PAR. Lab Exercise 1

Blocks World

The blocks world is one of the most famous planning domains in AI. Given a set of blocks with similar shapes and different colours sitting on a table. The goal is to build one or more vertical stacks of blocks. The catch is that only one block may be moved at a time: it may either be placed on the table or placed atop another block using a robot arm. Because of this, any blocks that are, at a given time, under another block cannot be moved.

World Block description

What is the Blocks World? -- The world consists of:

- A flat surface such as a tabletop
- An adequate set of identical blocks which are identified by letters.
- The blocks can be stacked one on one to form towers of apparently unlimited height.
- The stacking is achieved using a robot arm (i.e., lift) which has fundamental operations and states which can be assessed using logic and combined using logical operations.
- The robot can hold one block at a time and only one block can be moved at a time.

Actions	Predicates
UNSTACK(A,B) -- pick up clear block A from block B; **STACK(A,B)** -- place block A using the arm onto clear block B; **PICKUP(A)** -- lift clear block A with the empty arm; **PUTDOWN(A)** -- place the held block A onto a free space on the table.	**ON(A,B)** -- block A is on block B. **ONTABLE(A)** -- block A is on the table. **CLEAR(A)** -- block A has nothing on it. **HOLDING(A)** -- the arm holds block A. **ARMEMPTY** -- the arm holds nothing.

For the problem shown in the figure above, note: S0 is the initial state and g is the goal below, find the domain and problem PDDL files for solving the block word problem.

Domain and Problem PDDL files

How to define the domain pddl file:

Domain PDDL file:

```
(define domain blocksworld)
 (:requirements :strips :equality)
 (:predicates (clear ?x)
              (on-table ?x)
              (arm-empty)
              (holding ?x)
              (on ?x ?y))
 (:action pickup
   :parameters (?ob)
   :precondition (and (clear ?ob) (on-table ?ob) (arm-empty))
   :effect (and (holding ?ob) (not (clear ?ob)) (not (on-table ?ob))
                (not (arm-empty))))
 (:action putdown
   :parameters  (?ob)
   :precondition (and (holding ?ob))
   :effect (and (clear ?ob) (arm-empty) (on-table ?ob)
                (not (holding ?ob))))
 (:action stack
   :parameters  (?ob ?underob)
   :precondition (and  (clear ?underob) (holding ?ob))
   :effect (and (arm-empty) (clear ?ob) (on ?ob ?underob)
                (not (clear ?underob)) (not (holding ?ob))))
 (:action unstack
   :parameters  (?ob ?underob)
   :precondition (and (on ?ob ?underob) (clear ?ob) (arm-empty))
   :effect (and (holding ?ob) (clear ?underob)
                (not (on ?ob ?underob)) (not (clear ?ob)) (not (arm-empty)))))
```

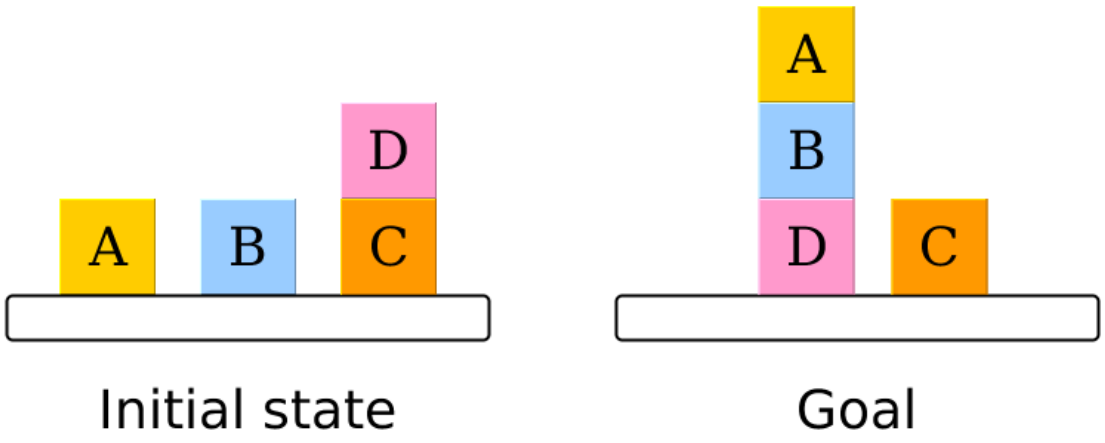
How to define the problem pddl file:

Problem PDDL file:

```
  (define (problem pb1)
   (:domain blocksworld)
      (:objects a b c)
      (:init (on-table a) (on-table b)   (on-table c)
             (clear a)  (clear b) (clear c) (arm-empty))
      (:goal (and (on a b) (on b c))))
```

Exercises:

1. Practise with Visual Studio code/Online PDDL editor to write and execute the two PDDL files above.
2. Create another problem file for the following problem:



3. Create another problem file with the following initial and goal states: