Faculty of Engineering, Mathematics and Science
School of Computer Science & Statistics

B.A. (Mod.) Computer Science & Language
Year 4 Annual Examinations

Knowledge Representation and Automata

Mon, 30 April 2018
SPORTS CENTRE
9:30 – 11:30

Dr Tim Fernando

Instructions to Candidates:
Attempt two questions. All questions carry equal marks. Each question is scored out of a total of 50 marks.

You may not start this examination until you are instructed to do so by the Invigilator.

Exam paper is not to be removed from the venue.

Materials permitted for this examination:
Non-programmable calculators are permitted for this examination — please indicate the make and model of your calculator on each answer book used.
1. (a) In what sense is computation reducible to searching a graph? Let us accept the Church-Turing Thesis, and assume searching a graph means looking for goal nodes connected (in the graph) to a given start node. [15 marks]

(b) What is the Halting Problem? What complications does it pose for a depth-first search of the computation graph of a non-deterministic Turing machine? Are these complications insurmountable? [15 marks]

(c) How much more concise can non-deterministic finite-state machines be as representations of regular languages than deterministic finite-state machines? [10 marks]

(d) Are sentences of Monadic Second-Order logic more concise than non-deterministic finite-state machines? Justify your answer. [10 marks]
2. (a) Given a string $s$ and a language $L$, what is the $s$-derivative of $L$ and how is it related to $L$-inseparability?

[10 marks]

(b) What are the derivatives of $a^*b^*$ and $\sum_{n \geq 0} a^n b^n$? What does the number of derivatives of a language $L$ say about whether or not $L$ is accepted by some finite-state machine?

[15 marks]

(c) What is reification and why is it said to lead to a promiscuous ontology?

[5 marks]

(d) Structure matching and tableaux represent two approaches to reasoning about subsumption $C \sqsubseteq C'$ in Description Logic. Outline the main differences between these approaches, explaining why negation is problematic for one but not for the other.

[10 marks]

(e) Given a set $A$, what is the $A$-reduct $\rho_A(s)$ of a string $s$ and how can we use $\rho_A$ to express the set of strings satisfying the MSO$_A$-sentence $\varphi_A$ below

\[ \forall x \left( \bigvee_{a \in A} (P_a(x)) \land \bigwedge_{a' \in A - \{a\}} \neg P_{a'}(x) \right) \]

saying there is a unique $a \in A$ in every string position $x$? How is $\varphi_A$ used to capture the accepting runs of an NFA?

[10 marks]
3. (a) When is the implication

\[ p : \neg q \]

true at an interpretation, and what are the logical consequences of a set of clauses?

[10 marks]

(b) Given a mechanical procedure $\vdash$ for deriving a clause $g$ from a knowledge base $KB$, written $KB \vdash g$, what does it mean for $\vdash$ to be sound? And what does it mean for $\vdash$ to be complete? Give an example of a mechanical procedure that is sound but not complete, and another example that is complete but not sound.

[10 marks]

(c) Let $(†)$ be the assertion that an atom $g$ is a logical consequence of $KB$ if and only if:

(i) $g$ is an instance of a fact in $KB$, or

(ii) there is an instance of a rule $g : \neg b_1, \ldots, b_k$ in $KB$ such that each $b_i$ is a logical consequence of $KB$.

Does $(†)$ provide a sound and complete proof system for definite clauses? Justify your answer.

[10 marks]

(d) Given a set $K$ of constants, let us say an interpretation $I = (D, \phi, \pi)$ is $K$-syntactic if its domain $D$ is the set $K$ of constants,

\[ D = K \]

and each constant $c$ in $K$ refers to itself

\[ \phi(c) = c. \]

How are $K$-syntactic interpretations sufficient to establish the soundness and completeness of a mechanical procedure $\vdash$ for Datalog clauses, but insufficient to relate $\vdash$ to reasoning about the agent’s environment?

[10 marks]
(e) Given that a problem in NP is computable, why is there no contradiction between Trakhtenbrot’s theorem, which asserts the uncomputability of checking if a first-order sentence has a finite-model, and Fagin’s theorem, \( \text{NP} = \Sigma_1^1 \), which equates NP with a fragment \( \Sigma_1^1 \) of second-order logic that properly includes first-order logic over finite structures?

[10 marks]