Exam: Artificial Intelligence – Algorithms and Application

Module Exam

Winter 2022/2023 Date: 30.03.2023

Important Information

TECHNISCHE UNIVERSITÄT DARMSTADT

WIRTSCHAFTS INFORMATIK

- Please check your exam copy for completeness.
  It covers **20 pages** (cover sheet included).
- Fill out the cover sheet immediately after receiving the exam.
- Use only the examination paper to solve the tasks. If you do not have enough space, you can receive additional paper during the examination. Additional papers must also be marked with your name and matriculation number.
- Please leave a **correction margin of 3 cm**.
- You have a total of **90 minutes** to complete the exam.
- Except for a **non-programmable calculator, no other aids** are allowed in the exam.

We wish you much success!

Please fill out clearly in block letters.

First Name ..... Last Name ..... Seat No. .....

❑ Master

Matr. No. ..... Course of Study ..... ❑ Diplom

Repeater:
❑ yes ❑ no

Section	Max. Points	Achieved Points
1	36	
2	24	
3	30	
Sum	90	

Exam Review („Klausureinsicht“):

(do not fill out before the review)

I have reviewed the corrected exam:

☐ There are no complaints about the correction.

☐ Complaints about the correction exist (see additional sheet).

Date: ……..

Signature: ……....

First Name..... Last Name..... Matr. No.....

1 Basic Concepts and Algorithms (36 Points)

1.1 The researcher **McCarthy** played a **crucial role** for the **field of artificial intelligence**. **Why**? (1 P)

1.2 Please briefly **explain** the **concept** of an **agent** in **artificial intelligence** based on the **definition** of **Russell & Norvig**. Please **draw** the **architecture** of a **"reflex agent"** and briefly **explain** it by **comparing** it to the **general agent model**. (6 P)

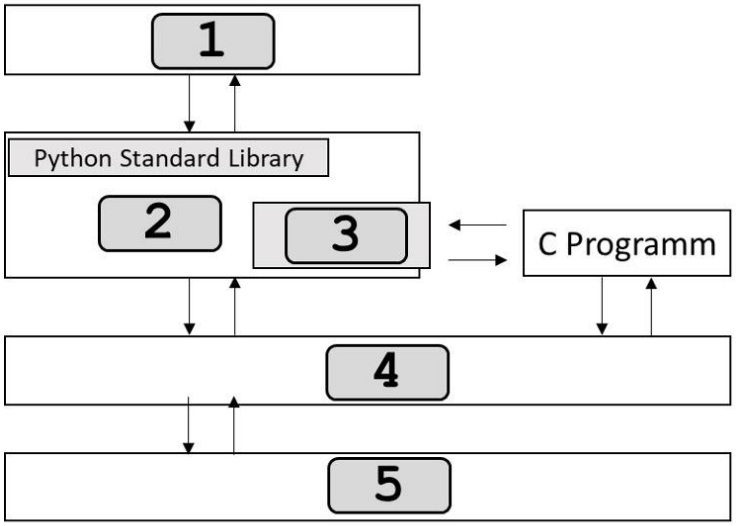
1.3 **Please** briefly **explain** the **difference** between a ***route-finding problem*** and the ***touring*** problem. **Which kind of problem** is the ***traveling salesman problem***? Please **explain** the **kind of problem** with the **national park example** from the lecture. (3 P)

1.4 Please briefly **explain**: What is a "***CAPTCHA***" and **how** is it **related** to **Artificial Intelligence**? (2 P)

First Name..... Last Name..... Matr. No.....

1.5 In the lecture, we discussed the characteristics and benefits of Python.

Please **insert in the table** the **following *Python concepts*** that are **missing in the figure** to map them to the corresponding numbers: *Hardware, Python Program, Python API, Operating System, Python Interpreter*. (2.5 P)



Number	Python Concept
1	
2	
3	
4	
5	

First Name..... Last Name..... Matr. No.....

1.6 Please briefly **explain** the **difference** between ***batch*** and ***online*** **learning** in machine learning. (2 P)

1.7 Please **name three other *scientific domains of AI*** than machine learning and knowledge reasoning that were introduced in the lecture. (1.5 P)

First Name..... Last Name..... Matr. No.....

1.8 One of your friends is building a machine learning model using a **logistic regression**. He wants to predict if a car has a specific problem (classification). The data has five different problem types. However, as soon as he runs his prediction, he only gets error messages. Please briefly **explain**: What is the **most probable reason** for those **errors**? (2 P)

1.9 Please **define**: **What is** a **model** in machine learning? (2 P)

1.10 In the context of Python programming, **what** is ***"beautiful soup"*** **used for**? (1 P)

1.11 The current AI Capstone was conducted in the field of Porsche's complaint management. Please briefly **describe *two potentials*** of Porsche's complaint management as a **sensor** of **product** and **service quality** that were presented in the guest lecture by Porsche. (2 P)
(**Note**: It is sufficient to only roughly name the two potentials.)

