Synthesis of Reactive Systems – Final Exam

ECI 2010

Submission: August 30th, 2010

Questions 1-3 weigh $33\frac{1}{3}\%$ each. Question 4 weighs 20% (bonus). The maximal grade is 100%. Feel free to contact me by email (firstname.lastname@doc.ic.ac.uk) if you have questions. If you are at the UBA you could also contact Nicolas. Please email me your answers by the date above. Good Luck!

1. (a) Translate the following English specifications to LTL. The letters p, q, and r are atomic formulas (i.e., describe sets of states).
   i. Every two p-states are separated by two r-states that occur strictly after the first p-state and strictly before the second p-state.
   ii. Every p-state is either in the initial position, comes right after the end of a q sequence that started from an r, or remains true until the occurrence of an r.

   (b) Construct a temporal tester for one of the properties that you wrote.

   (c) The following property is not expressible in LTL.

   The proposition p holds in every even position.

   Notice that this is very different from $p \wedge \Box(p \leftrightarrow \bigcirc \neg p)$. Indeed, the second one requires that p is false in all odd positions while the first one does not care about odd positions. This property is generally known as "LTL cannot count". Notice, that in an odd position the property checks all future odd positions and in an even position the property checks all future even positions.

   Show how to construct a temporal tester for this property. Unlike all temporal testers we have encountered so far, this temporal tester will have to use additional variables that serve as memory. It should, as usual, have one Boolean variable that signals the current truth value of the formula.

2. (a) Devise an algorithm to check *satisfiability* of LTL formulas. That is, given an LTL formula, check if there is some computation of some system that satisfies this LTL formula.

   (b) Show that validity of LTL formulas (do all computations of all systems satisfy it) and model checking of an LTL formula over an FDS can be reduced to satisfiability.

That is, suppose that you have a program that gets an LTL formula and answers "yes" if it is satisfiable and "no" if it is not. Use this program to construct a program that will answer "yes" or "no" to the questions of validity and model checking.

3. (a) Consider a game $G = \langle \mathcal{V}, \mathcal{X}, \mathcal{Y}, \Theta_e, \Theta_s, \rho_e, \rho_s, \Box p \rangle$. Prove that the algorithm in Figure 1 computes the set of states in the game from which the system can enforce $\Box p$. Recall that the proof has to include two parts:

   i. Soundness - from every state in the computed set *great* there is a strategy that enforces $\Box p$.

   ii. Completeness - every state from which $\Box p$ can be enforced is maintained in the fixoint throught the computation.

   You may assume that both $\Theta_s = \mathtt{T}$ and $\rho_s = \mathtt{T}$.

```
1. fix (great := p) {
2.   great := great ∧ ⟲ p;
3. end // fix
4. if (∀X . Θ_e → ∃Y . Θ_s ∧ great)
5.   Win!
6. else
7.   Lose!
```

Figure 1: Algorithm for controlling $\Box p$.

   (b) Consider the algorithm for solving GR(1) games (see slide 141 in the notes). Extract from it a *simpler* algorithm for solving games with a winning condition of the form $\Diamond \Box p$.

4. Consider a game $G = \langle \mathcal{V}, \mathcal{X}, \mathcal{Y}, \Theta_e, \Theta_s, \rho_e, \rho_s, \Box p \rangle$, where p is a temporal formula that uses past operators. Suppose that $\mathcal{T} = \langle \hat{\mathcal{V}}, \hat{\Theta}, \hat{\rho}, \emptyset, \emptyset \rangle$ is a temporal tester for p that is complete and deterministic with respect to $\mathcal{V}$. Let $x_p \in \hat{\mathcal{V}}$ be the variable of $\mathcal{T}$ such that in every computation $\sigma : s_0, s_1, \ldots$ of $\mathcal{T}$ we have $x_p = \mathtt{T}$ in a state s_i of σ iff $(\sigma, i) \models p$.

   I mentioned that the synchronous parallel composition of G and $\mathcal{T}$ is a safe way to reason about controlling $\Box p$. Recall that $G \,|||\, \mathcal{T} = \langle \mathcal{V} \cup \hat{\mathcal{V}}, \mathcal{X}, \mathcal{Y} \cup \hat{\mathcal{V}}, \Theta_e, \Theta_s \wedge \hat{\Theta}, \rho_e, \rho_s \wedge \hat{\rho}, \Box x_p \rangle$.

   Find a game G and a formula $\Box p$ such that adding the variables of $\mathcal{T}$ to the control of the environment is incomplete. That is, show that in the game

$$G \widetilde{|||}\, \mathcal{T} = \langle \mathcal{V} \cup \hat{\mathcal{V}}, \mathcal{X} \cup \hat{\mathcal{V}}, \mathcal{Y}, \Theta_e \wedge \hat{\Theta}, \Theta_s, \rho_e \wedge \hat{\rho}, \rho_s, \Box x_p \rangle$$

   The environment can win although the system can control $\Box p$.