**Full Name (as shown in attendance sheet):**

**[Optional] LumiNUS Account:**

**[Optional] NRIC / Passport / NUS Matriculation No.:**

(Select and provide the applicable)

**Institute of Systems Science**

**National University of Singapore**

**GRADUATE CERTIFICATE**

**INTELLIGENT REASONING SYSTEMS**

**Assessment**

**Subject: *Reasoning Systems***

SECTION A

|  |  |
| --- | --- |
| **Question** | **Marks** |
| **1** | **/1** |
| **TOTAL** | **/1** |

**Instructions for Paper**

Duration: Fifteen minutes exam

This is an *OPEN BOOK* examination. This examination paper consists of *one* Section and *one* Question. You are to answer *ALL* questions. There are a total of *1 Mark* for this paper.

1. Read **ALL** instructions before answering any of the examination questions.
2. Write your Student ID number on the **front page** of this examination paper in the box provided.
3. This is an **Open Book** examination. If you wish, you may use reference materials to answer a question. Reference materials can be *books, manuals, handouts* or *notes*.
4. Answers are to be written **only** in this **examination paper** and any **attachments** provided and will be considered for credit. Answers written in any appendices will **NOT** be marked.
5. Use a pen for writing your answers. Pencil may only be used for drawing diagrams and writing program code.
6. Non-programmable calculators may be used if required. **However, computers of any form (laptops, tablets, smart watches etc.) are not permitted to be brought into the examination hall.**
7. State clearly any assumptions you make in answering any question where you feel the requirement is not sufficiently clear.
8. At the end of the examination:
9. Hand-in the examination paper for **each** section **separately**, any appendices and attachments.
10. You are **not** allowed to remove the examination paper, appendices or attachments from the examination hall.

***REMEMBER:***

***This is an OPEN BOOK exam.***

***There are a total of 1 Mark for this paper.***

***You are required to answer ALL questions.***

***State clearly any assumptions you make in answering any question where you feel the requirement is not sufficiently clear.***

**SECTION A**

**Question 1** *(Total: 1 Mark)*

***SpamAssassin*** is a widely used open source spam filter. It calculates a score for an incoming e-mail, based on ***nine*** built-in *tests* (in SpamAssassin’s terminology). Scores for individual tests can be negative (indicating evidence suggesting the e-mail is *ham* rather than *spam*) as well as positive, in the range [-1, 1]. An e-mail is considered suspicious if the overall score (sum of the individual scores) is 5 or more, and in this case a ***junk*** flag and a summary report is added to the e-mail’s header.

Though such a tool helps filter out many spam e-mails successfully, it also overlooks some spam and sometimes categorizes important e-mails to be spam by false alarms. To improve the accuracy of spam detection, ***Smart Service*** has collected a sample dataset ***D*** of 5,000 past reports of spam and non-spam e-mails. The company has engaged you to develop a better decision model using IS techniques to classify spam e-mails, based on the ***nine*** individual test scores, to replace the simple sum of scores used in the current tool. Suppose a subset of data *D*\*, consisting of a randomly selected 80% of the samples from ***D***, has been established.

You are modeling a ***state space search*** task for an optimal set of weights together with a threshold *t* =0.5 that produces an error rate ***less than 5%*** in spam classification on the dataset*D*\*. The range of each weight is [0, 1] and the sum of all weights is normalized to be 1 when applied for classification.

References:

***State Space Search*** is a process used in the field of computer science, including artificial intelligence (AI), in which successive configurations or states of an instance are considered, with the intention of finding a goal state with a desired property. Depth-first search and Breadth-first search are forms of state space search.

Problems are often modelled as a state space, a set of ***states*** that a problem can be in. The set of states forms a graph where two states are connected if there is an ***operation*** that can be performed to transform the first state into the second.

<https://en.wikipedia.org/wiki/State_space_search>

**Answer the following questions:**

* 1. Propose a representation of **state** to carry out the **start space search**, and suggest an initial state.

*(0.5 Mark)*

[Answer]

* 1. Define a possible **search operator/action** assuming the **step** of weights change is 0.1 (e.g. a new weight will be obtained by adding 0.1 or subtracting 0.1 from the current one) before normalization.

*(0.5 Mark)*

[Answer]

**END OF ASSESSMENT PAPER**