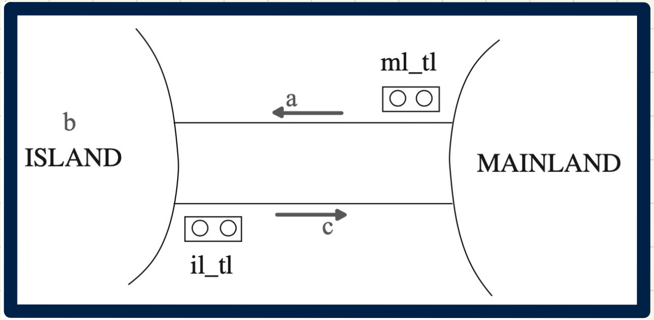
Fixing m2: Adding Actions



ML_tl_green
  when
    $ml\_tl = red$
    $a + b < d$
    $c = 0$
  then
    $ml\_tl := green$
    $il\_tl := red$
  end

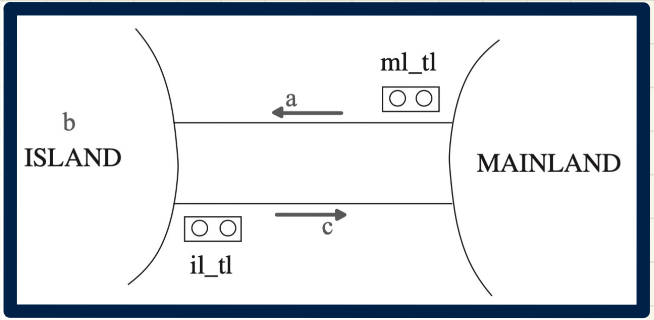
IL_tl_green
  when
    $il\_tl = red$
    $b > 0$
    $a = 0$
  then
    $il\_tl := green$
    $ml\_tl := red$
  end

ML_tl_green/inv2_5/INV

axm0_1	$\{\ d \in \mathbb{N}$
axm0_2	$\{\ d > 0$
axm2_1	$\{\ COLOUR = \{green, red\}$
axm2_2	$\{\ green \neq red$
inv0_1	$\{\ n \in \mathbb{N}$
inv0_2	$\{\ n \leq d$
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	$\{\ a = 0 \lor c = 0$
inv2_1	$\{\ ml\_tl \in COLOUR$
inv2_2	$\{\ il\_tl \in COLOUR$
inv2_3	$\{\ ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4	$\{\ il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5	$\{\ ml\_tl = red \lor il\_tl = red$

Exercise: Specify IL_tl_green/inv2_5/INV

Invariant Preservation: ML_out/inv2_3/INV



variables:
a, b, c
ml_tl
il_tl

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

IL_out
when
$il\_tl = green$
then
$b := b - 1$
$c := c + 1$
end

invariants:
inv2_1 : $ml\_tl \in COLOUR$
inv2_2 : $il\_tl \in COLOUR$
inv2_3 : $ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
inv2_4 : $il\_tl = green \Rightarrow b > 0 \wedge a = 0$

ML_out/inv2_3/INV

axm0_1	$\{ d \in \mathbb{N}$
axm0_2	$\{ d > 0$
axm2_1	$\{ COLOUR = \{green, red\}$
axm2_2	$\{ green \neq red$
inv0_1	$\{ n \in \mathbb{N}$
inv0_2	$\{ n \leq d$
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	$\{ a = 0 \vee c = 0$
inv2_1	$\{ ml\_tl \in COLOUR$
inv2_2	$\{ il\_tl \in COLOUR$
inv2_3	$\{ ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
inv2_4	$\{ il\_tl = green \Rightarrow b > 0 \wedge a = 0$
inv2_5	$\{ ml\_tl = red \vee il\_tl = red$
Concrete guards of *ML_out*	$\{ ml\_tl = green$
	$\vdash$
Concrete invariant **inv2_3** with *ML_out*'s effect in the post-state	$\{ ml\_tl = green \Rightarrow (a + 1) + b < d \wedge c = 0$

