CS206 End-Semester Examination

Max marks: 70 **Time: 3 hours**

- *You are required to answer each question only in the space provided with each question.*
- *Only material written within the allotted answering space for each question will be graded.*
- *Please attach all your rough sheets.*
- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- ***If you need to make any assumptions, state them clearly.***
- ***Do not copy solutions from others. Penalty for offenders: FR grade.***

1. *[10 marks]* Let $\psi = \forall x_0 (\forall x_1 \exists x_2 \forall x_3 \exists x_4 \, (\phi_1(x_1, x_3, x_4) \vee \phi_2(x_0, x_2, x_4)) \wedge \exists x_1 \forall x_2 \forall x_3 \exists x_4 \, (\phi_3(x_2, x_3, x_4) \vee \phi_4(x_0, x_1)))$, where ϕ_1, ϕ_2, ϕ_3 and ϕ_4 are *quantifier-free* predicate logic subformulae. You are also told that the ϕ_i's are syntactically constructed by applying propositional connectives on k ternary predicates $P_1, P_2, \ldots P_k$ and m binary predicates $Q_1, Q_2, \ldots Q_m$. Moreover, none of the ϕ_i's have any function symbols.

   A smart logician now claims that she has an algorithm that can take arbitrary three-argument formulae ϕ_1, ϕ_2, ϕ_3 and an arbitrary two-argument formula ϕ_4 subject to the restrictions in the previous paragraph, and can determine the validity of ψ given above. If you think the logician is right, briefly describe how you would go about algorithmically checking the validity of ψ (you could give a pseudo-code, for example).

If you think the logician is incorrect, describe your reasons for the same.

2. At an intersection in a busy city, there are two traffic lights controlling the flow of traffic in directions perpendicular to each other. Each light is controlled by a controller that communicates with the other controller by reading and/or writing a shared variable called `flag` stored with a central arbiter. The behaviour of each controller can be described by the Kripke structure shown in Fig. 1, where the labels on the states denote the state of the corresponding traffic light or the read/written values of `flag`.

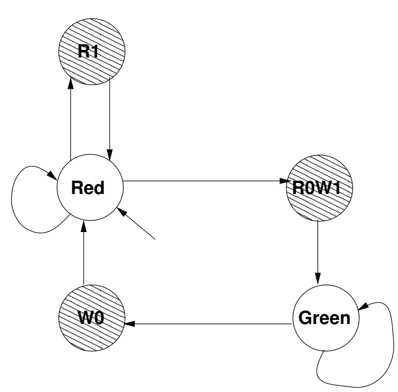

Figure 1:

The controllers run *synchronously*, i.e., both perform state transitions simultaneously. Thus, every state transition of one is necessarily accompanied by a state transition (possibly back to the same state) of the other. The controllers may, however, try to access the `flag` simultaneously during their operation. To resolve such access conflicts, an arbitration mechanism is needed. Thus, to access the `flag` (for reading and/or writing), each controller must present a request to a central arbiter, and wait until exclusive access to `flag` is granted by the arbiter. Therafter, it reads and/or writes the `flag` variable, and informs the arbiter of the completion of its operation and relinquishes its exclusive access to `flag`. The arbiter may then grant access to `flag` to the other controller, if there was a pending request. We will assume that the arbiter grants requests in a first-come-first-served order, except when requests come in simultaneously from the two controllers. In such cases, the arbiter decides which controller to grant acess in a random manner, giving rise to non-deterministic behaviour.

In Fig. 1, whenever the controller enters the `Red` or `Green` state, it issues a request to the arbiter for accessing `flag`. The shaded states in the figure represent states where the controller has exclusive access to `flag`. Label `R0W1` on the shaded state in Fig. 1 denotes that a value of 0 has been read and a value of 1 has been written to `flag` by the current controller in this state. Similarly, labels `R1` and `W0` denote reading a value of 1 and writing a value of 0 to `flag`, respectively. Note that the successor of every shaded state is a non-shaded state. Hence, after a controller enters a state where it has exclusive access to `flag`, it relinquishes its exclusive access in the very next state transition. Note also that since the arbiter never allows both controllers to simultaneously access `flag`, the two controllers cannot be in their shaded states simultaneously.

(a) *[10 marks]* Draw a Kripke structure for the overall behaviour resulting from the synchronous running of the two controllers with access to `flag` controlled by an arbiter as described above. In your Kripke structure, assign a unique numeric identifier, e.g. $1, 2, 3, \ldots$ to each state, and label each state with atomic propositions indexed by 1 or 2 to represent information about the colour of the corresponding light and the value of `flag`.

(b) For each of the properties described below, either give a CTL formula expressing the property, or describe why you think such a property cannot be expressed in CTL.

i. *[2.5 marks]* It is possible that one of the traffic lights never turns green even though the other traffic light turns red infinitely often.

ii. *[2.5 marks]* Whenever the first traffic light turns red, it either eventually turns green or the second traffic light eventually turns green.

3. Let f be a unary function and $\mathcal{M}$ be a model containing an interpretation of f. In general, some elements x in the universe (or domain) of $\mathcal{M}$ satisfy $f(x) = x$, while other elements satisfy $\neg(f(x) = x)$.

   (a) *[10 marks]* We wish to write a predicate logic sentence ϕ that evaluates to True in $\mathcal{M}$ if and only if the universe of $\mathcal{M}$ has more elements x satisfying $\neg(f(x) = x)$ than elements y satisfying $f(y) = y$. Is it possible to have such a predicate logic sentence ϕ? If your answer is in the affirmative, you must provide ϕ. If your answer is in the negative, you must prove that such a sentence ϕ cannot exist.

   (b) *[10 marks]* Suppose we now wish to obtain a predicate logic sentence ψ that evaluates to True in $\mathcal{M}$ if and only if there are infinitely many elements x in the universe of $\mathcal{M}$ that satisfy $f(x) = x$. Is it possible to have such a predicate logic sentence ψ? If your answer is in the affirmative, you must provide ψ. If your answer is in the negative, you must prove that such a sentence ψ cannot exist.

4. Consider the predicate logic sentence $\phi = \phi_1 \wedge \phi_2 \wedge \phi_3$, where

$$\begin{aligned}
\phi_1 &= \forall x\,(f(g(x)) = g(x)),\\
\phi_2 &= \forall x\,(g(f(x)) = f(x)),\\
\phi_3 &= \forall x \exists y\,((x = g(y)) \vee (x = f(y)))
\end{aligned}$$

   (a) *[7.5 marks]* Let $\mathcal{H}$ be the Herbrand Universe for the sentence ϕ. Let $M_{\mathcal{H}}$ be a model with the universe (or domain) being $\mathcal{H}$, and with the obvious interpretations of functions f and g, i.e., if t is a term in $\mathcal{H}$, the terms $f(t)$ and $g(t)$ in $\mathcal{H}$, are defined to be the result of applying f and g respectively to t. If $M_{\mathcal{H}} \models \phi$, how many distinct elements can be present in the universe of $M_{\mathcal{H}}$.

(b) *[7.5 marks]* Show using natural deduction for predicate logic that $\phi_1, \phi_2, \phi_3 \vdash (x = f(x))$.

(c) *[10 marks]* Let $\phi_4 = \exists x \neg(f(x) = g(x))$. Show using Herbrand's Theorem that $\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$ is unsatisfiable. Your solution for this part must not use results obtained in previous parts.