Team members:

1. Anowarul Kabir
2. Clayton Mottley
3. Xavier Gitiaux

Contributions:

As always we did our own solutions and discussed over it. In the final solutions, the mandatory portion reflects Anowarul and Clayton’s class solutions and the SMV specifications with additional left turn reflects Xavier and Clayton’s final solutions. Before making the SMV solutions, Clayton and Xavier extensively discussed over the solution outline on slack. Anowarul makes sure that it runs.

1. EF(started & !ready)
2. AG(request -> AF acknowledged)
3. AG(EF enabled)
4. AF(AG deadlocked)
5. AG(EF restart)
6. AG((direction==up & floor==2 & Button5Pressed) -> AX(direction==up U floor==2))
7. AG(floor==3 & doorClosed & idle -> EX(idle))
8. AG(switchClosed==true -> AX(AG(valueOpen!=true)))
9. AG(q -> AX(AG(p!=true U r))
10. AG(toggle -> AX(!toggle))

-- SMV Specification for simple EW,NS intersection

```
MODULE main
VAR
  NE : direction;
  EW : direction;
  NS : direction;
  WE : direction;
  SN : direction;
  ES : direction;
  WN : direction;
  SW : direction;
  EWGreen  : process ToGreen(EW, NS, SN, WN, SW, NE);
  WEGreen  : process ToGreen(WE, NS, SN, ES, NE, SW);
  NSGreen  : process ToGreen(NS, EW, WE, ES, WN, SW);
  SNGreen  : process ToGreen(SN, EW, WE, WN, NE, ES);
  WNGreen  : process ToGreen(WN, EW, NS, SN, SW, NE);
  NEGreen  : process ToGreen(NE, SN, EW, WE, ES, WN);
  SWGreen  : process ToGreen(SW, NS, EW, WE, ES, WN);
  ESGreen  : process ToGreen(ES, WE, NS, SN, SW, NE);
  EWYellow : process ToYellow(EW);
  NSYellow : process ToYellow(NS);
  WEYellow : process ToYellow(WE);
  SNYellow : process ToYellow(SN);
  SWYellow : process ToYellow(SW);
  NEYellow : process ToYellow(NE);
  WNYellow : process ToYellow(WN);
  ESYellow : process ToYellow(ES);
  EWRed : process ToRed(EW);
  NSRed : process ToRed(NS);
  WERed : process ToRed(WE);
  SNRed : process ToRed(SN);
  SWRed : process ToRed(SW);
  NERed : process ToRed(NE);
  WNRed : process ToRed(WN);
  ESRed : process ToRed(ES);


--check for at least one red light in conflicting situations
SPEC
   AG((WE.light = red) | (NS.light = red & SN.light = red & ES.light = red & NE.light = red & SW.light = red))

SPEC
   AG((EW.light = red) | (NS.light = red & SN.light = red & WN.light = red & NE.light = red & SW.light = red))

SPEC
   AG((NS.light = red) | (EW.light = red & WE.light = red & WN.light = red & SW.light = red & ES.light = red))

SPEC
   AG((SN.light = red) | (EW.light = red & WE.light = red & NE.light = red & WN.light = red & ES.light = red))

SPEC
   AG((SW.light = red) | (EW.light = red & WE.light = red & NS.light = red & ES.light = red & WN.light =red))

SPEC
   AG((ES.light = red) | (WE.light = red & SN.light = red & NS.light = red & SW.light = red & NE.light = red))

SPEC
   AG((NE.light = red) | (SN.light = red & EW.light = red & WE.light = red & ES.light = red & WN.light = red))

SPEC
   AG((WN.light = red) | (NS.light = red & EW.light = red & SN.light = red & SW.light = red & NE.light = red))

--check for intervening yellow
SPEC
   AG((NS.light = green ) -> AX (EW.light = red))

-- If the light for a given light is GREEN, then the lights for all conflicting directions are RED.
-- these are non-turning cases
SPEC
   AG((NS.light = green ) -> AX (EW.light = red & WE.light = red & WN.light = red & ES.light = red & SW.light = red))
```

