Exercise 6

Büchi Automata and LTL Model Checking

1. Build an automaton for the LTL formula $G(p \rightarrow F(q \vee c))$ and one for $F(p \wedge G(\neg q \wedge \neg c)$
   Use GOAL http://goal.im.ntu.edu.tw to check if your automaton is correct.
   GOAL allows you to
   - translate an LTL into an automaton,
   - check the equivalence of two automata,
   - minimize an automaton using simulation relation.

   In order to run goal, first download GOAL (e.g., `GOAL-20111010.zip`) from the homepage, then unzip it (e.g., `unzip GOAL-20111010.zip`) and execute the script goal in the newly created directory `GOAL-20111010` to run the program:
   `./goal`
   or use the installation in my home directory by calling:
   `/home/perms/jobstmab/GOAL-20111010/goal`

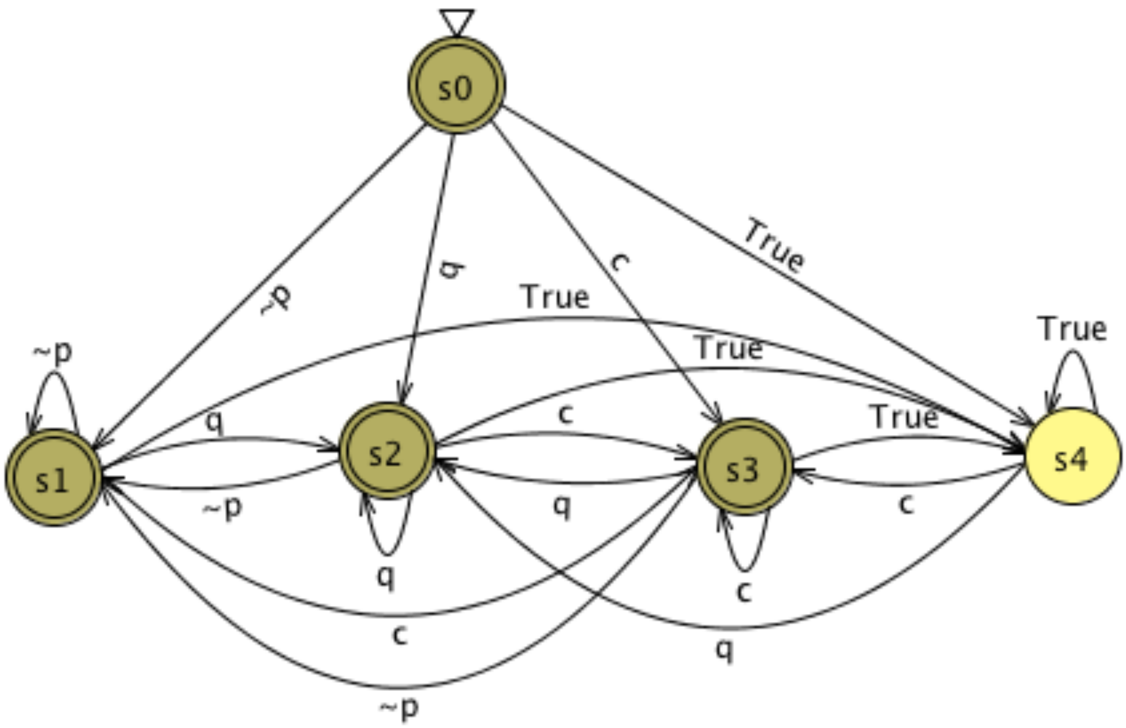
Solution for 1:

$A=G(p \rightarrow F(q \vee c)) = G(!p \vee F(q \vee c))= (!p \vee F(q \vee c)) \wedge X(A)=$
$=(!p \vee (q \vee c) \vee X(F(q \vee c))) \wedge X(A)=$
$=(!p \wedge X(A)) \vee (q \wedge X(A)) \vee (c \wedge X(A)) \vee (X(F(q \vee c)) \wedge X(A)) = B \vee C \vee D \vee E$

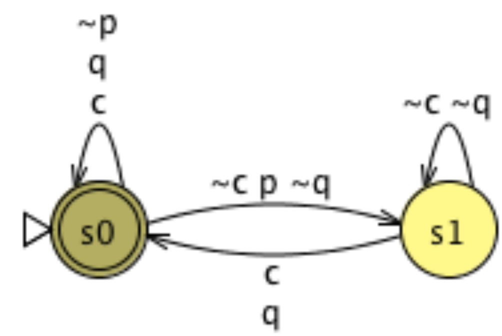
B,C,D have the same edges as A.

