Note:

- Due: **by 11:59pm on the due date**. Late assignments will not be marked.
- Total mark: **10**.
- Worth: This assignment counts towards 10% of your final mark.
- Submit three files: `euler.pddl`, `sokoban.pddl`, `question3.pdf` that contain your answers for question 1,2,3 respectively.

1. **[2 marks]** A directed graph is a data structure $G = (V, E)$ that consists of a set of *nodes* and *directed edges* connecting the nodes. An *Eulerian circuit* of G is a way to walk through all directed edges where each edge is used *exactly once*, returning to the starting node. For example for the graph defined below:

$$V = \{a, b, c, d, e, f\}$$

$$E = \{(d, e), (d, a)(a, d), (e, f), (a, b), (b, c), (c, a), (f, d)\}$$

An Eulerian circuit is the walk $a \to b \to c \to a \to d \to e \to f \to d \to a$

Write a PDDL domain file `euler.pddl` that specifies the four predicates:

- $node(n)$: indicating that n is a node.
- $edge(n1, n2, e)$: indicating that e is an edge from node $n1$ to $n2$.
- $at(n)$: indicating that the currently we are at node n.
- $used(e)$: indicating that the edge e is used by the walk.

The domain file should specify an action schema: $move(from, to, e)$ that represents one step of the walk from node $from$ to node to along edge e.

Your domain file should allow a problem file to find an Eulerian circuit in a graph. For example, a problem file may be specified as

```
(define (problem euler1)
  (:domain euler)
  (:objects a b c e1 e2 e3)
  (:init (at a) (node a) (node b) (node c)
         (edge a b e1) (edge b c e2) (edge c a e3))
  (:goal (and (used e1) (used e2) (used e3) (at a)))
  )
```

with the plan `(move a b e1) (move b c e2) (move c a e3)`

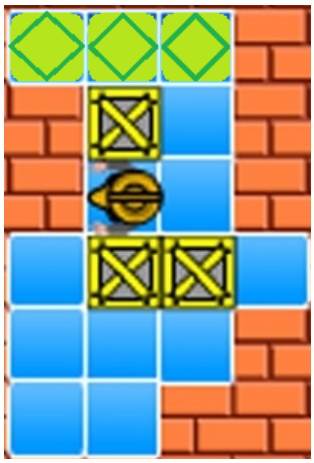
2. **[5 marks]** Sokoban is a puzzle game in which the player has to move around boxes so that they reach a certain place. The rules are that the player can *move horizontally* or *vertically* as long as there are no boxes or bricks, and can *push a box* as long as the player is in front of it and there is space for the box to move. However, the player cannot push two boxes at once. All the boxes have to reach the target (represented as diamonds in the drawing). Solve this problem as a planning problem by writing a PDDL description of the domain `sokoban.pddl`.

Your domain file should represent each location as a pair of coordinates (x, y) where x, y are integers. Then use two relations *inc* and *dec* to encode successor relations on integers, e.g., $inc(1,2), inc(2,3), inc(3,4), \ldots, dec(2,1), dec(3,2), dec(4,3), \ldots$

The domain file should define the following predicates:

- $wall(x, y)$: location (x, y) is a wall
- $box(x, y)$: (x, y) is a box
- $at(x, y)$: the agent is at location (x, y)
- $inc(p, q)$: p, q are integers and $q = p + 1$
- $dec(p, q)$: p, q are integers and $q = p - 1$

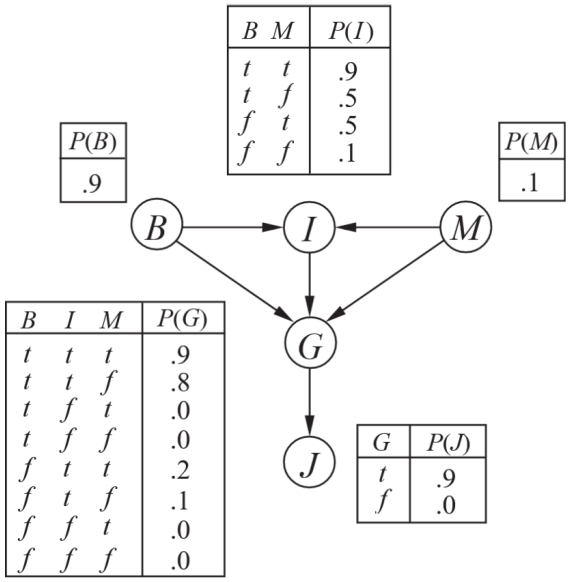
Your domain description should work on a PDDL description of the Sokoban problem instance. A sample problem file (as in the drawing) is below:



```
(define (problem sokoban2)
    (:domain sokoban)
    (:objects v1 v2 v3 v4 v5 v6)
    (:init (inc v1 v2) (inc v2 v3) (inc v3 v4) (inc v4 v5) (inc v5 v6)
           (dec v6 v5) (dec v5 v4) (dec v4 v3) (dec v3 v2) (dec v2 v1)
                                                           (wall v1 v4)
                       (wall v2 v1) (box v2 v2)            (wall v2 v4)
                       (wall v3 v1) (at  v3 v2)            (wall v3 v4)
                                    (box v4 v2) (box  v4 v3)
                                                           (wall v5 v4)
                                                (wall v6 v3) (wall v6 v4))
    (:goal (and (box v1 v1) (box v1 v2) (box v1 v3)))
)
```

3. **[3 marks]** The diagram below shows a simple Bayesian network modeling a court scenario with Boolean variables:

- B = BrokeElectionLaw,
- I = Indicted,
- M =PoliticallyMotivatedProsecutor,
- G = FoundGuilty,
- J = Jailed.



Apply the VE algorithm to calculate the probability that someone goes to jail given that they broke the law, have been indicted, and face a politically motivated prosecutor. Show working.