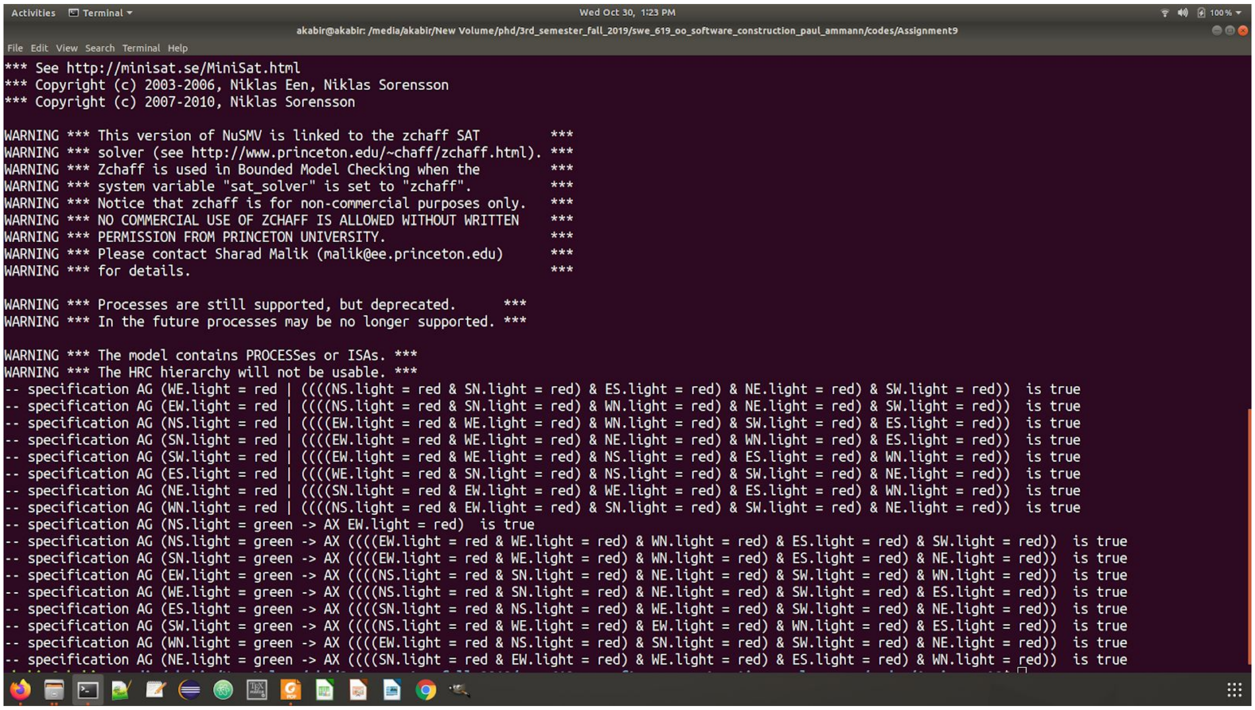
```
SPEC
  AG((SN.light = green ) -> AX (EW.light = red & WE.light = red & WN.light = red & ES.light = red & NE.light = red))

SPEC
  AG((EW.light = green ) -> AX (NS.light = red & SN.light = red & NE.light = red & SW.light = red & WN.light = red))

SPEC
  AG((WE.light = green ) -> AX (NS.light = red & SN.light = red & NE.light = red & SW.light = red & ES.light = red))
-- these are left turning cases
SPEC
  AG((ES.light = green) -> AX (SN.light = red & NS.light = red & WE.light = red & SW.light = red & NE.light = red))

SPEC
  AG((SW.light = green) -> AX (NS.light = red & WE.light = red & EW.light = red & WN.light = red & ES.light = red))

SPEC
  AG((WN.light = green) -> AX (EW.light = red & NS.light = red & SN.light = red & SW.light = red & NE.light = red))

SPEC
  AG((NE.light = green) -> AX (SN.light = red & EW.light = red & WE.light = red & ES.light = red & WN.light = red))


MODULE direction
VAR
  light : {red, yellow, green};
ASSIGN
  init(light) := red;


MODULE ToGreen(dir1, cdir1, cdir2, cdir3, cdir4, cdir5)
ASSIGN
  next(dir1.light) :=
          case
          (dir1.light = red) & (cdir1.light = red) & (cdir2.light = red) & (cdir3.light =red) & (cdir4.light = red) & (cdir5.light
= red): green;
          TRUE : dir1.light;
          esac;

MODULE ToYellow(dir1)
ASSIGN
  next(dir1.light) :=
          case
          (dir1.light = green) : yellow;
          TRUE : dir1.light;
          esac;

MODULE ToRed(dir1)
ASSIGN
  next(dir1.light) :=
          case
```

```
        (dir1.light = yellow) : red;
        TRUE : dir1.light;
        esac;
```

Here is the output:

```
-- specification AG (WE.light = red | ((((NS.light = red & SN.light = red) & ES.light = red) & NE.light = red) & SW.light
= red))  is true
-- specification AG (EW.light = red | ((((NS.light = red & SN.light = red) & WN.light = red) & NE.light = red) & SW.light
= red))  is true
-- specification AG (NS.light = red | ((((EW.light = red & WE.light = red) & WN.light = red) & SW.light = red) & ES.light
= red))  is true
-- specification AG (SN.light = red | ((((EW.light = red & WE.light = red) & NE.light = red) & WN.light = red) & ES.light
= red))  is true
-- specification AG (SW.light = red | ((((EW.light = red & WE.light = red) & NS.light = red) & ES.light = red) & WN.light
= red))  is true
-- specification AG (ES.light = red | ((((WE.light = red & SN.light = red) & NS.light = red) & SW.light = red) & NE.light
= red))  is true
-- specification AG (NE.light = red | ((((SN.light = red & EW.light = red) & WE.light = red) & ES.light = red) & WN.light
= red))  is true
-- specification AG (WN.light = red | ((((NS.light = red & EW.light = red) & SN.light = red) & SW.light = red) & NE.light
= red))  is true
-- specification AG (NS.light = green -> AX EW.light = red)  is true
-- specification AG (NS.light = green -> AX ((((EW.light = red & WE.light = red) & WN.light = red) & ES.light = red) &
SW.light = red))  is true
-- specification AG (SN.light = green -> AX ((((EW.light = red & WE.light = red) & WN.light = red) & ES.light = red) &
NE.light = red))  is true
-- specification AG (EW.light = green -> AX ((((NS.light = red & SN.light = red) & NE.light = red) & SW.light = red) &
WN.light = red))  is true
-- specification AG (WE.light = green -> AX ((((NS.light = red & SN.light = red) & NE.light = red) & SW.light = red) &
ES.light = red))  is true
-- specification AG (ES.light = green -> AX ((((SN.light = red & NS.light = red) & WE.light = red) & SW.light = red) &
NE.light = red))  is true
-- specification AG (SW.light = green -> AX ((((NS.light = red & WE.light = red) & EW.light = red) & WN.light = red) &
ES.light = red))  is true
-- specification AG (WN.light = green -> AX ((((EW.light = red & NS.light = red) & SN.light = red) & SW.light = red) &
NE.light = red))  is true
-- specification AG (NE.light = green -> AX ((((SN.light = red & EW.light = red) & WE.light = red) & ES.light = red) &
WN.light = red))  is true
```

