

GALWAY-MAYO INSTITUTE OF TECHNOLOGY

SEMESTER 2 EXAMINATIONS 2018/2019

MODULE: COMP08016 - ARTIFICIAL INTELLIGENCE

PROGRAMME(S):

GA_KSOAG_H08 BACHELOR OF SCIENCE (HONOURS) IN COMPUTING IN
SOFTWARE DEVELOPMENT

YEAR OF STUDY: 4

EXAMINER(S):

Dr. JOHN HEALY	(Internal)
Mr. Tom Davis	(External)
Dr. Des Chambers	(External)

TIME ALLOWED: 2 Hours

INSTRUCTIONS: Answer 4 questions. All questions carry equal marks.

PLEASE DO NOT TURN OVER THIS PAGE UNTIL YOU ARE INSTRUCTED TO DO SO.

The use of programmable or text storing calculators is expressly forbidden.

Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

Requirements for this paper:

1. Non-Programmable Calculators Allowed

1. (a) Describe, using examples where appropriate, how an Artificial Neural Network (ANN) can be trained to learn classification tasks that are **linearly-separable**. Include a fully labelled diagram in your answer, showing the structure of a **perceptron** and explain the key steps involved in the training process.

(15 Marks)

- (b) Discuss the structure and function of a **multilayer back-propagation neural network**. Your answer should include a diagram that illustrates the direction of information flow through the network, how errors are back-propagated and address the techniques for choosing a network **topology**.

(10 Marks)

2. **Figure 1** below depicts a semantic network of eight nodes interconnected by edges. The starting node is node 'A' and 'H' is the goal node. Each node is labelled with a letter in the upper compartment and a heuristic estimate of distance to the goal node in the lower compartment. The actual distance between two nodes is shown as a number along their connecting edge.

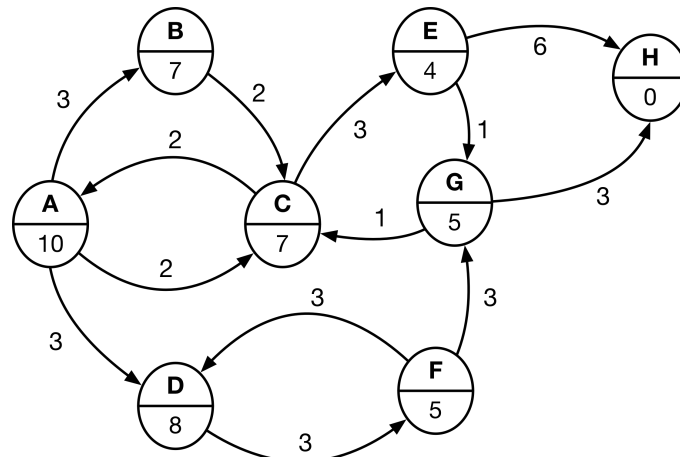


Fig. 1

- (a) Show how the **A* algorithm** can find the optimal path from the initial node (A) to the goal node (H). Your answer should clearly show the state of the **OPEN** and **CLOSED** queues for each iteration of the algorithm and how the path evaluation function, $f(n)$, is computed.

(11 Marks)

- (b) Discuss the efficiency of the A* algorithm and the parts of the algorithm that contribute most to the **computational complexity** of the search of a semantic network. Your answer should also address how different types of graph topologies may impact the performance of A*.

(6 Marks)

- (c) Using a diagram and code snippets where appropriate, discuss how **iterative deepening** can be applied to A* to reduce computational complexity without compromising algorithmic optimality and completeness.