1.12 Please briefly **describe two** possible ***data-related limitations*** when aiming to create a **deployable AI solution** based on your experience that you gained with the Porsche complaint data during the AI Capstone project. (2 P)

First Name..... Last Name..... Matr. No.....

1.13 Please briefly **define it** and **explain its structure**: What is a ***data frame*** in Python? (3 P)

1.14 Please briefly **explain** the **difference** between a ***bar chart*** and a ***histogram***. For **what** are they usually **used**? (3 P)

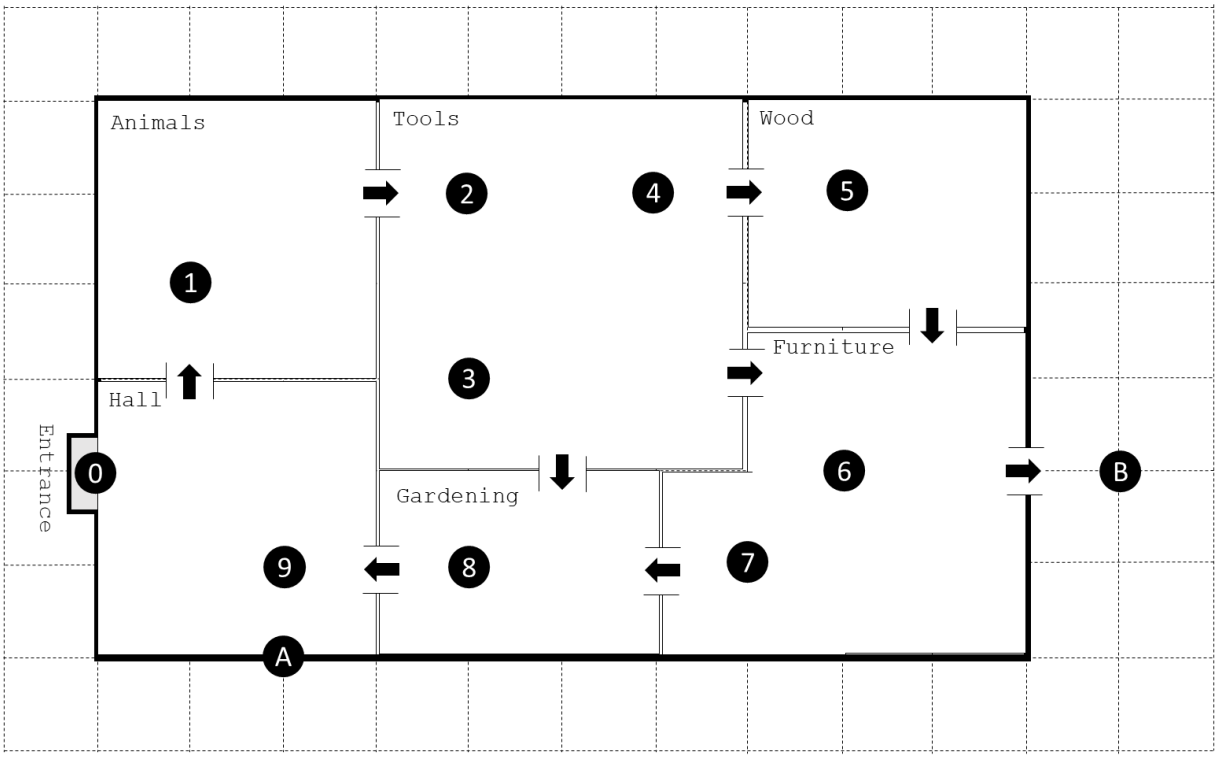
1.15 Please **define** the two steps ***mutation*** and ***crossover*** in the context of **genetic algorithms**. **What** is the **difference** between the **two steps**? (3 P)

First Name..... Last Name..... Matr. No.....

2 Application of Search Algorithms (24 Points)

Consider the following **AI problem**:

Your local DIY-store plans to improve the shopping experience with automated agents. The agents have a language interface and a route-finding module. Customers can ask the agent where to find specific tools and materials and the agent guides them through the store. Your job is to help develop the agent. For that purpose, you received the following floor plan:



On the floor plan, you can see that each section (e.g., "Tools") has subsections that are indicated by a number or letter, and that certain sections are connected with one another via doors. To simplify the development, you can assume that the agent travels in Manhattan distance, and that it travels along the walls. Of course, the agent can only enter a room through a door. As the store management wants to avoid collisions, the agent is only allowed to move towards increasing numbers and always only drives towards one subsection in each room it passes (e.g., exclusively one subsection from number 2, 3 to 4 in the tools section). On the map there are two exits, an official one near the checkout (A) and one in the outdoor area of the store (B).

