CS206 End-Semester Examination

Max marks: 60 **Time: 3 hours**

- *Be brief, complete and stick to what has been asked.*
- *If needed, you may cite results/proofs covered in class without reproducing them.*
- ***If you need to make any assumptions, state them clearly.***
- <u>***Do not copy solutions from others. Penalty for offenders: FR grade.***</u>

1. *[5 + 5 + 5 marks]* Let $\phi_1 = \forall x\, \exists y\, \exists z\, \forall w\ (P(y,z) \wedge P(z,w) \wedge P(w,x))$, and $\phi_2 = \exists x\, \forall y\, \forall z\, \exists w\ (P(y,z) \wedge P(z,w) \wedge P(w,x))$.

   (a) Show using natural deduction that $\phi_1 \vdash \phi_2$. You must use only the basic elimination and introduction rules in your proof.

   (b) We wish to determine if a model for $\phi_2 \wedge \neg\phi_1$ exists. If you think such a model exists, you must provide a model $\mathcal{M}$, i.e. a domain S of elements and a definition of the binary predicate P. If you think such a model does not exist, you must give a proof of unsatisfiability using Herbrand's Theorem.

   (c) An adventurous logician claims to have found a predicate logic sentence ϕ_3 in Skolem Normal Form that is semantically equivalent to ϕ_2, uses no Skolem functions or Skolem constants, and uses no predicate symbol other than P. If you think the logician is correct, you must provide ϕ_3 and give a natural deduction proof of $\phi_2 \vdash \phi_3$. If you think the logician is incrrect, you must give a justification why such a ϕ_3 cannot exist.

2. *[5 + 10 marks]* In this question, we will restrict ourselves to predicate logic formulae over the Boolean domain, i.e., $S = \{\mathsf{True}, \mathsf{False}\}$. Let $\phi = (\forall x\, \exists y\, \exists z\ \psi_1(x,y,z)) \rightarrow (\exists x\, \exists y\, \exists z\ (\psi_1(x,y,z) \wedge \psi_2(x,y,z)))$, where ψ_1 and ψ_2 are propositional logic formulae in CNF.

   A student has devised the following algorithm, called `CheckValidPhi`, that takes as inputs a pair of three-argument propositional logic formulae, ψ_1 and ψ_2, in CNF and determines whether ϕ, as defined above, is valid over the Boolean domain. Note that the student makes use of the DPLL procedure for checking propositional satisfiability. It is assumed that each call to DPLL returns whether the propositional logic formula fed as input is satisfiable (the satisfying assignment, if any, is ignored).

```
CheckValidPhi(psi1, psi2)

  if (DPLL(F1) = ‘‘Satisfiable’’) then
       return(‘‘Phi is Valid’’);
  else if (DPLL(F2) = ‘‘Satisfiable’’) then
       return(‘‘Phi is Not Valid’’);
  else
       return(‘‘Phi is Valid’’);
```

   (a) Express `F1` and `F2` in the pseudocode above in terms of ψ_1 and ψ_2. Your answer must ensure all of the following:

   - `F1` and `F2` must be obtained using ψ_1 and/or ψ_2 as subformulae, and by applying propositional logic connectives ($\neg, \vee, \wedge$) on the subformulae and variables.

- No more than two occurrences of each of ψ_1 and ψ_2 must be used as subformulae in either `F1` or `F2`.
- Neither ψ_1 nor ψ_2 must be used with one or more of their arguments substituted with propositional constants, $\{\mathsf{True}, \mathsf{False}\}$.
- Algorithm `CheckValidPhi` must always returns the correct answer.
- It must be possible for algorithm `CheckValidPhi` to take each of the three branches of the `if then else` structure for suitable input formulae ψ_1 and ψ_2.

You must give informal justification why your choice of `F1` and `F2` ensures that `CheckValidPhi` always gives the correct answer. Answers without justification will fetch no marks.

(b) Give ψ_1 and ψ_2 such that with your choice of `F1` and `F2` above, each of the three branches of the `if then else` structure are taken. In other words, you must give three pairs (ψ_1, ψ_2) of propositional logic formulae, one for each branch, and indicate which branch of the `if then else` structure is taken for the pair of formulae under consideration.

3. *[7.5 + 7.5 marks]* Consider the Kripke structure M given in Fig. 1, where each state is labeled with the set of atomic propositions that are true in that state. The entire set of atomic propositions is $\{a, b, c\}$.

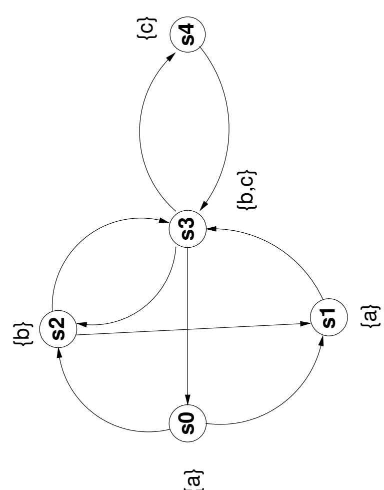

Figure 1:

(a) Use the explicit-state CTL model checking algorithm to determine if $M, s_0 \models \mathbf{AF}\,\mathbf{EG}\,c$ in the above Kripke structure. You must clearly indicate the set of states labeled at each step of the algorithm, along with an informal justification of why those states are labeled that way.

(b) Find a path π in the above Kripke structure that ***violates*** the LTL formula $(\mathbf{FG}\,\neg a) \vee (\mathbf{GF}\,(b \wedge \mathbf{X}\,c))$. You may describe the path π using the labels of the states, e.g. "s_2, s_3, s_4, s_3 repeated indefinitely" describes the infinite path $s_2, s_3, s_4, s_3, s_2, s_3, s_4, s_3, s_2, s_3, s_4, s_3, \ldots$. You must also provide justification of why your path violates the above LTL formula.

Answers without justification will fetch no marks.

4. *[2.5+2.5+5+5 marks]* Let $\{a, b, c\}$ be a set of atomic propositions used to construct formulae in CTL and LTL.

(a) Express the following property in CTL:
Whenever a becomes true, either b never becomes true subsequently in the future or c becomes true infinitely many times subsequently in the future.

(b) Express the following property in LTL. Whenever c becomes true, every subsequent occurrence of a in the future is followed by an occurrence of b in the next state.

(c) Consider the CTL formula $\phi_1 = (\mathbf{AF}\, b) \vee (\mathbf{AF}\, c)$, and the LTL formula $\phi_2 = (\mathbf{F}\, b) \vee (\mathbf{F}\, c)$.

i. We wish to know if it is possible to express the property described by ϕ_1 in LTL. In other words, does there exist an LTL formula, ϕ_3 such that $M, s_0 \models \phi_1$ iff $M, s_0 \models \phi_3$ for all Kripke structures M and states s_0?

ii. We also wish to know if it is possible to express the property described by ϕ_2 in CTL. In other words, does there exist a CTL formula ϕ_4 such that $M, s_0 \models \phi_2$ iff $M, s_0 \models \phi_4$ for all Kripke structures M and states s_0?

In both cases, if your answer is in the affirmative, you must give the corresponding formula and a brief justification of why you think the formulae describe the same property of any Kripke structure. If your answer is in the negative, you must give a brief justification of why you think the property cannot be described in one of the variants of temporal logic.