions can have unpredictable effects, but the possible effects can be listed in the action description axioms.

- ✓ For example, **when we flip a coin,** it is reasonable to say that the outcome will be **Heads or Tails.**

- ✓ An agent can cope with bounded indeterminacy by making plans that **work in all possible circumstances**.

Cont…

- ✓With unbounded indeterminacy, on the other hand, the **set of possible preconditions or effects** either is **unknown** or **is too large** to be enumerated completely.

- ✓This would be the case in very complex or dynamic domains such as **driving, economic planning**, and **military** strategy.

3.4.3.1 Planning methods in indeterminacy…

✓ There are **four planning methods** for handling indeterminacy.

✓ The **first two** are suitable for **bounded indeterminacy**, and the **second two** for **unbounded indeterminacy**:

i. **Sensorless Planning - conformant planning**
ii. **Conditional Planning - contingency**
iii. **Execution monitoring and replanning**
iv. **Continuous planning.**

i. Sensorless planning…

- ✓ Also called **conformant planning**, this method constructs standard, sequential plans that are to be executed **without perception.**

- ✓ The sensorless planning algorithm must ensure that the plan achieves the goal in all possible circumstances, **regardless of the true initial state** and the actual action outcomes.

ii. Conditional Planning…

- ✓ Also known as **contingency** planning, this approach deals with **bounded indeterminacy** by constructing a conditional plan with different branches for the different contingencies that could arise.

- ✓ Just as in classical planning, the agent plans first and then executes the plan that was produced.

- ✓ The agent finds out **which part of the plan** to execute by including sensing actions in the plan to test for the **appropriate conditions.**

iii. Execution monitoring and replanning…

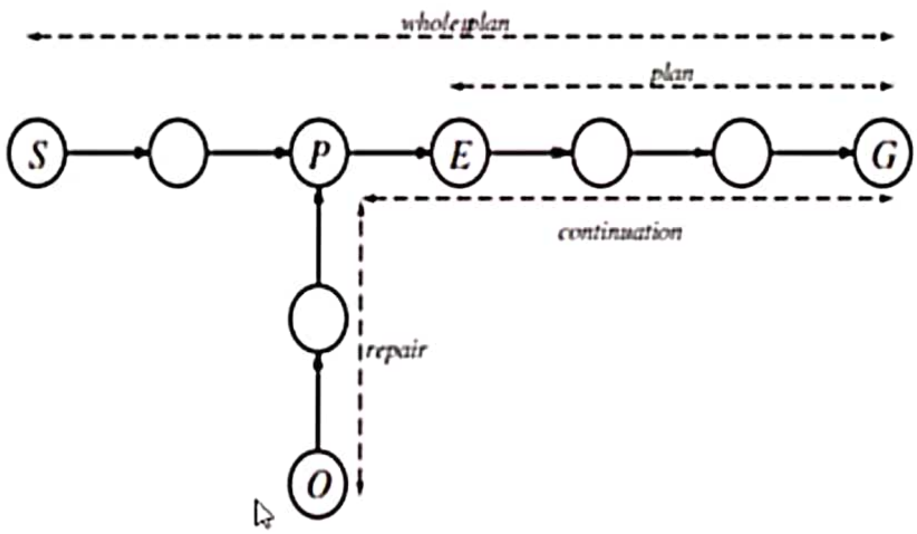


Figure 11.9 Before execution, the planner comes up with a plan, here called *whole plan*, to get from S to G. The agent executes steps of the plan until it expects to be in state E, but observes it is actually in O. The agent then replans for the minimal repair plus continuation to reach G.

Cont…

- ✓ In this approach, the **agent can use** any of the preceding planning techniques **(classical, sensorless, or conditional)** to construct a plan.

- ✓ But it also uses execution monitoring to judge whether the plan has a provision for the actual current situation or need to be revised.

- ✓ **Replanning occurs** when something goes **wrong.**

- ✓ In this way, the agent can **handle unbounded indeterminacy.**

iv Continuous planning…

- ✓ All the planners we have seen so far are designed to achieve a goal and then stop.
- ✓ A continuous planner is designed **to persist over a lifetime.**
- ✓ It can **handle unexpected circumstances** in the environment, even if these occur while the agent is in the middle of constructing a plan.
- ✓ It can also handle the **abandonment of goals** and the creation of additional goals by goal formulation.

Defining Generative AI

To understand generative artificial intelligence (GenAI), we first need to understand how the technology builds from each of the AI subcategories listed below.

Artificial Intelligence

The theory and methods to build machines that think and act like humans.

Expert System AI

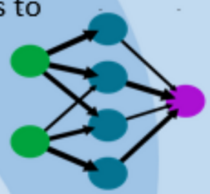
Programmers teach AI exactly how to solve specific problems by providing precise instructions and steps.

Machine Learning

The ability for computers to learn from experience or data without human programming.

Deep Learning

Mimics the human brain using artificial neural networks such as **transformers** to allow computers to perform complex tasks.



Generative AI

Generates new text, audio, images, video or code based on content it has been **pre-trained** on.

ChatGPT *Midjourney* *Bard*

AI for Education

aiforeducation.io

What is generative AI?

