15-817: Assignment 1

Please do Parts 1 and 2 ASAP!

Part 1. Sign up for the course mailing list. Email the TA (`wklieber@cs.cmu.edu`) with "`[15-817] Subscribe` *YourEmailAddress*" as the subject line; leave the body blank. If there are corrections or clarifications to this assignment, you will be notified by email and the updates will also be posted on the course website.

Part 2. Run Cadence SMV on `simple-01.smv` (the file is on the Assignments page of the course website). Please do this ASAP so that we can help you if you encounter difficulties in getting SMV to work. (You can download Cadence SMV and run it on your computer, or you can run Cadence on the Andrew Unix servers. See the Assignments page for more info.) (You don't need to turn in any work for Part 2.)

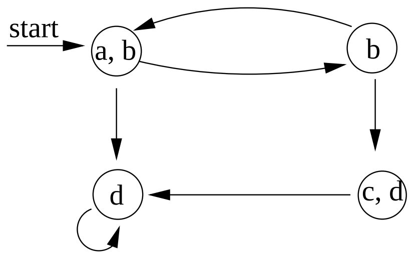
Part 3. Express the following properties in CTL:

- A. "It is invariantly true that: We will necessarily eventually receive a request."
- B1. "It is invariantly true that: If we receive a request, then we will possibly eventually send an acknowledgement."
- B2. "It is invariantly true that: If we receive a request, then we will necessarily eventually send an acknowledgement."
- C. "It is invariantly true that: If we receive a request, then we will necessarily send an acknowledgement in at most two clock cycles (state transititions)."
- D. "Infinitely many requests will occur during the operation of the system." (This is not directly translatable into CTL, but you can write a CTL property that entails it.)
- E. "From any state it is possible to get to a *restart* state."
- F. "From any state it is possible to get to a *restart* state in exactly two steps."
- G. "After p is true, q is never true." (Interpret this ~~as a property for the current state, not~~ as a global invariant.)

Use the following atomic propositions:

- `req` : We receive a request,
- `ack` : We send an acknowledgement,
- `restart` : We are in a *restart* state,

Part 4. Consider the automaton given below:



and the following CTL formulas:

1. $\mathbf{EG}\, d$
2. $\mathbf{EF}\ \mathbf{EG}\, d$
3. $\mathbf{EF}\ \mathbf{AG}\, d$
4. $\mathbf{AG}\ \mathbf{EF}\, d$
5. $\mathbf{A}[b\,\mathbf{U}\,c]$
6. $\mathbf{A}[b\,\mathbf{U}\,d]$

Encode the above automaton as an SMV model. Use one variable to represent the current state. Have this variable range over the four possible states (`Sab`, `Sb`, `Sd`, `Scd`). Define the individual propositional variables (a, b, c, d) in terms of the one state variable. Encode the CTL formulas as specifications to be checked of the model. (Please keep them in order.)

Name your file "*Username*-`hw1-p4.smv`" and submit it using a method to be explained on the course website. Also run SMV on your file and print out the results of whether each property is true or false. (Don't include the traces or stats, just the final answers for each property.)

Part 5. Show that $\mathbf{A}[f\ \mathbf{U}\ g] \equiv \neg\mathbf{E}[\neg g\ \ \mathbf{U}\ \ \neg f \wedge \neg g] \ \wedge\ \neg\mathbf{EG}\,\neg g$

At some points, you may need to justify something by giving an informal explanation in terms of the semantics of CTL*. Try to be as clear and rigorous as possible.

Hints:

- It may be useful to rewrite $[f\ \mathbf{U}\ g]$ as $[(f\ \mathbf{W}\ g)\ \wedge\ \mathbf{F}\ g]$, where $\mathbf{W}$ is the *weak until* operator. "$f\ \mathbf{W}\ g$" has the same semantics as "$f\ \mathbf{U}\ g$" except that g is not required to eventually be true.
- "$\neg g\ \ \mathbf{U}\ \ \neg f \wedge \neg g$" means that we eventually reach a point in time where f and g are both false, and that g cannot be true before we reach such a point.
- Try to establish that $(f\ \mathbf{W}\ g) \equiv \neg(\neg g\ \ \mathbf{U}\ \ \neg f \wedge \neg g)$ using an informal argument, being as clear and rigorous as possible.