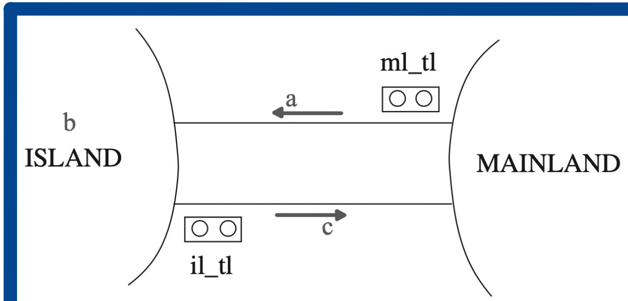
Exercise: Specify IL_out/inv2_4/INV

Discharging POs of m2: Invariant Preservation

First Attempt

$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$ml\_tl = green \Rightarrow (a + 1) + b < d \land c = 0$

ML_out/inv2_3/INV



$$\frac{H \vdash P \qquad H \vdash Q}{H \vdash P \land Q}\ \textbf{AND\_R}$$

$$\frac{H, P, Q \vdash R}{H, P \land Q \vdash R}\ \textbf{AND\_L}$$

$$\frac{H, P \vdash Q}{H \vdash P \Rightarrow Q}\ \textbf{IMP\_R}$$

MON

$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$\vdash$
$ml\_tl = green \Rightarrow (a + 1) + b < d \land c = 0$

IMP_R

$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$ml\_tl = green$
$\vdash$
$(a + 1) + b < d \land c = 0$

IMP_R

$a + b < d \land c = 0$
$ml\_tl = green$
$\vdash$
$(a + 1) + b < d \land c = 0$

AND_L

$a + b < d$
$c = 0$
$ml\_tl = green$
$\vdash$
$(a + 1) + b < d \land c = 0$

AND_R

$a + b < d$
$c = 0$
$ml\_tl = green$
$\vdash$
$(a + 1) + b < d$

??

$a + b < d$
$c = 0$
$ml\_tl = green$
$\vdash$
$c = 0$

HYP

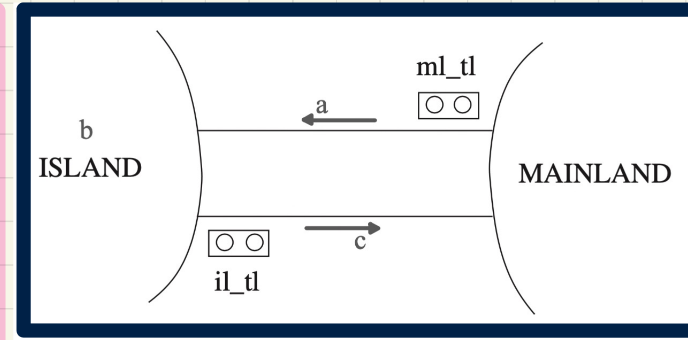
Understanding the Failed Proof on INV

variables:
a, b, c
ml_tl
il_tl

invariants:
inv2_1 : $ml\_tl \in COLOUR$
inv2_2 : $il\_tl \in COLOUR$
inv2_3 : $ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
inv2_4 : $il\_tl = green \Rightarrow b > 0 \wedge a = 0$

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

IL_out
  when
    $il\_tl = green$
  then
    $b := b - 1$
    $c := c + 1$
  end

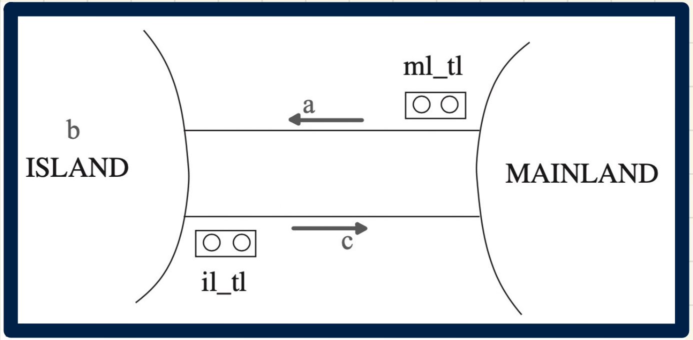


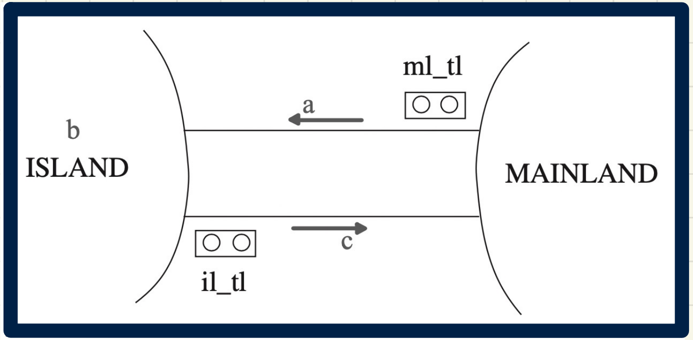
Unprovable Sequent:

$$
\begin{array}{ll}
 & a + b < d \\
\wedge & c = 0 \\
\wedge & ml\_tl = green \\
\vdash & \\
 & (a + 1) + b < d
\end{array}
$$

