Universität Zürich[UZH]

Introduction to Artificial Intelligence
Exercise Sheet 1

Laurin van den Bergh, 16-744-401
Yufeng Xiao, 19-763-663
Nora Beringer, 19-734-227

Universität Zürich
Institut für Informatik

Due: March 2, 2022

Exercise 1.1

(a) Weather forecast reports in tabular form can be generated by computers equally well as by humans, but significantly faster. For example see Microsofts automated weather reports. (see clickable link: Microsoft Power Automat Template)

(b) According to https://www.theverge.com/2019/10/30/20939147/deepmind-google-alphastar-starcraft-2-research-grandmaster-level Googles AI can play the game of Starcraft 2 better than most human players.

(c) According to https://mathscholar.org/2019/04/google-ai-system-proves-over-1200-mathematical-theorems/ Google's AI can, unaided by humans, prove many basic mathematical theorems mostly in the area of linear algebra, real analysis and complex analysis. It was able to prove correctly 58% of the training data proves and 38.9% of the test data proves. So it is currently possible but below human level. Also AI has brought human mathematicians close to proving an old long unsolved conjecture.
(see https://innovationorigins.com/en/selected/mathematicians-use-ai-to-prove-new-theorems/)

Exercise 1.2

The rationality of the actions of the AI depend entirely on the objective given to the agent. If the objective is to find the shortest or fastest path, then it would be irrational. If the objective is however to provide a nice view to the customer, then it is rational.

Exercise 1.3

(a) Tetris is
- partially observable, because at each point the agent knows the current pieces on the board and the next piece, but not the following ones.
- a single-agent environment, because only one agent can play the game and it plays against the environment, so there is no competitive or cooperative second agent.
- stochastic, because the agent doesn't know the second to next piece but it is relevant to him and it makes sense to reason about their occurence with probabilities.
- sequential, because for example the current possible actions and states depend on what was done previously.
- dynamic, because if for instance the agent just waits, then the pieces continue dropping down and changing the environment.
- discrete, because the game is divided into a (although fine) grid, thus it has only finitely many states and actions.

(b) A group of robots dancing synchonously is

- partially observable, because they knows all the angles of its joints, its positions and the positions of its legs, but not necessarily the positions of the other robots. I.e. if they don't have a 360 digree view, then they can't observe a robot if it is behind them.
- multi-agent environment, where many robots collaboratively act, i.e. dance. However, it could be considered single-agent if there is one computer/server controlling all of the robots.
- non-deterministic, as some parts of the robots could unexpectedly break, or they could trip, but it doesn't make sense to reason about the probabilities of such events occuring.
- sequential, because every (dance) move depends on the previous one, i.e. it is no independent from it.
- static, because while the robots wait and contemplate their next actions, they might get out of sync, but the environment (i.e. floor, room, etc) does not change.
- continuous, becasue each robot potentially infinitely many moves. E.g. it can change its joint anywhere between 0 and, say, 180 degrees. These are theoretically infinitely many options.

Exercise 1.4

(a) A simple reflexive agent could perform depth first search on a graph like the one depicted here (as long as it is a tree). To clarify our concept we provide pseudo code below:

```
...
if left block is free,
    turn left and move forward
else if block in front is free,
    move forward
else if right block is free,
    turn right and move forward
else
    turn right twice and move forward.
...
```

(b) If the turning action has a chance to fail (e.g. turning fails with some probabilitiy), then the agent can still work as intended, as it will just try taking turning actions until it works. If the sensors fail, then it won't work as intended (unless it gets lucky and the sensors don't matter for the labyrinth). The agent can for example get stuck in some dead end if the sensors tell that left is free, when it actually is not.