Lab8: PVS/PLC Function Table Secifications

EECS4312 JSO

November 6, 2018

Revisions

Date	Revision	Description
12 September 2017	1.0	Initial notes for this document
28 September 2018	2.0	Revise

1 Precondition

Make sure you are up to date with previous Labs, required self-directed readings. You should already be skilled in the use of the PVS tool with propositional logic, predicate logic, set theory, induction etc. applied to the **specification** of hardware and software systems.

2 Goals

- Using PVS, understand and apply function tables to check completeness and disjointness of specifications
- Use PVS function tables to specify and validate hard real-time systems

3 To Do: top.pvs

You must specify and prove theories listed below in the `top.pvs` file below:

```
% Exercises for Lab8
% proveit --importchain --clean top.pvs
top : THEORY
BEGIN
  IMPORTING Hysteresis_tcc % Hysteresis non-disjoint
  IMPORTING Hysteresis
  IMPORTING test
  IMPORTING Limits_Alarm
  IMPORTING bridge
END top
```

See the `top.summary` file listing all the specification proofs you must complete. Yours might be slightly different for the Car-Interlock (I added a function `car_enter` to help with the function table; you may do it differently).

4 PLC function table specifications in PVS (slides 15)

4.1 Hysteresis_TCC.pvs

The PLC standard provides a basic Hysteresis block that can be used to build bigger systems.

```
    %         +-------------+
    %         | HYSTERESIS  |
    % REAL---|XIN1        Q|---BOOL
    % REAL---|XIN2         |
    % REAL---|EPS          |
    %         +-------------+

    % monitored variables
    XIN1 : VAR [DTIME -> real] % signal value
```

```
    XIN2 : VAR [DTIME -> real] % set-point
    EPS  : VAR [DTIME -> real] % hysteresis band size
```

The Hysteresis block in the standard is informal and thus is not a precise specification of this block. In `Hysteresis_TCC.pvs`, you must explore what conditions are needed to correct and validate the standard. At the end, you should be able to achieve the following:

```
Proof summary for theory Hysteresis_tcc
   BOOL_TCC1.............................proved - complete   [shostak](0.01 s)
   hysteresis_st_TCC1....................proved - complete   [shostak](0.03 s)
   hysteresis_st_TCC2....................proved - complete   [shostak](0.00 s)
   hysteresis_st_TCC3....................proved - complete   [shostak](0.02 s)
   hysteresis_req1_TCC1..................unfinished          [shostak](0.05 s)
   hysteresis_req1_TCC2..................proved - complete   [shostak](0.02 s)
   correct_hysteresis_st1................unfinished          [shostak](0.11 s)
   correct_hysteresis_st2................proved - incomplete [shostak](0.17 s)
   Theory totals: 8 formulas, 8 attempted, 6 succeeded (0.42 s)
```

- `BOOL_TCC1`: You will need to prove this TCC
- `hysteresis_req1_TCC1`: **unfinished** because it is not a disjoint specification of the Hysteresis block. This cannot be proved because the standard is missing an environmental assumption. By grinding we can simplify the proof obligation and from a counterexample see what environment assumption we need.
- `correct_hysteresis_st1`: **unfinished**. Asserts that the implementation of the Hysteresis block satisfies its specification. Cannot be proved without the relevant environmental assumption.
- `env (XIN1, XIN2, EPS)`: complete this environmental assumption and use it to prove `correct_hysteresis_st2`. The proof is marked incomplete due to an unfinished TCC (but we could in principal delete the incorrect descriptions).

It's ok to use `(grind)` in this example.

4.2 Hysteresis.pvs

Incorporate the environmental assumption needed for the Hysteresis block in `Hysteresis.pvs` and prove all the TCCs and conjectures

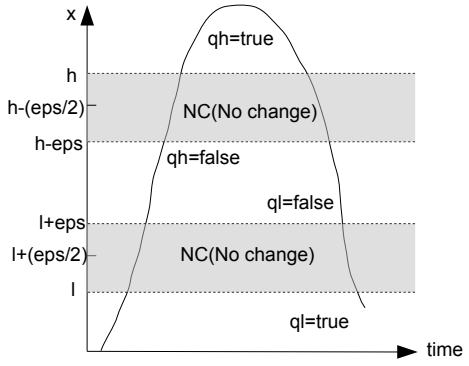
without `grind`. This will require you to prove all the logical paths. There are three TBDs (to be dones).

You should obtain the following:

```
Proof summary for theory Hysteresis
   BOOL_TCC1.............................proved - complete   [shostak](0.00 s)
   hysteresis_st_TCC1....................proved - complete   [shostak](0.02 s)
   hysteresis_st_TCC2....................proved - complete   [shostak](0.00 s)
   hysteresis_st_TCC3....................proved - complete   [shostak](0.00 s)
   hysteresis_req_TCC1...................proved - complete   [shostak](0.06 s)
   hysteresis_req_TCC2...................proved - complete   [shostak](0.03 s)
   correct_hysteresis_st.................proved - complete   [shostak](0.13 s)
   Theory totals: 7 formulas, 7 attempted, 7 succeeded (0.25 s)
```

4.3 Limits_Alarm.pvs: slides 15

The figure illustrates the functionality of the Limits Alarm:



- Build a more complex limit-alarm function block from basic blocks such as Hysteresis, minus plus, divide and gte_one etc.
- Check its completeness and disjointness
- Validate the limit-alarm block by showing that its specification entails certain system safety invariants

Fig. 1 shows the PVS specification of the Monitored and Controlled variables.

