**Research review**

The part of this project I will discuss these three important developments in artificial intelligence such as Partial order planning, the Stanford Research Institute Problem Solver (STRIPS), and binary decision diagram.

Partial-order planning is an approach to automated planning that leaves decisions about the ordering of actions as open as possible. It contrasts with total-order planning, which produces an exact ordering of actions. Given a problem in which some sequence of actions is required in order to achieve a goal, a partial-order plan specifies all actions that need to be taken, but specifies an ordering of the actions only where necessary [1]. The idea underlying partial-order planning includes the detection of conflicts (Tate, 1975a) and the protection of achieved conditions from interference (Sussman, 1975). The construction of partially ordered plans (then called task networks) was pioneered by the NOAH planner (Sacerdoti, 1975, 1977) and by Tate's (1975b, 1977) NONLINE system [2].

The STRIPS was developed by Richard Fikes and Nils NIllson in 1971 for the purpose of automated planning. It was designed as an actual planner but it is the language which represents its most significant contribution to AI. The STRIPS language became the basis for most automated planning languages currently in use. This includes both ADL and PDDL. A strips instance consists of a familiar scheme. That is, an initial state, specified goal states, and a set of actions. For each action in the action set preconditions must be established prior to the action and post conditions must be established after the action. A critical restriction of STRIPS, which limits its utility, is the assumption of perfect knowledge of the initial state. That is, any condition not in the initial state is deemed False [3]. This is not always the case for planning problems and in those cases this assumption could prevent valid actions from being explored.

A binary decision diagram (BDD) or branching program is a data structure that is used to represent a Boolean function. On a more abstract level, BDDs can be considered as a compressed representation of sets or relations [4]. There has been interest in the representation of plans as binary decision diagrams, compact data structures for Boolean expressions widely studied in the hardware verification community (Clarke and Grumberg, 1987; McMillan, 1993). There are techniques for proving properties of binary decision diagrams, including the property of being a solution to a planning problem. Cimatti et al. (1998) present a planner based on this approach. Other representations have also been used; for example, Vossen et al. (2001) survey the use of integer programming for planning [5].

References

[1] https://en.wikipedia.org/wiki/Partial-order\_planning

[2] Russell, Stuart and Norvig, Peter. Artificial Intelligence: A Modern Approach 3nd Edition, P.394.

[3] <https://en.wikipedia.org/wiki/STRIPS>

[4] <https://en.wikipedia.org/wiki/Binary_decision_diagram>

[5] Russell, Stuart and Norvig, Peter. Artificial Intelligence: A Modern Approach 3nd Edition, P.395-396.