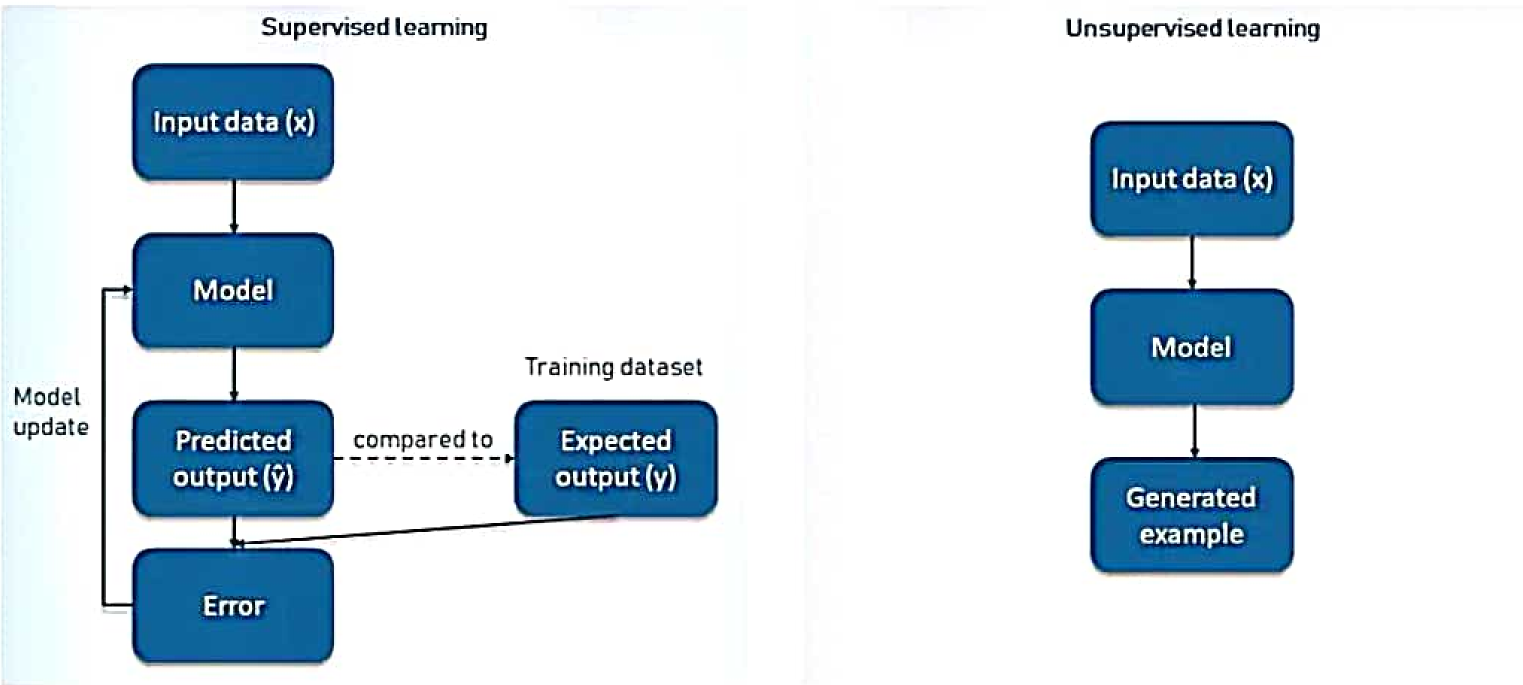
Generative AI refers to unsupervised and semi-supervised machine learning algorithms that enable computers to use existing content like text, audio and video files, images, and even code to create new possible content. The main idea is to generate completely original artifacts that would look like the real deal.

Discriminative & Generative Model

Discriminative modeling is used to classify existing data points (e.g., images of cats and guinea pigs into respective categories). It mostly belongs to supervised machine learning tasks.

Generative modeling tries to understand the dataset structure and generate similar examples (e.g., creating a realistic image of a guinea pig or a cat). It mostly belongs to unsupervised and semi-supervised machine learning tasks.

SUPERVISED AND UNSUPERVISED LEARNING



Discriminative modeling

Most machine learning models are used to make predictions. Discriminative algorithms try to classify input data given some set of features and predict a label or a class to which a certain data example (observation) belongs.

Say we have training data that contains multiple images of cats and guinea pigs. These are also called samples. Each sample comes with a set of features (X and a class label (Y), which, in our case, can be "cats" or "guinea pigs." We also have a neural net aiming to understand how to discriminate between two classes and predict the probability of a given sample belonging to one of them.

During the training, each prediction ($\hat{y}$) is compared to the actual label (Y). Based on the difference between the two values, the model gradually learns the relationships between features and classes and correlates its results.

Generative modeling

Generative algorithms do the complete opposite — instead of predicting a label given to some features, they try to predict features given a certain label. Discriminative algorithms care about the relations between X and Y; generative models care about how you get X from Y.

Mathematically, generative modeling allows us to capture the probability of x and y occurring together. It focuses on learning features and their relations to get an idea of what makes cats look like cats and guinea pigs look like guinea pigs. As a result, such algorithms not only distinguish the two animals but also recreate or generate their images.

GAN architecture.

In their architecture, GANs have two deep learning models:

- generator — a neural net whose job is to create fake input or fake samples from a random vector (a list of mathematical variables with unknown values); and
- discriminator — a neural net whose job is to take a given sample and decide if it’s a fake from a generator or a real observation.