E: $(F(q \vee c) \wedge A) = F(q \vee c) \wedge G(!p \vee F(q \vee c)) =$
$= F(q \vee c) \wedge (!p \vee F(q \vee c)) \wedge X(G(!p \vee F(q \vee c))) =$ (note that $y \wedge (y \vee z) = y$)
$= F(q \vee c) \wedge X(G(!p \vee F(q \vee c))) = (q \vee c) \vee X(F(q \vee c)) \wedge X(G(!p \vee F(q \vee c))) =$
$= ((q \wedge X(A)) \vee (c \wedge X(A)) \vee (X(F(q \vee c)) \wedge X(G(!p \vee F(q \vee c)))) = C \vee D \vee E$

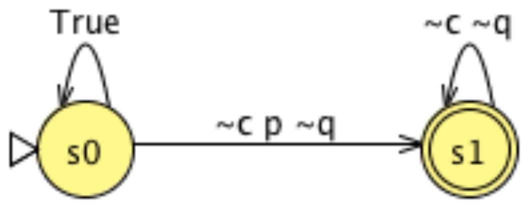
E from all the temporal sub-formula of $A= G(p \rightarrow F(q \vee c))$, which are $F(q \vee c)$ and A itself, only one is an Until-formula, namely $F(q \vee c)$, so we get one set of Büchi states. All the state that do not promise to satisfy $F(q \vee c)$ are accepting. These are the states A,B,C, and D. The automaton is depict below: A=s0, B=s1, C=s2, D=s3, and E=s4



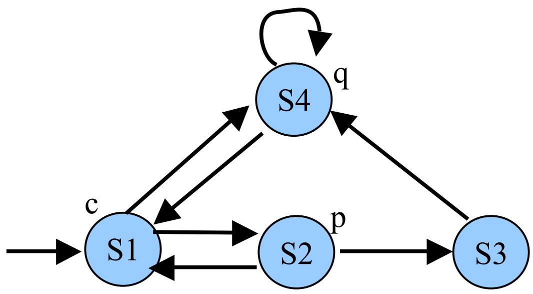
Automaton for $G(p \rightarrow F(q \vee c))$ constructed with GOAL