(8 Marks)

3. **Figure 2** below depicts a 4-*ply* game tree having leaf nodes decorated with a score that represents the computation of a static evaluation function:

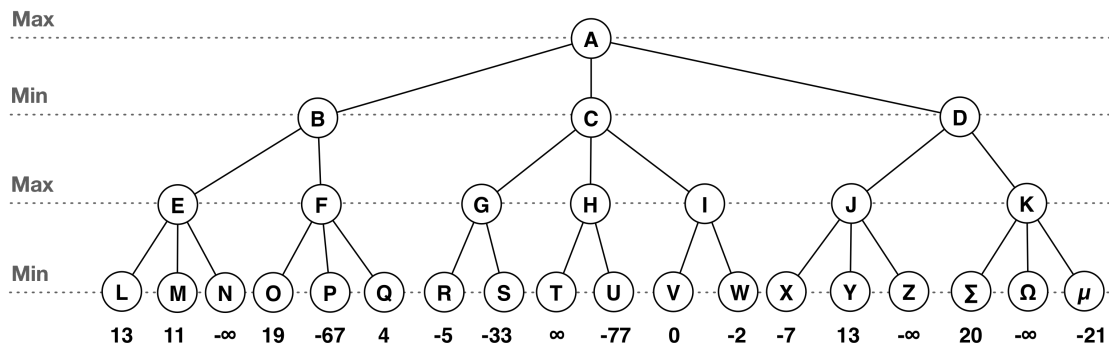


Fig. 2

- (a) Show, using labelled diagrams, how the **minimax** algorithm can determine the best move to make from node 'A'. Your answer should clearly illustrate how MAX and MIN values are computed at each level.

(10 Marks)

- (b) Describe how **alpha-beta pruning** can be applied to the game tree in **Figure 2** to reduce the number of nodes to be generated and examined. Your answer should show the pruned game tree, indicate the alpha and beta cut-off points and address the computational effectiveness of alpha-beta pruning.

(15 Marks)

4. (a) Explain, using examples, the following terms as they apply to **heuristic search** algorithms:

- Admissibility **(3 Marks)**
- Monotonicity **(3 Marks)**
- Foothills and Plateaux **(3 Marks)**

- (b) Discuss how **steepest-ascent** can overcome the limitations of the basic hill-climbing algorithm and contrast hill-climbing and **best-first** approaches. Use diagrams and pseudocode or Java snippets to illustrate your answer.

(10 Marks)

- (c) Explain how **simulated annealing** can be used to overcome the limitations of hill-climbing algorithms when searching for an optimal solution to a search problem.

(6 Marks)

5. (a) Explain the **fuzzy set operations** that underpin fuzzy logic and explain how they relate to Boolean logic. Include diagrams with your answer where appropriate.

(8 Marks)

- (b) An autonomous car has a braking system implemented using fuzzy logic, that applies pressure to a brake in response to the distance of an object ahead of it and the amount of rain on a road surface. **Figures 3, 4 and 5** below depict fuzzy sets that describe the linguistic variables *distance*, *road* and *brake* respectively that are used by the braking system. The universe of discourse ranges from 0 – 100 metres for the variable *distance* and from 0 – 5mm for the variable *road*. The linguistic variable *brake* is defined using singleton spikes and has a universe of discourse spanning the range 0-100%.

The following three rules describe the reasoning used by a fuzzy inference system:

- IF *distance* IS *very very short* OR *road* IS *extremely dry* THEN *brake* IS *more or less hard*
- IF *distance* IS *not slightly long* AND *road* IS *somewhat moist* THEN *brake* IS *not soft*
- IF *distance* IS *medium* AND *road* IS *not very wet* THEN *brake* IS *very normal*

Compute, using the **Sugeno** inference method, the predicted braking pressure with input parameters of *distance* = 30 metres and *road* = 3mm of rain. Your answer should clearly show each step in the fuzzy inference process.

(17 Marks)

