

EECS3342 System Specification and Refinement  
(Winter 2022)

Q&A - Week 10 Lectures

Thursday, March 31

## Announcements

+ Lecture W11 released

+ Written Test 4

FTP

Here we state that ml\_pass and il\_pass are 1 because init does not establish them and neither ML\_tl\_green or IL\_tl\_green have occurred, but if init does not establish them, would they not have a nondeterministic value prior to the first occurrences of ML\_tl\_green or IL\_tl\_green?

d = 2	ml_pass	il_pass
< init,	1	1
ML_tl_green,	0	1
ML_out_1,	1	1
ML_out_2,	1	1
IL_in,	1	1
IL_in,	1	1
IL_tl_green,	1	0
IL_out_1,	1	1
IL_out_2,	1	1
ML_in,	1	1
ML_in	1	1
>		

variant

ml\_pass + il\_pass

say init does not initialize:

ml\_pass: 0

il\_pass: 0

ML\_tl\_green  $\rightarrow$  ml\_pass: 0  
il\_pass: 0

✓  
ml\_tl, il\_tl both read

$v: 2$

For the PO of DLF in  $m_2$ , why do we not include the abstract guards from  $m_0$ ? We include the axioms and invariants from that model, so why not the guards as well?

*Safety Properties (should be preserved across models)*

```

axm0.1  d ∈ ℕ
axm0.2  d > 0
axm2.1  COLOUR = {green, red}
axm2.2  green ≠ red

inv0.1  n ∈ ℕ
inv0.2  n ≤ d
inv1.1  a ∈ ℕ
inv1.2  b ∈ ℕ
inv1.3  c ∈ ℕ
inv1.4  a + b + c = n
inv1.5  a = 0 ∨ c = 0
inv2.1  ml.tl ∈ COLOUR
inv2.2  il.tl ∈ COLOUR
inv2.3  ml.tl = green ⇒ a + b < d ∧ c = 0
inv2.4  il.tl = green ⇒ b > 0 ∧ a = 0
inv2.5  ml.tl = red ∨ il.tl = red
inv2.6  ml.pass ∈ {0, 1}
inv2.7  il.pass ∈ {0, 1}
inv2.8  ml.tl = red ⇒ ml.pass = 1
inv2.9  il.tl = red ⇒ il.pass = 1
    
```

### Abstract $m_1$

variables:  $a, b, c$

invariants:  
 inv1.1:  $a \in \mathbb{N}$   
 inv1.2:  $b \in \mathbb{N}$   
 inv1.3:  $c \in \mathbb{N}$   
 inv1.4:  $a + b + c = n$   
 inv1.5:  $a = 0 \vee c = 0$

ML.out  
 when  
 $a + b < d$   
 $c > 0$   
 then  
 $a := a + 1$   
 end

ML.in  
 when  
 $c > 0$   
 then  
 $c := c - 1$   
 end

IL.in  
 when  
 $a > 0$   
 then  
 $a := a - 1$   
 $b := b + 1$   
 end

IL.out  
 when  
 $b > 0$   
 $a = 0$   
 then  
 $b := b - 1$   
 $c := c + 1$   
 end

### Concrete $m_2$

ML.tl.green  
 when  
 $ml.tl = red$   
 $a + b < d$   
 $c = 0$   
 $il.pass = 1$   
 then  
 $ml.tl := green$   
 $il.tl := red$   
 $ml.pass := 0$   
 end

IL.tl.green  
 when  
 $il.tl = red$   
 $b > 0$   
 $a = 0$   
 $ml.pass = 1$   
 then  
 $il.tl := green$   
 $ml.tl := red$   
 $il.pass := 0$   
 end

ML.out.1  
 when  
 $ml.tl = green$   
 $a + b + 1 \neq d$   
 then  
 $a := a + 1$   
 $ml.pass := 1$   
 end

IL.out.1  
 when  
 $il.tl = green$   
 $b = 1$   
 then  
 $b := b - 1$   
 $c := c + 1$   
 $il.pass := 1$   
 end

ML.out.2  
 when  
 $ml.tl = green$   
 $a + b + 1 = d$   
 then  
 $a := a + 1$   
 $ml.tl := red$   
 $ml.pass := 1$   
 end

IL.out.2  
 when  
 $il.tl = green$   
 $b = 1$   
 then  
 $b := b - 1$   
 $c := c + 1$   
 $il.tl := red$   
 $il.pass := 1$   
 end

IL.in  
 when  
 $a > 0$   
 then  
 $a := a - 1$   
 $b := b + 1$   
 end

ML.in  
 when  
 $c > 0$   
 then  
 $c := c - 1$   
 end

Disjunction of abstract guards

$a + b < d \wedge c = 0$   
 $c > 0$   
 $a > 0$   
 $b > 0 \wedge a = 0$

guards of ML.out in  $m_1$   
 guards of ML.in in  $m_1$   
 guards of IL.in in  $m_1$   
 guards of IL.out in  $m_1$

