Leonard Chan
CS 380
2016/04/07

1.
    a. Watson could be applied in any field that requires parsing and analyzing large amounts of data since that's what it does when answering a Jeopardy question. Watson can be used in healthcare for predicting if/when a patient will get sick or finding patterns in market prices.
    It is harder to create an AI system that plays Jeopardy than chess because the range of topics that appears in Jeopardy could be counted as limitless. For the person/team create the AI, they would have to either create a system that can easily efficiently consume and refer back to to a limitless amount of information efficiently. Chess is more constrained in that there are a limited number of rules, agents, and actions that each agent can make. Answering a Jeopardy question requires parsing and semantically understanding the question first before constructing a sensible answer.
    b. The Chinese room argument states that a computer does not have a consciousness no matter how intelligently it may behave. If a person is locked in a room, and knows no Chinese, but is able to translate English to Chinese by means of some mapping or dictionary such that someone who did know Chinese could understand him, it would appear that this person knew Chinese despite that they are just translating. If a computer were in his place, the computer would also not understand Chinese.
    Limiting the scope of the argument to just the Chinese room thought experiment, I am convinced by the argument because neither a person, or computer, placed in this scenario would understand Chinese, just know how to translate to Chinese. I also do not think humans are intelligent in the sense that there is a soul or mind. I instead think that a person is more like a computer in the sense that it displays intelligence by being able to perform complicated tasks.
    c. Small Towers of Hanoi:
        - Performance measures
            i. Number of moves made
            ii. Number of disks in the correct position on the last rod
        - Environment
            i. Number of rods
            ii. Total number of disks
        - Actuators
            i. Move disk from one rod to another
        - Sensors
            i. Ability to detect number of disks on each rod
            ii. Ability to detect location and size of a disk to be able to differentiate them

- State can be represented by tuple of tuples, where each nested tuple indicates the order of the disks on that rod from top to bottom. For the three vertical rods and three disks, the initial state would like like ((D1, D2, D3), (), ()), the goal state would be ((), (), (D1, D2, D3)), and some intermediate state could be ((D3), (D2), (D1)). The state is changed by applying the action of moving one disk to another tuple.

Pac Man:

- Performance measures
    i. Number of moves made
    ii. Number of pellets eaten
- Environment
    i. Rectangular grid
    ii. Scattered pellets
    iii. Walls
- Actuators
    i. Move up, down, left, or right
    ii. Eat pellet
- Sensors
    i. Ability to detect if a pellet was eaten
    ii. Ability to detect locations of walls next to Pac Man
- State can be represented by the location of Pac Man and each of the pellets on the grid as a 2D array, where each element in the array will contain nothing, Pac Man, or a pellet. Walls are not accounted for since walls aren't changing their positions. The initial state could look like [[null, null, pellet, …], [pellet, null, null, Pac Man, ...], ...], and the final state would be a 2D array that contains no pellets, so all the elements will be either null or Pac Man. The state is changed by applying the action of moving to another space, and eating the pellet if Pac Man moves to a space with one.