CS3061 – Artificial Intelligence
2017 Exam Solutions

Question 1

1. (a) What is the *Halting Problem* and what does it have to do with AI?

[8 marks]

(b) What is a *Universal Turing machine* and what does it have to do with AI and the Halting Problem?

[12 marks]

(c) What is Marr's tri-level hypothesis and what does it have to do with AI?

[10 marks]

(d) What is the *Boolean Satisfiability Problem* (SAT), and what does it have to do with the question *P=NP*? What does this question have to do with AI?

[10 marks]

(e) What is a *Constraint Satisfaction Problem* (CSP)? Is the Halting Problem a CSP? Is SAT a CSP? Explain, formulating either or both as a CSP in case either or both are.

[10 marks]

a) The **halting problem** is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running (i.e halt) or to continue to run forever.

The **halting problem**, devised by Alan Turing, holds that no algorithm is able to correctly predict whether another algorithm will run forever or it will eventually halt. Consider a super intelligent AI body with a program that contains every other program in existence. Researchers have provided a logical proof that if such an AI could be contained, then the halting problem would by definition be solved. To contain that AI, the argument is that you'd have to simulate it first, but it already simulates everything in existence so we arrive at a paradox.

Similarly to this, with regards to the moral ethics of an AI body. It is impossible to design an algorithm that will determine whether or not this AI body will act morally, thus causing considerate concern amongst the field.

b) The **Universal Turing Machine** is a Turing machine that can simulate an arbitrary Turing machine on an arbitrary input. The universal machine essentially achieves this by reading both the description of the machine to be simulated as well as the input thereof its own tape.

With encoding action tables as strings it becomes possible in principle for Turing machines to answer questions about the behaviour of other Turing machines. Most of these questions, however are undecidable, meaning that the function in question cannot be calculated mathematically.

Rice's theorem states that an intelligence (human) cannot be able to determine whether another intelligence (AI) possesses a certain property, such as being friendly. While we can design an AI to be friendly, it does not mean that we can check whether an arbitrary AI is friendly. So, while we can create a friendly AI, we also need to be able to ensure that IT cannot create another unfriendly AI.

c) Marr treated vision as an information processing system. He put forth the idea that one must understand information processing systems at three distinct, complementary levels of analysis. This idea is known in cognitive science as Marr's Tri-Level Hypothesis:

- **Computational Level:** What does the system do, what problems does it solve and why does it do these things.
- **Algorithmic Level:** How does the system do what it does, specifically, what representations does it use and what processes does it employ to build and manipulate the representations.
- **Implementation/Physical Level:** How is the system physically realised (e.g computer, neural structures, mechanical implementation etc…)

d) The **Boolean Satisfiability Problem** is sometimes abbreviated as **SATISIFIABILITY** or **SAT**. It is the problem of determining if there exists an interpretation that satisfies a given Boolean formula. If this is true the formula is deemed *satisfiable* and otherwise *unsatisfiable*. Take for example the Boolean formula:

$$"a\ AND\ NOT\ b"$$

This can be satisfied using the values:

- a = TRUE
- b = FALSE

However, if we consider the Boolean formula

$$a\ AND\ NOT\ a"$$

There is no possible value of a which can satisfy the above formula.

The **Boolean Satisfiability Problem** has to do with the question P=NP as it was the first problem that was proven to be NP complete (Cook-Levin Theorem). This means that any problem in NP can be reduced in polynomial time by a deterministic Turing machine to the problem of determining whether a Boolean formula is satisfiable. An important consequence of this theorem is that if there exists a deterministic polynomial time algorithm for solving Boolean satisfiability, then every NP problem can be solved by a deterministic polynomial time algorithm.

The SAT problem is, given a formula, to check whether it is satisfiable. This **decision problem** is of central importance in the area of artificial intelligence.

e) A **Constraint Satisfaction Problem (CSP)** is a mathematical question defined as a set of objects who must satisfy a number of constraints or limitations. CSP's represent the entities in a problem as a collection of finite constraints over variables, which is solved by constraint satisfaction methods. CSP's are very popular in the field of artificial intelligence since the regularity in their formulation provides a common basis to analyse and solve problems of many seemingly unrelated families. CSP's often exhibit high complexity, requiring a combination of heuristics and combinational search methods to be solved in a reasonable time.

The **Boolean Satisfiability Problem (SAT)** can be roughly thought of as a CSP. Consider the example problem:

$$x_1 \vee x_3$$

We can describe this as a CSP tuple $\langle X, D, C \rangle$ as follows:

- $X = \{x_1, x_3\}$ *The set of variables*
- $D = \{\langle T, F \rangle, \langle F, T \rangle, \langle T, T \rangle\}$ *The domain of values the variables can take*
- $C = \{(x_1 \vee x_3)\}$ *The set of constraints on the variables*

With regards to the **halting problem**, this cannot be represented as a CSP as by the very bounds of its problem it cannot be solved nor represented by a Turing machine.