EXAMINATION

Modeling and Analysis of Concurrent Systems 2

course code: 192135320
date: Januray 31, 2014
time: 13.45–17.15

General

- This is an 'open book' exam. Printed handouts (both slides and papers) may be used, but no hand-written notes, previous examinations, or their answers.
- This exam consists of 3 pages with 4 questions. In total 100 points can be earned: 70 points with the material of the first four lectures (Question 1-3) and 30 points with the material from the research papers (Question 4).

Question 1 (35 points)

Consider the following formulas from CTL*:

formula (I) $\mathbf{EG}\,(p \rightarrow \mathbf{AG}\,p)$

formula (II) $\mathbf{EG}\,(p \rightarrow \mathbf{G}\,p)$

A. *(5 pts)* Indicate whether **(I)** and **(II)** are in LTL and/or CTL.

B. *(5 pts)* Draw a Kripke structure of maximally 3 states with no deadlocks, in which precisely one of **(I)** and **(II)** holds. Explain your answer.

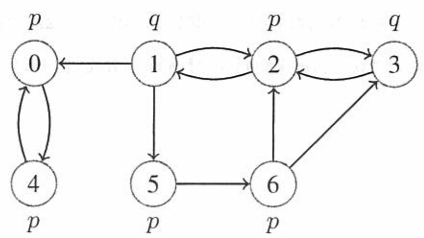
C. *(5 pts)* Explain how an LTL model checker could be used to check formula **(II)** and to obtain a witness.

D. *(5 pts)* Rewrite formula (I) into an equivalent formula **(III)** in the fragment $\{\mathbf{EG}, \mathbf{EU}, \neg, \vee\}$.

E. *(10 pts)* Apply symbolic model checking to verify **(III)** on the Kripke structure M below. Indicate intermediate results of the fixed point computations, and indicate the set of states where **(III)** hold. You don't have to represent these sets by BDDs, but may use normal set notation.

F. *(5 pts)* Characterize the paths that are fair under the fairness constraint $\{2\}$.

Kripke structure M:



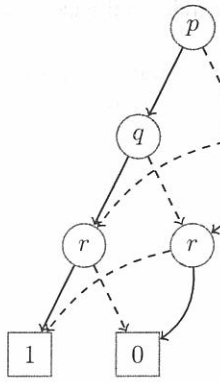
Question 2 (20 points)

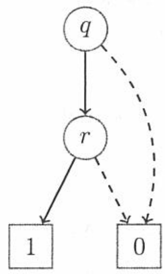
A. *(7 pts)* Represent the following formula as an OBDD, ordering the variables as $p < q < r$:

$$(p \vee (p \wedge q)) \wedge (p \wedge (r \vee q))$$

B. *(6 pts)* BDD libraries keep their BDDs *ordered* and *reduced*. Mention two advantages of this form.

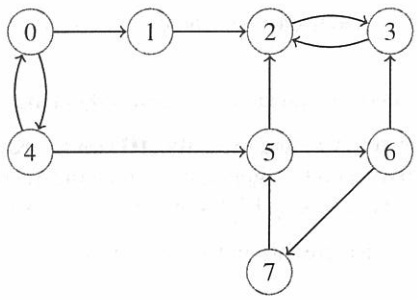
C. *(7 pts)* Demonstrate the APPLY function when computing the conjunction ($\wedge$) of the following BDDs; It is sufficient to draw the call graph and the final result.





Question 3 (15 points)

We want to compute the strongly connected components of the following graph in parallel:



A. *(10 pts)* The FB algorithm (forward-backward) depends on the choice of a pivot node. Choose a pivot so that FB immediately splits the graph in four non-empty subgraphs, and list those subgraphs.

B. *(5 pts)* What is the worst-case time complexity of **FB**? Give a short explanation.

Question 4 (30 points)

Please provide short and to-the-point answers to the following questions:

A. *(4 pts)* The shared hashtable in [Laarman et al., 2010] uses open addressing rather than chaining. Why is that beneficial given the memory hierarchy in modern hardware?

B. *(4 pts)* The approach of [Dillinger & Maniolis, 2009] ends up with using Bloom filters to compactly store a set of state vectors. How can this influence the results of the model checker?

C. *(5 pts)* [Clarke et al., 2001] use a SAT solver for bounded model checking. Mention one advantage and one disadvantage of putting a *bound* on model checking.

D. *(5 pts)* [Dill and Ip, 1996] exploit symmetry arising from permuting elements in a *scalar set*. Which operation on scalar sets is allowed? Why are all other operations forbidden?

E. *(4 pts)* If an abstraction in [Clarke etal., 2000] leads to a spurious counterexample, it is refined and model checking starts all over. Why does this CEGAR process of abstraction-refinement terminate?

F. *(4 pts)* The ASym rule in [Pasareanu etal., 2008] derives $\langle True \rangle M_1 || M_2 \langle P \rangle$ from $\langle True \rangle M_1 \langle A \rangle$ and $\langle A \rangle M_2 \langle P \rangle$. What is the role of A and why doesn't the user have to specify A?

G. *(4 pts)* Verisoft [Godefroid, 2003] runs on actual software, rather than models. Mention two techniques to avoid excessive memory usage.