Automaton for $F(p \wedge G(\neg q \wedge \neg c)$ constructed with GOAL

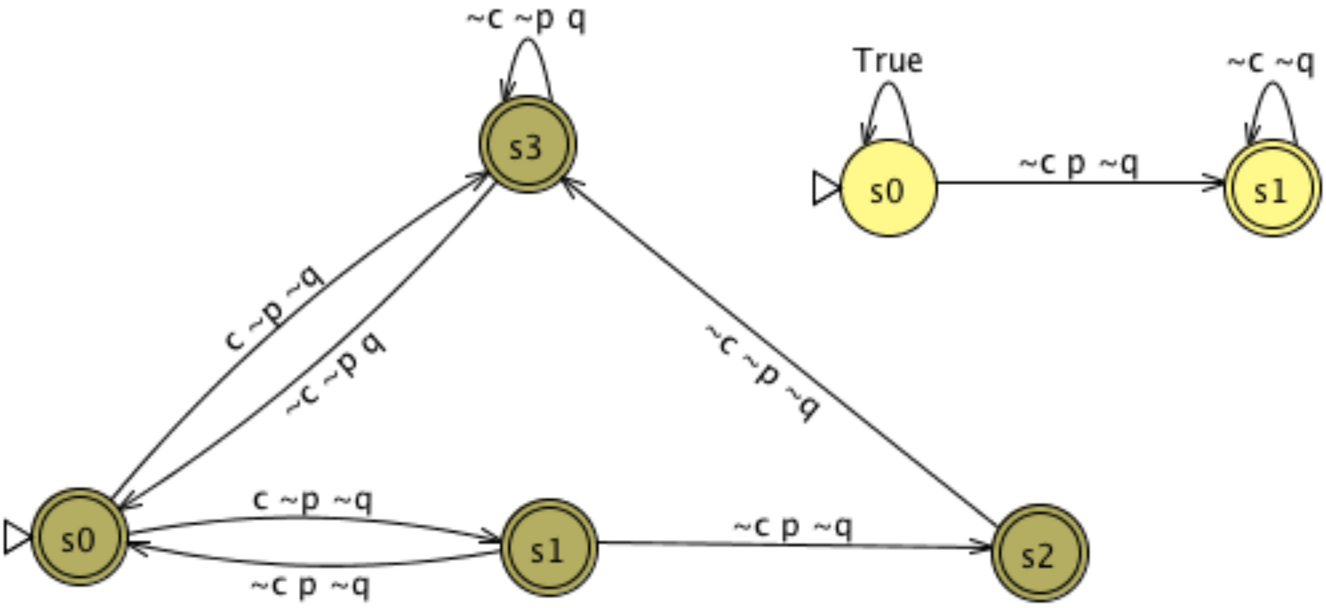


2. Consider the Kripke structure K below, compute if K satisfies $G(p \rightarrow F(q \vee c))$.

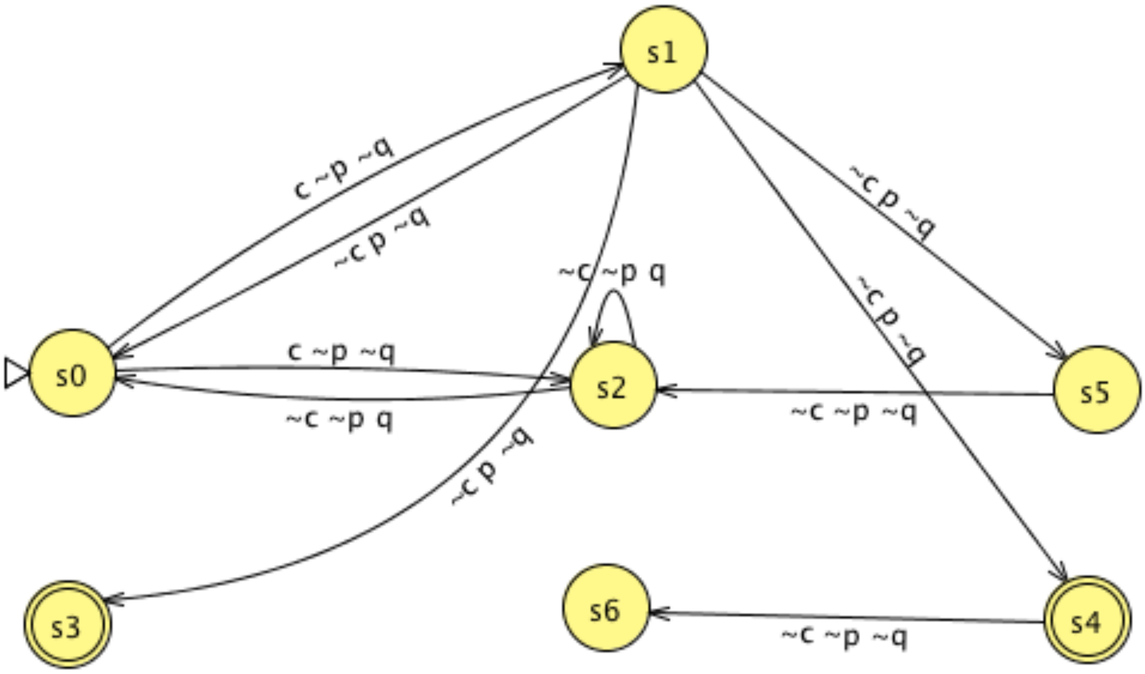


In order to check if K satisfies $G(p \rightarrow F(q \vee c))$., we check if the intersection between a Buchi automaton representing K and a Buchi automaton representing $\neg G(p \rightarrow F(q \vee c)).= F(p \wedge G(\neg q \vee \neg c)$ is empty.

A Buchi automaton for $F(p \wedge G(!q \vee !c)$ can be found in Question 1 (above).
A Buchi automaton representing K is shown on the next page. Note that in the Kripke structure the labels correspond only to a single letter, which means if a state is labeled {p,q}, then only atomic proposition p and q are true, all other propositions are false. While in the **notation used by GOAL**, if we have an edge from s to s' label p q, then p and q have to be true but **all other propositions can either be true OR false.** Below you'll find the structure K using the GOAL notation.



Finally, we construct the produce (shown below) of the automaton for F(p ∧ G(¬q ∧ ¬c) (shown again above on the right) and the automaton for K (shown above on the left).



The language of the product automaton is empty because there exists no infinite path starting from the initial state that visits an accepting state (State s3 and s4) infinitely often. We can also see this by looking at the non-trivial strongly connected components, there is only one non-trivial SCC consisting of s0, s1, s2, and s5. Since this SCC doesn't intersect the accepting states, we have no accepting run and the language is empty.