Research Review by Kenan Sooklall

Planning is the concept of taking an agent in state X and getting it to state Y, where X is the initial state and Y is the goal state. There can be many intermediate states between between X and Y thus making search for an optimal path difficult. The agent can move around and perform actions that are allowed in a certain state. Throughout the years many researchers have collaborated to tackle the problem of planning and solving NP-Hard problems in general. Below are some historical developments in planning.

STATE-SPACE:
Explicit state-space search is the earliest and most straightforward method for solving some of the most important problems about planning. These include uninformed search algorithms like breadth-first search and depth first search along with systematic heuristic algorithms with optimality guarantees like A* with variants like Iterative Deepening A*. However use of these methods are highly dependent on the size of the search space and are not always the optimal approach.

STRIDES:
The Stanford Research Institute Problem Solver also knows as STRIPS was developed by Richard Fikes and Nils Nilsson in 1971 at SRI International. A STRIPS instance is composed of an initial state, the specification of the goal states and a set of actions. For each action there is preconditions, what must be established before the action is performed and a postconditions, what is established after the action is performed. Below is an example of STRIDES in action taken from wikipedia:

```
Initial state: At(A), Level(low), BoxAt(C), BananasAt(B)
Goal state:    Have(bananas)

Actions:
            // move from X to Y
            _Move(X, Y)_
            Preconditions:  At(X), Level(low)
            Postconditions: not At(X), At(Y)

            // climb up on the box
            _ClimbUp(Location)_
            Preconditions:  At(Location), BoxAt(Location), Level(low)
            Postconditions: Level(high), not Level(low)

            // climb down from the box
            _ClimbDown(Location)_
            Preconditions:  At(Location), BoxAt(Location), Level(high)
            Postconditions: Level(low), not Level(high)

            // move monkey and box from X to Y
```

```
_MoveBox(X, Y)_
Preconditions:  At(X), BoxAt(X), Level(low)
Postconditions: BoxAt(Y), not BoxAt(X), At(Y), not At(X)

// take the bananas
_TakeBananas(Location)_
Preconditions:  At(Location), BananasAt(Location), Level(high)
Postconditions: Have(bananas)
```

PARTIAL-ORDER PLANNING (POP):

The construction of POP was pioneered by the NOAH planner and by the NONLIN system. POP is an approach to automate planning that leaves decisions about the ordering of actions as open as possible. Given a problem, a sequence of actions and a goal, a POP plan specifies all actions that need to be taken, but doesn’t specify the ordering of the actions unless necessary. For example, planning to build a deep learning computer. A POP would be:

- Go to an electronic website
- Get motherboard, gpu, cpu, power supply, etc
- Assemble items

That is a POP because the order of finding all the components is not specified, the agent can wander around the web collecting all items on the list until the list is complete.

GRAPHPLAN:

Graphplan was created by Avrim Blum and Merrick Furst in 1995. Graphplan takes as input a planning problem expressed in STRIPs and produces, if possible, a sequence of operations for reaching a goal state. The graph has the property that useful information for constraining search can quickly be propagated through the graph as it is being built. GP then exploits this information in the search for a path to the goal. GP always returns a shortest partial-order plan.

References

1. Norvig, P. and Russel, S. (2010). Artificial Intelligence: A Modern Approach. Third Edition.
2. https://en.wikipedia.org/wiki/STRIPS