Disjunction of concrete guards

$ml.tl = red \wedge a + b < d \wedge c = 0 \wedge il.pass = 1$   
 $il.tl = red \wedge b > 0 \wedge a = 0 \wedge ml.pass = 1$   
 $ml.tl = green \wedge a + b + 1 \neq d$   
 $ml.tl = green \wedge a + b + 1 = d$   
 $il.tl = green \wedge b = 1$   
 $il.tl = green \wedge b = 1$   
 $a > 0$   
 $c > 0$

guards of ML.tl.green in  $m_2$   
 guards of IL.tl.green in  $m_2$   
 guards of ML.out.1 in  $m_2$   
 guards of ML.out.2 in  $m_2$   
 guards of IL.out.1 in  $m_2$   
 guards of IL.out.2 in  $m_2$   
 guards of ML.in in  $m_2$   
 guards of IL.in in  $m_2$

*abstract guards for proving relative DLF of  $m_2$  only.*

Towards the end of the lecture you explain the splitting of events.

I was wondering for ML\_out.1, the guard,

would  $a + b + 1 < d$  be a better design than  $a + b + 1 \neq d$ ?

If we reach the edge case where we hit the max amount of cars, the guard would still pass

Or is the ML\_tl.green guard ( $a + b < d$ ) sufficient enough to resolve this issue since we have the predecessor statement "ml\_tl = green"?

The same applies to IL\_out.1 where  $b > 1$  instead of  $b \neq 1$ .

Thanks!

disjunction of guards of ML\_out.1 and ML\_out.2  

$$: (P \wedge Q) \vee (P \wedge \neg Q) \equiv P$$

```
ML_out.1
when
  ml_tl = green
  a + b + 1 ≠ d
then
  a := a + 1
end
```

```
ML_out.2
when
  ml_tl = green
  a + b + 1 = d
then
  a := a + 1
  ml_tl := red
end
```

-  $a + b + 1 < d \vee a + b + 1 = d \neq \text{TRUE}$

- Simplification of DLF

```
IL_out.1
when
  il_tl = green
  b ≠ 1 = b - 1 ≠ 0
then
  b := b - 1
  c := c + 1
end
```

```
IL_out.2
when
  il_tl = green
  b = 1 = b - 1 = 0
then
  b := b - 1
  c := c + 1
  il_tl := red
end
```

ML\_out when  
 ml\_tl = green  
 then  
 end  
 $a := a + 1$

Disjunction of abstract guards



Disjunction of concrete guards

$a + b < d \wedge c = 0$  guards of ML\_out in  $m_1$   
 $c > 0$  guards of ML\_in in  $m_1$   
 $a > 0$  guards of IL\_in in  $m_1$   
 $b > 0 \wedge a = 0$  guards of IL\_out in  $m_1$

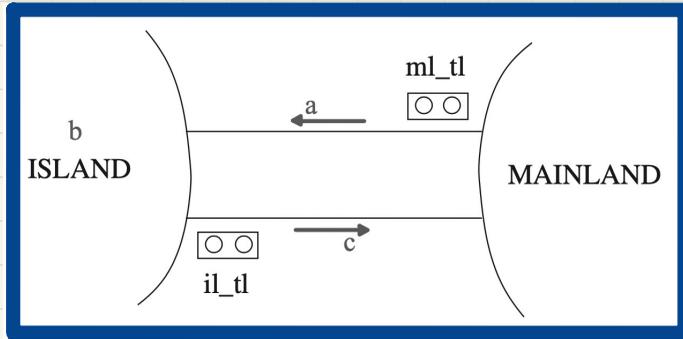
$ml\_tl = red \wedge a + b < d \wedge c = 0 \wedge il\_pass = 1$   
 $il\_tl = red \wedge b > 0 \wedge a = 0 \wedge ml\_pass = 1$   
 $ml\_tl = green \wedge a + b + 1 \neq d$   
 $ml\_tl = green \wedge a + b + 1 = d$   
 $il\_tl = green \wedge b \neq 1$   
 $il\_tl = green \wedge b = 1$   
 $a > 0$   
 $c > 0$

ml\_tl = green

When trying to figure out  $\text{inv2\_3}$ , can you explain how  $\text{ml\_tl} = \text{green} \Rightarrow a+b < d \wedge c=0$

at least one more car

can be allowed to exit from ML to BAI.



invariants:

$\text{inv2\_1} : \text{ml\_tl} \in \text{COLOUR}$

$\text{inv2\_2} : \text{il\_tl} \in \text{COLOUR}$

$\text{inv2\_3} : \text{ml\_tl} = \text{green} \Rightarrow a+b < d \wedge c=0$

$\text{inv2\_4} : \text{il\_tl} = \text{green} \Rightarrow b > 0 \wedge a=0$

$$\underline{a+b+1} \leq d$$