Maryam Saeedmehr
Stud. NO. : 9629373

Critical Summary of Chapter 6 of
"Python code for Artificial Intelligence: Foundations of Computational Agents"

Chapter 6 : planning with certainty

This Chapter contains 5 parts. The first part is about representing actions and planning problems. It explains about The STRIPS representation of an action and then it solves two problems called robot delivery and blocks world.
The blocks world consist of blocks and a table. Each block can be on the table or on another block. A block can only have one other block on top of it. A state is defined by the two features:

- on where on (x) = y when block x is on block or table y
- clear where clear (x) = True when block x has nothing on it.

There is one parameterized action

- move (x, y, z) move block x from y to z, where y and z could be a block or the table. To handle parameterized actions (which depend on the blocks involved), the actions and the features are all strings, created for the all combinations of the blocks. Note that we treat moving to a block separately from moving to the table, because the blocks needs to be clear, but the table always has room for another block.

Section 6.2 is about forward planning. In a forward planner, a node is a state. A state consists of an assignment, which is a variable:value dictionary. In order to be able to do multiple-path pruning, we need to define a hash function, and equality between states.
Each planning domain requires its own heuristics. If you change the actions, you will need to reconsider the heuristic function, as there might then be a lower-cost path, which might make the heuristic non-admissible.
Note that the current state is a complete description; there is a value for
every feature. However the goal need not be complete; it does not need to define a value for every feature. Before checking the value for a feature in the goal, a heuristic needs to define whether the feature is defined in the goal.
In section 6.3 ,we discuss regression planning. In regression planner a node is a sub-goal that need to be achieved. A sub-goal object consists of an assignment, which is variable:value dictionary. We make it hashable so that multiple path pruning can work. The hash is only computed when necessary (and only once). A regression search has sub-goals as nodes. The initial node is the top-level goal of the planner. The goal for the search (when the search can stop) is a sub-goal that holds in the initial state.
Section 6.4 explains planning as CSP. A CSP can use any of the CSP algorithms to solve (e.g., stochastic local search or arc consistency with domain splitting).This assumes the same action representation as before; we do not consider factored actions (action features), nor do we implement state constraints.
Section 6.5 discuss the Partial-Order Planning. A partial order planner maintains a partial order of action instances. An action instance consists of a name and an index. We need action instances because the same action could be carried out at different times.