MSc in Mathematics and Foundations of Computer Science

COMPUTER-AIDED FORMAL VERIFICATION

Michaelmas Term 2024

Submission deadline: 12 noon, Wednesday 8th January 2025, via Inspera.

There is a total of 100 marks available for this paper, you should attempt all parts of the paper.

NB: You must not discuss this examination paper with anyone.

TURN OVER

Question 1

Temporal Logic

(a) Consider each pair of temporal logic formulae below. Identify which logic each formula is written in and then show whether the pair are are equivalent or not. If needed, you can use known equivalences from propositional logic or for dualities between temporal operators.

(i) $(\Box a \vee \Box b) \rightarrow \Diamond(a \wedge b)$ and $\Box(\neg a \vee \neg b) \rightarrow (\Diamond \neg a \wedge \Diamond \neg b)$

(ii) $\forall \Box(p \rightarrow \forall \Diamond \forall \bigcirc q)$ and $\forall \Box(p \rightarrow \Diamond \bigcirc q)$

(8 marks)

(b) Assuming that a, b and c are atomic propositions, translate each of the following statements into the temporal logic LTL.

(i) At most two of a, b and c are ever true simultaneously.

(ii) If a and b are ever true simultaneously then, from that point on, at least one of them is always true at any point.

(iii) c is true at exactly two distinct time steps and these are not consecutive.

(iv) Each of b and c are true infinitely often but the number of times that they are true simultaneously is finite.

(7 marks)

(c) For each of the properties expressed in part (b), state whether or not it belongs to these linear-time property classes: invariant, safety, liveness. Justify your answers. (7 marks)

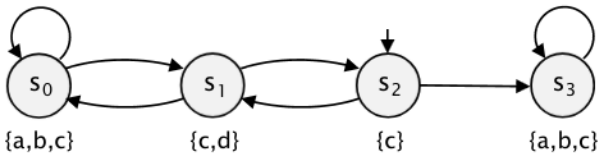
Question 2

Model Checking Algorithms

(a) Illustrate the application of the CTL model algorithm checking algorithm to determine whether the LTS $\mathcal{M}$ below satisfies the formula:

$$\phi = \forall\Diamond\neg\exists[\neg d \,\mathsf{U}\, \forall\bigcirc a]$$

If it is not satisfied, you should explain why not.



(8 marks)

(b) Illustrate the application of the LTL model algorithm checking algorithm to determine whether the same LTS $\mathcal{M}$ above satisfies the formula:

$$\psi = \Box(a \rightarrow \bigcirc(b \rightarrow \bigcirc(c \wedge \Diamond a)))$$

again explaining why not if it is not satisfied. (12 marks)

(c) Consider the temporal operator $\Diamond^{[l,u]}$ where $l \in \mathbb{N}$, $u \in \mathbb{N} \cup \{\infty\}$ and $l \leq u$, which is a step-bounded variant of the eventually operator $\Diamond$ imposing lower and upper bounds on the occurrence of an event. For example, to add this to CTL, for an arbitrary CTL formula ϕ and path π through an LTS:

$$\pi \models \Diamond^{[l,u]}\phi \;\text{ iff }\; \exists k \in \mathbb{N} \text{ such that } k \geq l, k \leq u \text{ and } \pi[k] \models \phi$$

The operator can be added to LTL in a similar fashion.

For each of the two logics, CTL and LTL: (i) explain whether adding $\Diamond^{[l,u]}$ increases expressivity; (ii) describe an appropriate way to extend the existing model checking algorithm to incorporate this operator and analyse the efficiency of the resulting algorithms. (15 marks)

TURN OVER

Question 3

Model Checking Implementations

Consider the following small program which manipulates a variable x, whose initial value is unknown but in the range 0 to $2n-1$, for a constant $n \geq 1$. Two distinct approaches to analysing the behaviour of this program would be: (i) symbolic model checking; (ii) bounded model checking via SAT or SMT.

```
while (true) {
    x := x mod n;
    x := x + 3 mod 2n;
}
```

(a) Fixing $n = 2$, show in detail how the *transition relation* representing this program is represented using each approach as: (i) a binary decision diagram (BDD); and (ii) a symbolic expression to be passed to a SAT or SMT solver, respectively. (12 marks)

(b) For each of the two approaches to model checking discussed above, give an example of a formally specified property of the program that could be checked using that method and which would be less suited to analysis with the other approach. Explain your reasoning in each case. (6 marks)

Question 4

Runtime Verification

One of the key strengths of model checking is that it performs an exhaustive search of the model of a system's executions, yielding firm guarantees on whether temporal logic specifications of correctness are satisfied or not. However, this often comes at the expense of limited scalability. An alternative approach, called *runtime* verification instead only checks the correctness of individual system executions.

For this question, you are asked to study the paper "Runtime Verification for LTL and TLTL" by Bauer, Leucker and Schallhart, which introduces this approach to verification. For convenience, a copy of this paper is provided under "Reading Material" on the course webpage.

Explain, using your own examples where relevant, the key ideas behind this verification technique. Then discuss how runtime verification relates to the methods covered on this course: the similarities between them, the ways in which they differ, and the reasons behind this. You may wish to cover the types of correctness property supported, but you do not need to cover technical details from Section 4 of the paper (on timed extensions).

Your answer to this question should not exceed three pages. Answers will be judged according to the correctness and clarity of their exposition, and the level of insight shown in comparing runtime verification to the methods you have studied.

(25 marks)

LAST PAGE