CTL* Exercises

1. Consider the following formulae, where p, q are atomic propositions:

   $$(\mathbf{A}) \quad \mathsf{A}\ \mathsf{F}\ \mathsf{G}\ (\neg p \vee q)$$
   $$(\mathbf{B}) \quad q \wedge \mathsf{A}\ \mathsf{F}\ q \wedge \neg(\mathsf{E}\ [(\neg q)\ \mathsf{R}\ (\neg p)])$$

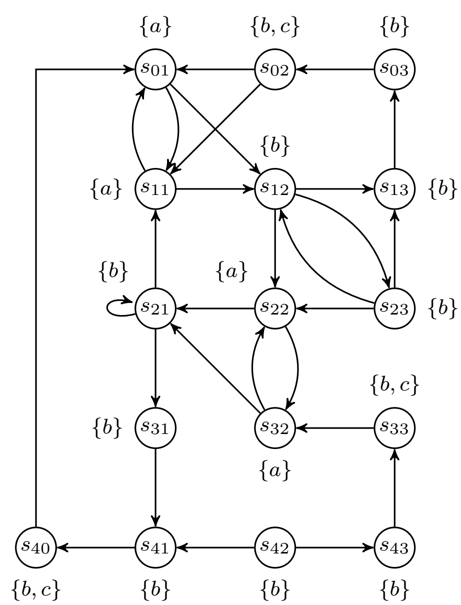
   Answer the following questions for **both** formulae **(A)** and **(B)** and provide a brief motivation for every answer that you give.

   (a) Is the formula in LTL? Is it in CTL? Is it in ACTL*?

   (b) Draw a Kripke Structure with a single initial state in which it holds.

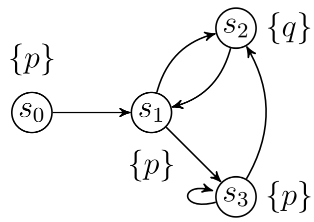
   (c) Draw a Kripke Structure with a single initial state in which it does not hold, but in which it does hold fairly with an appropriate fairness constraint. Also provide this fairness constraint.

2. Consider the following Kripke structure over the set of atomic propositions $\{a, b, c\}$.



   Compute for which states the properties $\mathsf{A}\ [(b \wedge c)\ \mathsf{R}\ (b \vee a)]$ and $\mathsf{E}\ [(\mathsf{E}\ \mathsf{G}\ a)\ \mathsf{U}\ (\mathsf{E}\ [b\ \mathsf{U}\ c])$ hold, and compute the largest autobisimulation (i.e., bisimulation on the Kripke Structure and itself).

3. Consider the following Kripke Structure:



Consider the following formulae, where p and q are atomic propositions:

(**C**) $\mathsf{A}\,\mathsf{F}\,q$
(**D**) $\mathsf{A}\,[q\;\mathsf{R}\;p]$
(**E**) $\mathsf{E}\,\mathsf{F}\,(\mathsf{A}\,[q\;\mathsf{R}\;p])$

(a) Determine the set of states where (**C**) holds using the labelling algorithm for CTL model checking algorithm, based on graph algorithms (chapter 4, *Model Checking* by Clarke, Grumberg and Peled). Show the intermediate steps.

(b) Determine the set of states where (**D**) holds fairly (with fairness constraint $\mathcal{F} = \{\{s_2\}, \{s_3\}\}$), using the symbolic model checking algorithm for CTL. Use explicit set notation to represent states instead of BDDs. Show the intermediate steps.

(c) Determine the set of states where (**E**) holds using the symbolic model checking algorithm for CTL model checking. Show the intermediate steps.

□