$d = 3, b = 0, a = 0$	[$(a + 1) + b < d$ evaluates to ***true***]
$d = 3, b = 1, a = 0$	[$(a + 1) + b < d$ evaluates to ***true***]
$d = 3, b = 0, a = 1$	[$(a + 1) + b < d$ evaluates to ***true***]
$d = 3, b = 0, a = 2$	[$(a + 1) + b < d$ evaluates to ***false***]
$d = 3, b = 1, a = 1$	[$(a + 1) + b < d$ evaluates to ***false***]
$d = 3, b = 2, a = 0$	[$(a + 1) + b < d$ evaluates to ***false***]

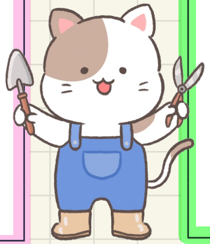
Fixing m2: Splitting Events





ML_out_1
when
$ml\_tl = green$
$a + b + 1 \neq d$
then
$a := a + 1$
end

ML_out_2
when
$ml\_tl = green$
$a + b + 1 = d$
then
$a := a + 1$
$ml\_tl := red$
end



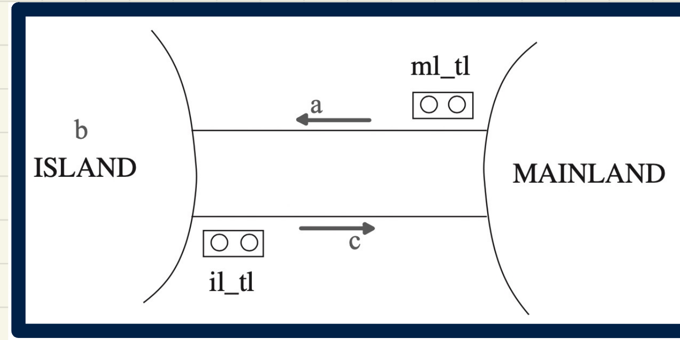
IL_out_1
when
$il\_tl = green$
$b \neq 1$
then
$b := b - 1$
$c := c + 1$
end

IL_out_2
when
$il\_tl = green$
$b = 1$
then
$b := b - 1$
$c := c + 1$
$il\_tl := red$
end

Current m2 May Livelock

ML_tl_green
  when
    $ml\_tl = red$
    $a + b < d$
    $c = 0$
  then
    $ml\_tl := green$
    $il\_tl := red$
  end

IL_tl_green
  when
    $il\_tl = red$
    $b > 0$
    $a = 0$
  then
    $il\_tl := green$
    $ml\_tl := red$
  end



$\langle$ *init*, *ML_tl_green*, *ML_out_1*, *IL_in*, *IL_tl_green*, *ML_tl_green*, *IL_tl_green*, $\ldots\rangle$

init	*ML_tl_green*	*ML_out_1*	*IL_in*	*IL_tl_green*	*ML_tl_green*	*IL_tl_green*
$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$
$a' = 0$	$a' = 0$	$a' = 1$	$a' = 0$	$a' = 0$	$a' = 0$	$a' = 0$
$b' = 0$	$b' = 0$	$b' = 0$	$b' = 1$	$b' = 1$	$b' = 1$	$b' = 1$
$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$
$ml\_tl = red$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = red$	$ml\_tl' = green$	$ml\_tl' = red$
$il\_tl = red$	$il\_tl' = red$	$il\_tl' = red$	$il\_tl' = red$	$il\_tl' = green$	$il\_tl' = red$	$il\_tl' = green$

Fixing m2: Regulating Traffic Light Changes

Divergence Trace: <init, ML_tl_green, ML_out_1, IL_in, IL_tl_green, ML_tl_green, IL_tl_green, ...>