The screen-shot goes to next page that shows it runs and spec are correct.



Activities
Terminal
Wed Oct 30, 1:23 PM
100 %
akabir@akabir: /media/akabir/New Volume/phd/3rd_semester_fall_2019/swe_619_oo_software_construction_paul_ammann/codes/Assignment9
File
Edit
View
Search
Terminal
Help

```
*** See http://minisat.se/MiniSat.html
*** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
*** Copyright (c) 2007-2010, Niklas Sorensson

WARNING *** This version of NuSMV is linked to the zchaff SAT         ***
WARNING *** solver (see http://www.princeton.edu/~chaff/zchaff.html). ***
WARNING *** Zchaff is used in Bounded Model Checking when the         ***
WARNING *** system variable "sat_solver" is set to "zchaff".          ***
WARNING *** Notice that zchaff is for non-commercial purposes only.   ***
WARNING *** NO COMMERCIAL USE OF ZCHAFF IS ALLOWED WITHOUT WRITTEN    ***
WARNING *** PERMISSION FROM PRINCETON UNIVERSITY.                     ***
WARNING *** Please contact Sharad Malik (malik@ee.princeton.edu)      ***
WARNING *** for details.                                              ***

WARNING *** Processes are still supported, but deprecated.      ***
WARNING *** In the future processes may be no longer supported. ***

WARNING *** The model contains PROCESSes or ISAs. ***
WARNING *** The HRC hierarchy will not be usable. ***
-- specification AG (WE.light = red | ((((NS.light = red & SN.light = red) & ES.light = red) & NE.light = red) & SW.light = red))  is true
-- specification AG (EW.light = red | ((((NS.light = red & SN.light = red) & WN.light = red) & NE.light = red) & SW.light = red))  is true
-- specification AG (NS.light = red | ((((EW.light = red & WE.light = red) & WN.light = red) & SW.light = red) & ES.light = red))  is true
-- specification AG (SN.light = red | ((((EW.light = red & WE.light = red) & NE.light = red) & WN.light = red) & ES.light = red))  is true
-- specification AG (SW.light = red | ((((EW.light = red & WE.light = red) & NS.light = red) & ES.light = red) & WN.light = red))  is true
-- specification AG (ES.light = red | ((((WE.light = red & SN.light = red) & NS.light = red) & SW.light = red) & NE.light = red))  is true
-- specification AG (NE.light = red | ((((SN.light = red & EW.light = red) & WE.light = red) & ES.light = red) & WN.light = red))  is true
-- specification AG (WN.light = red | ((((NS.light = red & EW.light = red) & SN.light = red) & SW.light = red) & NE.light = red))  is true
-- specification AG (NS.light = green -> AX EW.light = red)  is true
-- specification AG (NS.light = green -> AX ((((EW.light = red & WE.light = red) & WN.light = red) & ES.light = red) & SW.light = red))  is true
-- specification AG (SN.light = green -> AX ((((EW.light = red & WE.light = red) & WN.light = red) & ES.light = red) & NE.light = red))  is true
-- specification AG (EW.light = green -> AX ((((NS.light = red & SN.light = red) & NE.light = red) & SW.light = red) & WN.light = red))  is true
-- specification AG (WE.light = green -> AX ((((NS.light = red & SN.light = red) & NE.light = red) & SW.light = red) & ES.light = red))  is true
-- specification AG (ES.light = green -> AX ((((SN.light = red & NS.light = red) & WE.light = red) & SW.light = red) & NE.light = red))  is true
-- specification AG (SW.light = green -> AX ((((NS.light = red & WE.light = red) & EW.light = red) & WN.light = red) & ES.light = red))  is true
-- specification AG (WN.light = green -> AX ((((EW.light = red & NS.light = red) & SN.light = red) & SW.light = red) & NE.light = red))  is true
-- specification AG (NE.light = green -> AX ((((SN.light = red & EW.light = red) & WE.light = red) & ES.light = red) & WN.light = red))  is true
```