Tutorial - Week 3

1. 1. Environment is partially observable. The agent knows say things like it's internal state like orientation, speed, fuel level, etc. It also can observe the nearby world within a certain range and field of view. But beyond things like these, everything else is unknown.
   2. It is a combination of collaborative and competitive agents.
      1. Collaborative because the car and other vehicles are working together with the common, collaborative goal of driving safely in the road.
      2. Competitive because the car and other vehicles are optimising for their own goals as well (like reaching a destination quickly for the case of taxi). This competes with the goal of maybe another truck which has to go at a low speed. So now the car has to overtake the truck thereby causing a bit of inconvenience to the taxi because the truck is blocking the taxi's goals in this case.
   3. Stochastic. There is randomness in a lot of things like the inputs from sensor, the actuators executing the actions like steering, unpredictable pedestrian and vehicle movements, etc. All this make it a stochastic environment where next states are not fully predictable.
   4. Its mostly sequential. But there is some episodic component too.
      1. Episodic - While planning to reach a destination, the journey itself is one episode. One journey does not affect other journeys. Also, within a single journey, there can be multiple checkpoints, and there will be little dependency between checkpoints.
      2. Sequential - Within a single episode, the current state depends on past state(s). Need to plan for the end-goal while taking any action.
2. 1. At(Mon, A) ^ At(Ban, B) ^ At(Bx, C) ^ Height(Mon, Low) ^ Height(Bx, Low), Height(Ban, High) ^ Holding(Mon, None) ^ Monkey(Mon) ^ Object(Ban) ^ Object(Bx)
   2. 1. Action(Go(Source, Destination),
         1. PRECOND: At(Mon, Source) ^ Monkey(Mon)
         2. EFFECT: !At(Mon, Source) ^ At(Mon, Destination))
      2. Action(Push(Mon, Obj, Source, Destination),
         1. PRECOND: Monkey(Mon) ^ Object(Obj) ^ At(Obj, Source) ^ At(Mon, Source)
         2. EFFECT: !At(Mon, Source) ^ !At(Obj, Source) ^ At(Mon, Destination) ^ At(Obj, Destination))
      3. Action(ClimbUp(Mon, Obj),
         1. PRECOND: Monkey(Mon) ^ Object(Obj) ^ Height(Mon, Low) ^ Height(Obj, Low)
         2. EFFECT: !Height(Mon, Low) ^ Height(Mon, High))
      4. Action(ClimbDown(Mon, Obj),
         1. PRECOND: Monkey(Mon) ^ Object(Obj) ^ Height(Mon, High) ^ Height(Obj, Low)
         2. EFFECT: !Height(Mon, High) ^ Height(Mon, Low))
      5. Action(Grasp(Mon, Obj),
         1. PRECOND: Monkey(Mon) ^ Object(Obj) ^ Holding(Mon, None)
         2. EFFECT: !Holding(Mon, None) ^ Holding(Mon, Obj)))
      6. Action(Ungrasp(Mon, Obj),

1. PRECOND: Monkey(Mon) ^ Object(Obj) ^ Holding(Mon, Obj)
2. EFFECT: !Holding(Mon, Obj) ^ Holding(Mon, None))

3. No, because for the monkey to push the box back to its original place and still have the banana in its hand, it needs to first get down from the box with banana in its hand. This action is not supported by this action schema. The ClimbDown does not allow for the monkey to be holding any object while climbing down. For this to be possible, the ClimbDown and also the ClimbUp action schemas need to update height of object that the monkey is grabbing at that point as well.

3. 1. Init(On(A, Table), ^ On(C, A) ^ On(B, Table) ^ Block(A) ^ Block(B) ^ Block(C) ^ Clear(B) ^ Clear(C) ^ Clear(Table))
      Goal(On(C, Table) ^ On(B, C) ^ On(A, B))
      Action(Stack(O, Source, Destination),
      PRECOND: On(O, Source) ^ Block(O) ^ Block(Destination) ^ Clear(O) ^ Clear(Destination) ^ (O != Source) ^ (O != Destination) ^ (Source != Destination)
      EFFECT: ! On(O, Source) ^ ! Clear(Destination) ^ On(O, Destination) ^ Clear(Source))
      Action(Unstack(O, Source),
      PRECOND: On(O, Source) ^ Block(O) ^ Block(Source) ^ Clear(O)
      EFFECT: ! On(O, Source) ^ On(O, Table) ^ Clear(Source))
      1. given on pg 365 of Russel. Have one doubt: In the def of Goal state it does not mention On(C, Table). Also, for unstack shouldn't ! On(O, Table) be a precondition.
   2. Unstack(C, A), Stack(B, Table, C) , Stack(A, Table, B)
   3. In the above case even interleaved will work. But for the goal say Goal(On(C, Table) ^ On(A, B) ^ On(B, C)) then interleaved planner won't work.