The PVS file that you are provided with will not be type correct. Once you have it in a type correct format, the validation of TCCS and

```
% Input-Output Declarations of the Limits_Alarm function block
% ============================================================
%                            +---------+
%                            | LIMITS_ |
%                            |  ALARM  |
%     (* High limit *) REAL--|H      QH|--BOOL (* High flag *)
% (* Variable value *) REAL--|X       Q|--BOOL (* Alarm output *)
%    (* Lower limit *) REAL--|L      QL|--BOOL (* Low flag *)
%     (* Hysteresis *) REAL--|EPS      |
%                            +---------+

% ============================================================
% Monitored Variables
% ============================================================
H   : VAR [DTIME -> real]  % high limit
X   : VAR [DTIME -> real]  % signal value
L   : VAR [DTIME -> real]  % low limit
EPS : VAR [DTIME -> real]  % hysteresis band size

% ============================================================
% controlled variables
% ============================================================
QH  : VAR [DTIME -> BOOL]
Q   : VAR [DTIME -> BOOL]
QL  : VAR [DTIME -> BOOL]
```

Figure 1: Monitored and Controlled variables

```
Proof summary for theory Limits_Alarm
   high_alarm_req_TCC1...................proved - complete   [shostak](0.06 s)
   high_alarm_req_TCC2...................proved - complete   [shostak](0.04 s)
   high_alarm_req_TCC3...................proved - complete   [shostak](0.03 s)
   high_alarm_req_TCC4...................proved - complete   [shostak](0.02 s)
   high_alarm_req_TCC5...................proved - complete   [shostak](0.01 s)
   low_alarm_req_TCC1....................proved - complete   [shostak](0.03 s)
   low_alarm_req_TCC2....................proved - complete   [shostak](0.04 s)
   low_alarm_req_TCC3....................proved - complete   [shostak](0.03 s)
   correct_limits_alarm_fbd..............untried              [Untried]( n/a s)
   req_entails_inv.......................untried              [Untried]( n/a s)
   limits_alarm_req2_TCC1................unfinished           [shostak](0.18 s)
   limits_alarm_req2_TCC2................proved - complete   [shostak](0.05 s)
   Theory totals: 12 formulas, 10 attempted, 9 succeeded (0.50 s)
```

Figure 2: Validation TCCS and conjectures to be proved

conjectures to be proved is shown in Fig. 2. Some of the TCCs can be proved automatically.

To Do

Complete and prove all the specifications in the PVS file. With an appropriate top.pvs you should see something like:

```
***
*** top (14:17:11 11/5/2018)
*** Generated by proveit - ProofLite-6.0.9 (3/14/14)
*** Trusted Oracles
***   MetiTarski: MetiTarski Theorem Prover via PVS proof rule metit
***
 Proof summary for theory top
    Theory totals: 0 formulas, 0 attempted, 0 succeeded (0.00 s)

 Proof summary for theory Time
    r2d_TCC1.............................proved - complete   [shostak](0.16 s)
    d2r_TCC1.............................proved - complete   [shostak](0.02 s)
    held_for_TCC1........................proved - complete   [shostak](0.05 s)
    Theory totals: 3 formulas, 3 attempted, 3 succeeded (0.22 s)

 Proof summary for theory Hysteresis
    BOOL_TCC1............................proved - complete   [shostak](0.00 s)
    hysteresis_st_TCC1...................proved - complete   [shostak](0.00 s)
    hysteresis_st_TCC2...................proved - complete   [shostak](0.00 s)
    hysteresis_st_TCC3...................proved - complete   [shostak](0.00 s)
    hysteresis_req_TCC1..................proved - complete   [shostak](0.06 s)
    hysteresis_req_TCC2..................proved - complete   [shostak](0.04 s)
    correct_hysteresis_st................proved - complete   [shostak](0.19 s)
    Theory totals: 7 formulas, 7 attempted, 7 succeeded (0.30 s)

 Proof summary for theory Limits_Alarm
    high_alarm_req_TCC1..................proved - complete   [shostak](0.03 s)
    high_alarm_req_TCC2..................proved - complete   [shostak](0.05 s)
    high_alarm_req_TCC3..................proved - complete   [shostak](0.02 s)
    high_alarm_req_TCC4..................proved - complete   [shostak](0.01 s)
    high_alarm_req_TCC5..................proved - complete   [shostak](0.02 s)
    low_alarm_req_TCC1...................proved - complete   [shostak](0.03 s)
    low_alarm_req_TCC2...................proved - complete   [shostak](0.05 s)
    low_alarm_req_TCC3...................proved - complete   [shostak](0.04 s)
    correct_limits_alarm_fbd.............proved - complete   [shostak](0.82 s)
    req_entails_inv......................proved - complete   [shostak](0.41 s)
    limits_alarm_req2_TCC1...............proved - complete   [shostak](0.21 s)
    limits_alarm_req2_TCC2...............proved - complete   [shostak](0.05 s)
    Theory totals: 12 formulas, 12 attempted, 12 succeeded (1.73 s)

Grand Totals: 22 proofs, 22 attempted, 22 succeeded (2.26 s)
```

5 bridge.pvs

Read and then do the work described in the accompanying **bridge.pdf**.