First Name..... Last Name..... Matr. No.....

2.1 Please **classify** the **agent's task environment** with the **PEAS** framework. (4 P)

2.2 To implement a first proof-of-concept agent, you **model** the **store** with **simple search trees**. For that purpose, you perform the following two tasks (a and b):

a) Please **draw** the **subset** of a possible **search tree** of the **DYI store** to the official exit in state with a ***depth-first search***. **How many nodes** do you have to **visit** until you **reach** the **exit** in this case? (**Note**: You can decide whether your algorithm always chooses the right or the left node when searching the tree. Please indicate your assumption.) (4 P)

First Name..... Last Name..... Matr. No.....

b) Please **draw** again the **subset** of a possible **search tree** of the **DYI store** until it reaches the official exit but this time you use the ***greedy best first strategy***. Please use the **Euclidian distance** to the **official exit** from each state's nearest section door as a heuristic. Please **explain** your **results**. (8 P)

First Name..... Last Name..... Matr. No.....

2.3 As a next step, you are asked to implement an emergency program. You decide to implement an ***A* search algorithm*** to find the **fastest path** to one of the **exists** (A or B). By doing so, please assume that **you** are in **state 2** in the **tools section** and the fire alarm starts. **Which path** is the **best** for ***each*** **exit**? Use the same **heuristic** as in task 2.2 b) for the A*-algorithm. (8 P)

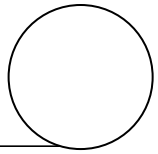
First Name..... Last Name..... Matr. No.....

3 Agent Programming with Python (30 Points)

The manager of a local construction market hires you to design the agent program for an agent in Python. For this purpose, you have received the following code from your friend Sandra, which she has copied from her lecture “AI Algorithms and Applications with Python”.

Python Code

```
vacuum_world = {"1":[["2"], False],
                "2":[["1", "3", "4"], False],
                "3":[["2"], True],
                "4":[["2"], False]}

class Cleaner:
    def __init__(self, room, world):
        self.location = room
        self.world = world

    def percepts(self):
        is_dirty = self.world[self.location][1]
        self.act(is_dirty)

    def drive(self):
        neigbor_rooms = self.world[self.location][0]
        num_rooms = len(neigbor_rooms)
        r = numpy.random.randint(low = 0, high = num_rooms)
        self.location = neigbor_rooms[r]

    def suck(self):
        self.world[self.location][1] = False

    def act(self, ______ ):
        if(is_dirty == True):
            self.suck()
        else:
            self.drive()
```



First Name..... Last Name..... Matr. No.....

3.1 Based on Sandra's code on the prior page, please **classify** the ***type of agent*** Sandra has implemented and **briefly explain** your **decision.** (2 P)

3.2 Sandra uses a specific Python data structure to **model** the **vacuum_world**. **Which kind of *data structure*** does she use to do so? Please also **draw** a **map** of the **vacuum_world** based on the information from the code and **mark** the **agent's starting position** in the map. (4P)

First Name..... Last Name..... Matr. No.....

3.3 The `act()` function is missing a parameter: `def act(self, _____ )`

Please **correct** the **function** and **fill** in the **missing code**. (2 P)

First Name..... Last Name..... Matr. No.....

3.4 The current agent version is missing an energy management. Please **extend** the **agent class** with a **"power consumption" functionality** in Python considering the following rules (11 P):

- The agent consumes two energy levels every time it sucks up some dirt and one energy level when it moves.
- If the energy level drops below 5, the agent should drive back to the location "1".
- The starting energy is 18 energy units.

Python Code

```
class Cleaner:



    def __init__(self, room, world):
        self.location = room
        self.world = world




    def percepts(self):
        is_dirty = self.world[self.location][1]
        self.act(is_dirty)




    def drive(self):
        neigbor_rooms = self.world[self.location][0]
        num_rooms = len(neigbor_rooms)
        r = numpy.random.randint(low = 0, high = num_rooms)
        self.location = neigbor_rooms[r]
```

First Name..... Last Name..... Matr. No.....

***Note**: This code block continues on the next page.*

Python Code

```
    def suck(self):
        self.world[self.location][1] = False

    def act(self, ______ ):
        if(is_dirty == True):
            self.suck()
        else:
            self.drive()
```

First Name..... Last Name..... Matr. No.....

3.5 Please **write** some **Python** to **start** a **cleaning simulation** with the **agent.** (6 P)

3.6 What will be a **possible problem** of this **type of agent** if the **number of rooms** in the construction market ***increases***? **How** can you **solve** it? (2 P)

3.7 What is the **problem** of **this Implementation** if the ***number of rooms*** **changes**? **How** would you **solve** this **problem**? (3 P)