3 Finite Markov Decision Processes

3.1 Devise three examples of your own that fit the MPD framework, identifying for each its states, actions and rewards. Make the three examples as *different* from each other as possible. The framework is abstract and flexible and can be applied in many different ways. Stretch its limits in some way in at least one of your examples.

text

- A train driving down a train track. This would obviously have to be tested *very extensively* before releasing it into the wild.
  - Environment: The current speed of the train, and a map of the upcoming track (including at least upcoming curves and train stations.
  - Action: gas or break.
  - Reward: 1 if the train is still running, 0 if not. Can be multiplied by the current speed to encourage speedy driving.
- A robot shooting a bow and arrow at a board.
  - Environment: Wind speed and direction. Can be extended with the position if the robot is not in a fixed place.
  - Action: How far to pull back the arrow and where to aim it.
  - Reward: The number of points on the board.
- A machine delivering advertisement content to a website visitor.
  - Environment: The current web page and everything that is known about the customer through e.g. customer data or cookies. Could include age, location, previous website visits, etc.
  - Action: The different pieces of content available to the machine.
  - Reward: 1 if the visitor clicks, 0 if not.

3.2

Is the MDP framework adequate to usefully represent all goal-directed learning tasks? Can you think of any clear exceptions?

- A limitation of the MPD framework is that it needs immediate reward feedback. Tasks where the feedback is delayed would not be adequately represented by this framework. Furthermore, the states need to posess the Markov property, that the state-action pairs only depend on the current state. It is conceivable that they depend on past states as well, though I cannot think of an example where that could not be solved by appending the relevant past states to the current state.

- Chess comes to mind, where the reward is either a loss, win or draw at the very end, and evaluating intermediate states is difficult. Or poker, when in addition to the reward being delayed (though only by a few moves), important information is hidden from the agent (the opponent's cards).

3.3 question

Consider the problem of driving. You could define the actions in terms of the accelerator, steering wheel, and brake, that is, where your body meets the machine. Or you could define them farther out—say, where the rubber meets the road, considering your actions to be tire torques. Or you could define them farther in—say, where your brain meets your body, the actions being muscle twitches to control your limbs. Or you could go to a really high level and say that your actions are your choices of where to drive. What is the right level, the right place to draw the line between agent and environment? On what basis is one location of the line to be preferred over another? Is there any fundamental reason for preferring one location over another, or is it a free choice?

- I would consider the level of deciding where to drive separate from the other three. The other three are about controlling the vehicle once that decision has been made.
  The level where you directly control the tire torques introduces the fewest layers of abstractions between the action and the repsonse. If you define the action as muscle twitches, you go through several systems with possibly unexpected responses before getting feedback in the form of a response. I would think this is more difficult to learn.

- However, all seem possible, and defining the problem where your body meets the machine has the advantage that this is how cars are built, so it will be easier to set up a test environment (you don't need a custom car or a brain implant). Furthermore, the number of of possible actions is severely limited in this case (gas, break, steering wheel, if you have an automatic gear shift). This might make the actions easier to learn.