AI Planning and Research

Planning is a sub field of Artificial Intelligence which is being researched for more than three decades. Planning techniques have been applied in a variety of tasks including Robotics, process planning, autonomous agents and space mission control systems. Planning is the key to an intelligent machine, with reasoning capabilities. It can be considered as a special type of Search problem which is defined by a State space – set of states where we search for a solution, Initial State – the state we search from, Operators which map a state to a new state and a Goal condition.

Researchers have applied various methods in solving a planning problem. They can be grouped into

1. State space search
2. Situation calculus based on FOL(First Order Logic)
3. STRIPS(Stanford Research Institute Problem Solver)
4. Partial Order planning or Hierarchial Task Network.
5. Probabilistic Planning

State Space Search: A state space consists of all states that be reached from an initial state by taking some sequence of actions. A state can be represented by a search node. So here the basic idea is to start from the initial state, test for the goal state and then expanding its successors and their successors by applying some operators until we reach the Goal state. The order of expanding the nodes depends on the Search strategy used.

Situation Calculus: The situation calculus is a first Order logic language for representing changes. It was first introduced by McCarthy in 1963, and described in further details by McCarthy and Hayes [4] in 1969. It consists of situations, actions and fluents. According to McCarthy and Hayes [4], a situation is “the complete state of the universe at an instance of time”. Actions are what changes the world from one situation to another situation. A fluent is used to describe the state. Fluents can be relational where it can take values True/ False or functional where it can take a range of values. For instance, to say that a plane p is in airport x in situation s, one would use a binary predicate like “At” and write At(p,x,s). The change in a situation by an action can be described by effect and frame axioms. Effect axioms describe how actions change the situations in which they are applied into new situations while frame axioms describe what aspects of a situation remain unaffected as actions are applied. The planning problem is to find the sequence of actions that lead to a goal which was reduced to theorem solving. The problem in Situation Calculus is its large search space and large number of axioms to be defined for one action and the proof may not lead to its shortest path.

STRIPS: Planning emerged as a specific sub-field with the seminal work of Fikes and Nilsson[Fikes/Nilsson 1971] on the Stanford Research Institute Problem Solver(STRIPS). The Strips system was used to plan the motion of a robot called “ Shakey “ and to control it as it pushed a set of boxes through a number of interconnecting rooms. A STRIPS planning problem specifies :

1. An initial state which is a set of well formed formulas(wff) describing the present state of the world
2. A Goal state stated as a wff
3. A set of STRIPS actions described by a set of operators, including a description of their effects and their precondition wff schemata. The effect of the action is based on the STRIPS assumption: All the primitive features not mentioned in the description of the action stay unchanged. E.g.The action flying an aeroplane from airport x to airport y can be represented in STRIPS by:

Action(Fly(p,x,y)

PRECOND: Plane(p) Λ Airport(x) Λ Airport(y) Λ At(p,x)

Add list : At(p,y) delete list: At(p,x)

There can be two procedures of Planning with STRIPS like representation – Progressive and Regressive. Progressive Planning searches forward from a given state to find the sequence of operators which satisfies the goal. Regressive Planning searches backward from the goal for a sequence of operations that will reduce the goal to a state that is satisfied in the initial state. A common language for writing STRIPS domain and problem sets is the Planning Domain Definition Language (PDDL). With STRIPS AI planning, a *planning graph* can be constructed that contains all available states and the actions that bring you to each state. As with most trees and graphs, we can traverse them using a variety of algorithms like breadth-first-search, depth-first-search, and the most intelligent approach - A* search. The concentration in most of the work on planning has been on generating plans from scratch, not learning from experience.

<u>*Partial Order planning or Hierarchial Task Network*</u>: With state-space planning, we generate plans by searching through state space like in STRIPES. In Partial plans we also record the rationale behind each action, e.g. to achieve the precondition of another action. In aggregate these partial plans may form the solution to the problem (i.e. they act as component plans). Instead of achieving goals, we want to accomplish *tasks*. Tasks are high-level descriptions of some activity we want to execute - this is typically accomplished by decomposing the high-level task into lower-level subtasks. This is the approach for **Hierarchical Task Network (HTN) planning**. There is a simpler version called STN(Simple Task Network) planning as well. Rather than searching through the state-space, we search through *plan-space* - a graph of partial plans. The nodes are partially-specified plans, the arcs are *plan refinement operations* (which is why this is called *refinement planning*), and the solutions are *partial-order plans*.

<u>*Probabilistic Planning*</u>: Thus far all the approaches have assumed the outcome of actions are deterministic. However, the outcome of actions are often uncertain, so the resulting state is uncertain. If we can quantify the degree of uncertainty (i.e. we know the probabilities of different outcomes given an action) we have, we can use **probabilistic planning**. Probabilistic planning is naturally formulated using Markov Decision Processes (MDP) (Puterman 1994) and many probabilistic planning techniques have been developed based on MDP formulations Instead of simple state transition systems, we use a *partially observable Markov decision process* (POMDP) as our model.

References:

1. Erol, K., Hendler, J. & Nau, D.S. Ann (1996). Annals of Mathematics and Artificial Intelligence March 1996, Volume 18, Issue 1, pp 69–93
2. https://web.stanford.edu/class/cs227/Lectures/lec16.pdf accessed on 8/12/2017
3. J. McCarthy and P. Hayes.(1971) Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors. Machine Intelligence, vol. 4, pages 463–502. Edinburgh University Press, Edinburgh, 1969
4. J. McCarthy and P. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, Machine Intelligence, 4:463–502. Edinburgh University Press, 1969
5. Peter Norvig, Stuart J. Russell (1994). Artificial Intelligence: A Modern Approach
6. Puterman, M. 1994. Markov Decision Processes. Wiley, New York
7. Richard E. Fikes, Nils J. Nilsson (Winter 1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving" (PDF). Artificial Intelligence. 2 (3–4): 189–208.
8. R. Reiter (1991). The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In Vladimir Lifshitz, editor, Artificial intelligence and mathematical theory of computation: papers in honour of John McCarthy, pages 359–380, San Diego, CA, USA. Academic Press Professional, Inc. 1991.