Kevin Dombrosky 9/8/16

HW1 Section 1

1) Will AI Procrastinate?

I believe that whether or not an AI procrastinates will ultimately determine how we program the AI to behave. Following the article, how we program the AI to behave will ultimately depend on how experts finally decide intelligence, consciousness, and creativity are defined and occur. In the case for creativity, the article states that for some people, procrastination is a prerequisite for creativity. Also, if we wanted the AI to behave like an actual human to perhaps pass the modern day Turing test, the programmers may include the behavior to procrastinate so as to more closely imitate a human's behavior and have a higher chance of fooling the human that it is being tested with. Another point that would support an AI procrastinating would be if the AI is programmed to have complete emotion. If the AI has a sense of enjoyment and conversely resentment, then it may choose to weight tasks differently based on how urgent they are to be completed versus how it can maximize its enjoyment of other activities in the meantime. I believe this is the root of procrastination for most people, the desire to do something other than a more important task.

On the other hand, there are few yet strong cases supporting that an AI would not procrastinate. If it is programmed with the idea to weight creating a solution to a given problem quickly and efficiently very highly, then there would be very little incentive to procrastinate. Another possibility to offer a counterpoint to one of the scenarios supporting procrastination, is if an AI was given the concept of enjoyment, it would hopefully be able to realize that objectively it can have more "fun" doing its normally procrastination activities once its work is done. If the work was complete, it would not need to waste CPU time or memory space worrying about having enough time to complete its obligations.

Ultimately, I think that an AI procrastinating will be dependent on how close to an actual human mind we would want it to function. Essentially, if we want an exact copy down to every facet of emotion, then yes, an AI will likely procrastinate as it wants to be doing things that create joy for it, whatever it perceives that to be. Conversely, I believe that we will ultimately want to create AI that will be specialized toward a certain goal or always strive to a particular goal. For most problems, we will want that goal as quickly and efficiently as possible, which leaves no room for procrastination. Furthermore, you can make the argument that an AI procrastinating for creativity is only doing so because it decided that would be the best way to accomplish its goal, so since it's a means to an end, I don't believe this is necessarily procrastination. In conclusion, I believe that an ideal, computational problem solving AI will never truly procrastinate, as this is generally perceived as wasteful behavior and the AI will strive to eliminate such behavior.

2a) There exist task environments in which no pure reflex agent can behave rationally

True, the task environment could have incomplete information that could disallow rational behavior.

2b) A perfectly rational poker-playing agent never loses.

False, while the agent would likely play based on the most likely outcomes, there still exists probabilities such that an undesirable outcome occurs. If the agent were to be consistently unlucky in such a way, it would ultimately eventually lose.

3)

Bidding on an auction

Performance - Total value of items successfully won / total cost

Environment - Auction hall or ebay like website

Actuators - Make higher bid or abstain from bidding

Sensors - Current bid, retail/market value for product, available money

Proving a mathematical theorem

Performance - Theorem is proven or not.

Environment - Target theorem, known lemmas/facts/laws

Actuators - Applying the mentioned lemmas/facts/laws

Sensors - Theorem proven or not, relevance of current lemmas/facts to current state of proof

Human playing soccer

Performance - Perceived effect on game

Environment - Soccer field

Actuators - Positioning, choice of passing/shooting the ball (in the most simple form)

Sensors - Sight/Sound of the game - positioning of other players in immediate vision, relative location of threats based on listening for other players or listening to communication from other players, friendly or otherwise

Robots playing soccer

Performance – Score

Environment – Robot Soccer field

Actuators – Positioning of other robots, position of ball, proximity to ball, referee state

Sensors – Computer Vision system and FM transmitter/receiver systems

4)

a) What would be the state representation you would use for this problem?

The current subset of numbers and the goal number.

b) What is the initial state for the first example given at Wikipedia?

[75, 50, 2, 3, 8, 7], 812

c) What are your successor functions? Be specific.

Combine two of the numbers in the list using one of the 4 basic operations. Track the operations applied.

d) What is the branching factor of your approach?

68 maximum at the initial node. All subsequent nodes should be less than this.

e) What is the maximum search depth of your approach?

5

f) Imagine that instead you wanted to generate all possible three-digit numbers that can be achieved with a given initial set of six numbers, instead of just reaching a target. Would the answers to the previous parts change? If so, how? If not, why not?

The successor function would change. If an operation is applied to a number that is greater than 3 digits, it will be limited to division or subtraction operations with the larger number. The branching number could then be smaller since there could be less operations to apply to the initial state if the initial state contained any greater than 3 digit numbers. The initial state would no longer track the "goal" state and the state representation would no longer track the goal number.

5)

- Bfs(5) = 8
- Bfs(8) = 31
- Bfs(10) = 11
- Bfs(13) = Not possible with C based math/float representation

6)

a) The branching factor is 4.

b) Worst case number of distinct states would be 4k

c) The maximum number of nodes expanded by breadth-first tree search is n, the number of vertices (if you start at the furthest point of a one-sided tree)

d) The maximum number of nodes expanded by breadth-first graph search is x*y

e) Yes, since it is a grid pattern, the number of moves will always be the difference in coordinates between the goal and the current area, summed together.

f) The number of states expanded by A* would be x+y.

g) Yes, it will now be even further of an underestimate.

h) No, it could now overestimate for some states where there is a shorter path due to the new links.