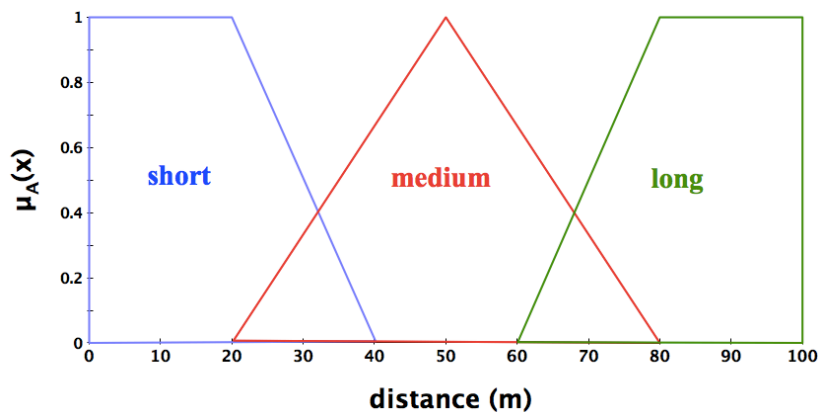


Fig. 3: Fuzzy sets for the variable Distance

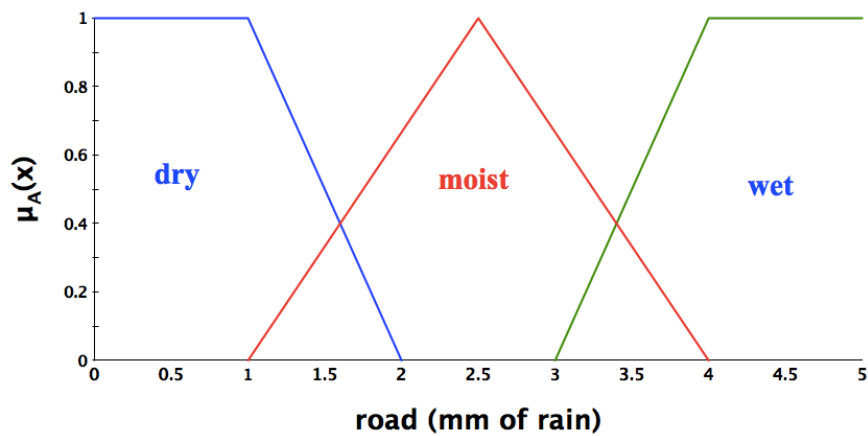


Fig. 4: Fuzzy sets for the variable *Road*

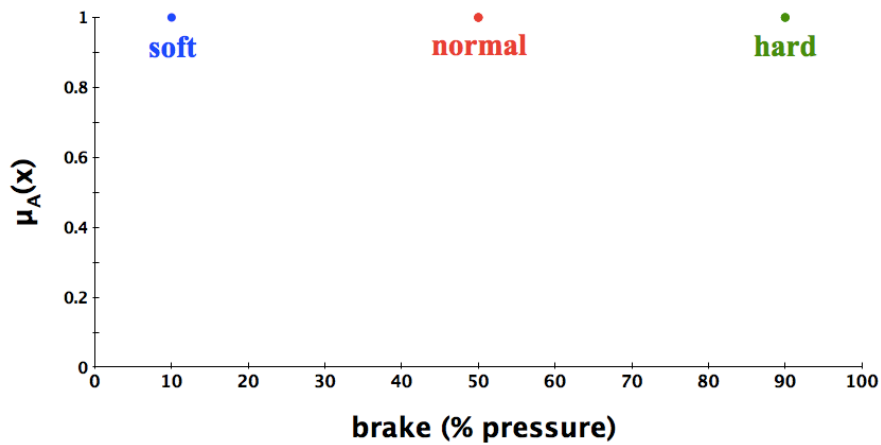


Fig. 5: Singletons for the variable *Brake*

Hedge	Formula
<i>Somewhat</i>	$\sqrt[3]{\mu_a(x)}$
<i>More or Less</i>	$\sqrt{\mu_a(x)^2}$
<i>Indeed</i>	$2(\mu_a(x))^2$ if $0 \leq \mu_a(x) \leq 0.5$ $1 - 2(1 - \mu_a(x))^2$ if $0.5 < \mu_a(x) \leq 1$
<i>A Little</i>	$\mu_a(x)^{1.3}$
<i>Slightly</i>	$\mu_a(x)^{1.7}$
<i>Very</i>	$\mu_a(x)^2$
<i>Extremely</i>	$\mu_a(x)^3$
<i>Very Very</i>	$\mu_a(x)^4$

Table 1: Fuzzy Hedges