CS206 Spring 2005 Quiz 3

Time: 60 mins | Total marks: 30

- ***The exam is open book and notes.***
- ***Results/proofs covered in class/problem sessions/assignments may simply be cited, unless specifically asked for.***
- ***If you need to make any assumptions, state them clearly.***
- ***Do not copy solutions from others or indulge in unfair means.***

1. *[5 + 10 marks]* In this question, we will try to prove the validity of a predicate logic sentence by invoking Herbrand's Theorem. Consider the formula $\phi = ((\phi_1 \wedge \phi_2 \wedge \phi_3) \rightarrow \phi_4)$, where:
   $\phi_1 = \exists x\, P(x, g(x))$
   $\phi_2 = \forall y\, P(y, f(y))$
   $\phi_3 = \forall u\, \forall v\, \forall w\, (P(u, v) \wedge P(v, w) \rightarrow P(u, w))$
   $\phi_4 = \exists z\, P(z, f(g(z)))$

   (a) Give a predicate logic formula ψ in Skolem Normal Form, such that ϕ is valid iff ψ is unsatisfiable. In order to score marks, ψ must be in Skolem Normal Form.

   (b) Use Herbrand's Theorem to show that ψ is indeed unsatisfiable. In your solution, you must clearly show the following:
   - Elements of the Herbrand Universe (ground terms) that are used in your proof.
   - The finite set of ground clauses whose conjunction is unsatisfiable.
   - A justification for the propositional unsatisfiability of the above set of ground clauses.

2. *[5 + 10 marks]* Two processes P_1 and P_2, as described below, are run concurrently on a uniprocessor computer system.

Process P1:	Process P2:
`repeat forever`	`repeat forever`
`i11:  x := (1 + y) mod 3;`	`i21:  y := (1 + x) mod 3;`
`i12:  y := (x + 1) mod 2;`	`i22:  x := (y + 1) mod 2;`

   In the above description, variables `x` and `y` are of type `natural number`. The symbol `:=` refers to an assignment operation and `mod` refers to the usual remaindering operation when dividing by integers. Label i_{kl} refers to the l^{th} instruction of the k^{th} process. Each such instruction is assumed to be *atomic*, i.e., once it starts executing on the processor, it cannot be terminated until it has completed execution, and the variable to which an assignment is made in the statement has actually assumed its new value.

   Variables `x` and `y` are shared by both the processes, i.e., both processes read from and write to the same memory locations when referring to the same variable. However, since there is a single processor, only one of the instructions can be executing at any given time. The execution of the instructions can, however, be interleaved in all possible ways, as long as an i_{k1} instruction is executed between every two successive executions of i_{k2} and vice versa. Thus, the following is an allowed execution sequence of instructions: $i_{11}, i_{12}, i_{11}, i_{21}, i12, i_{11}, i_{22}, \ldots$.

   (a) We wish to describe the evolution of the above system using a Kripke structure, where each state of the Kripke structure is identified by the values of the variables x and y. Draw a Kripke structure describing the behaviour of the above system, assuming that both x and y are initialized (prior to the start of execution of P_1 and P_2) to 1. You must label each state of your Kripke structure with a tuple (x, y) giving the values of x and y in that state.

   (b) Using atomic propositions $q_1 = (x > 2)$ and $q_2 = (y < 2)$, express the following property as an LTL formula and as a CTL formula:
   Whenever x becomes ≤ 2, the system evolves to ensure that y grows to at least as large as 2 at some time in the future (excluding the current time instant), and then x grows to a value larger than 2 at some time instant after that.