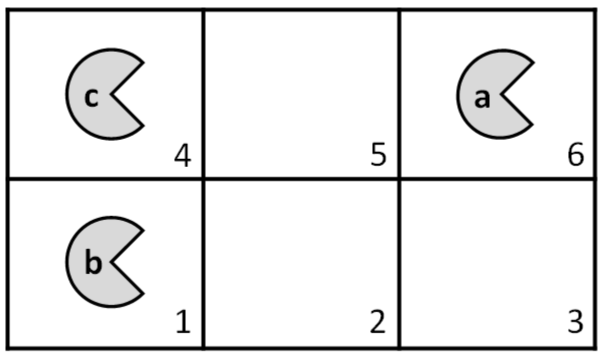
Given is the planning domain below. The rules are the same as you mentioned during lectures: (1) the robots (or PacMen) can move up, down, left, or right into adjacent cells, (2) there can be at most one robot in any cell at any time, and (3) the move cannot start if the target cell is not empty at that moment (e.g. moving **b** up is not possible even if we now that **c** is currently moving to the right). The two relations in the domain are at(X,Y) meaning robot X is in cell Y and c(X) meaning cell X is clear. The only available action type is m(X,A,B) meaning robot X moves from cell A to cell B.

The goals for this problem are: [at(a,1), at(b,2), at(c,3)].



(1) These questions are regarding goal regression means-ends planning.
   a. State the formula for goal regression, i.e. how does one calculate the regressed goals through an action.
   b. Find all actions A that a goal regression planner would consider in the first step when regressing given goals for this problem. The actions should be free of variables, i.e. list concrete fully specified actions, e.g. m(b,1,2).
   c. Regress goals [at(a,1), at(c,3), c(4), c(5)] through action m(c,2,3).
   d. Regress goals [at(a,1), at(c,3), c(4), c(5)] through action m(a,X,Y). List all possibilities.

(2) These questions are regarding the POP planner.
   a. Find all **optimal** plans the POP planner would return.
   b. What actions would the POP planner consider on the first step? List all possibilities.
   c. Simulating the POP planner means at each step selecting an action A with an open precondition P and selecting an action B that achieves P. It further means keeping track of the causal links and of the order constraints, including resolving possible ordering conflicts. For a possible end result see slide 19 of the POP planner lecture. Simulate the following first seven steps a POP planner might consider **(follow these selections at each step)**: (1) A=FINISH, P=at(b,2), B=m(b,5,2); (2) A=FINISH, P=at(a,1), B=m(a,2,1); (3) A=m(a,2,1), P=at(a,2), B=m(a,3,2); (4) A=m(b,5,2), P=c(2), B=m(a,2,1); (5) A=FINISH, P=at(c,3), B=m(c,6,3); (6) A=m(c,6,3), P=c(3), B=START; (7) A=m(a,3,2), P=at(a,3), B=m(a,6,3).
   d. Does the state of the simulation in the previous task necessarily require backtracking?