Formal Methods in Software Development

Written Test – Rome, February 12, 2020

Name
Surname

Q1 (Modelling): Why does the parallel composition of n processes give rise to a Labeled Transition System (or a Kripke structure) with a number of states exponential in n?

Q2 (Temporal Logic): Let f and g be two atomic propositions. Exhibit a Kripke structures $\mathcal{M}$ such that $\mathcal{M} \models \mathbf{EG}(f \,\mathbf{U}\, g)$ and $\mathcal{M} \not\models \mathbf{AG}(f \,\mathbf{U}\, g)$.

Q3 (Liveness): Let AP be a set of atomic propositions and let $P_1, P_2 \subseteq (2^{AP})^\omega$ be two liveness properties. Is $P_1 \cup P_2$ a safety property? Motivate your answer shortly.

Q4 (Fixed Points): Let S be a *finite* set and let $a \in S$. Which are the least and the greatest fixed points of the operator $T : 2^S \mapsto 2^S$ defined by: $T(X) = X \cup \{a\}$?

Q5 (Equivalences): Among the following temporal logics, which ones are always invariant with respect to stuttering equivalence?
☐ $\text{LTL}_{-\mathbf{X}}$
☐ $\text{CTL}_{-\mathbf{X}}$
☐ CTL
☐ LTL

Q6 (Algorithms): Briefly describe the (idea of the) procedure *checkEU*(f,g) that labels all states satisfying $\mathbf{E}[f\ \mathbf{U}\ g]$, assuming that states satisfying f and g are correctly labeled.

Q7 (Automata): Let us consider the alphabet $\Sigma = \{a, b\}$. Give a Büchi automaton recognising the language $\mathcal{L} \subseteq \Sigma^\omega$ of words ending with an infinite sequence of a or an infinite sequence of b. Formally $\mathcal{L} = \Sigma^* \cdot a^\omega + \Sigma^* \cdot b^\omega$.

Q8 (μ-calcolus): Write a μ-calculus expression that is the translation of the CTL formula $\mathbf{EG}\,\mathbf{E}(f\,\mathbf{U}\,g)$.