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

ML_out_2
when
$ml\_tl = green$
$a + b + 1 = d$
then
$a := a + 1$
$ml\_tl := red$
$ml\_pass := 1$
end

IL_out_1
when
$il\_tl = green$
$b \neq 1$
then
$b := b - 1$
$c := c + 1$
$il\_pass := 1$
end

IL_out_2
when
$il\_tl = green$
$b = 1$
then
$b := b - 1$
$c := c + 1$
$il\_tl := red$
$il\_pass := 1$
end

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

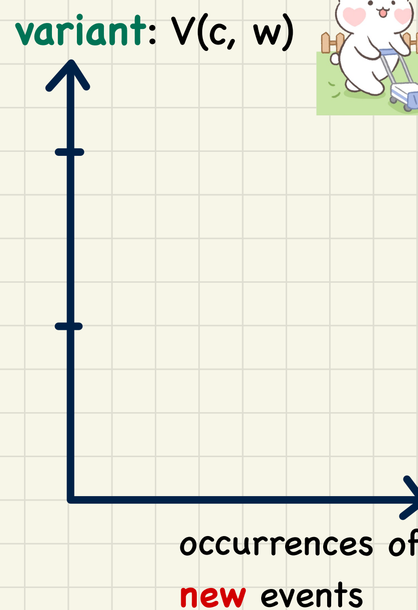
Fixing m2: Measuring Traffic Light Changes

```
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
```

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
>		

variants: *ml_pass* + *il_pass*



PO of Convergence/Non-Divergence/Livelock Freedom

A New Event Occurrence Decreases Variant

$$\frac{\begin{array}{l} A(c) \\ I(c,v) \\ J(c,v,w) \\ H(c,w) \\ \vdash \\ V(c,F(c,w)) < V(c,w) \end{array}}{}\ \text{VAR}$$

Variants: ml_pass + il_pass

ML_tl_green/VAR

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

$$\begin{array}{lll}
d \in \mathbb{N} & d > 0 & \\
COLOUR = \{green, red\} & green \neq red & \\
n \in \mathbb{N} & n \leq d & \\
a \in \mathbb{N} & b \in \mathbb{N} & c \in \mathbb{N} \\
a + b + c = n & a = 0 \lor c = 0 & \\
ml_tl \in COLOUR & il_tl \in COLOUR & \\
ml_tl = green \Rightarrow a + b < d \land c = 0 & il_tl = green \Rightarrow b > 0 \land a = 0 & \\
ml_tl = red \lor il_tl = red & & \\
ml_pass \in \{0,1\} & il_pass \in \{0,1\} & \\
ml_tl = red \Rightarrow ml_pass = 1 & il_tl = red \Rightarrow il_pass = 1 & \\
ml_tl = red & a + b < d & c = 0 \\
il_pass = 1 & & \\
\vdash & & \\
0 + il_pass < ml_pass + il_pass & &
\end{array}$$

PO of Relative Deadlock Freedom

axm0_1	$d \in \mathbb{N}$
axm0_2	$d > 0$
axm2_1	$COLOUR = \{green, red\}$
axm2_2	$green \neq red$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
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	$a = 0 \lor c = 0$
inv2_1	$ml\_tl \in COLOUR$
inv2_2	$il\_tl \in COLOUR$
inv2_3	$ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4	$il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5	$ml\_tl = red \lor il\_tl = red$
inv2_6	$ml\_pass \in \{0, 1\}$
inv2_7	$il\_pass \in \{0, 1\}$
inv2_8	$ml\_tl = red \Rightarrow ml\_pass = 1$
inv2_9	$il\_tl = red \Rightarrow il\_pass = 1$

Disjunction of ***abstract*** guards

$a + b < d \land c = 0$	**guards of** *ML_out* **in** m_1
$\lor\ c > 0$	**guards of** *ML_in* **in** m_1
$\lor\ a > 0$	**guards of** *IL_in* **in** m_1
$\lor\ b > 0 \land a = 0$	**guards of** *IL_out* **in** m_1

$\vdash$

Disjunction of ***concrete*** guards

$ml\_tl = red \land a + b < d \land c = 0 \land il\_pass = 1$	**guards of** *ML_tl_green* **in** m_2
$\lor\ il\_tl = red \land b > 0 \land a = 0 \land ml\_pass = 1$	**guards of** *IL_tl_green* **in** m_2
$\lor\ ml\_tl = green \land a + b + 1 \neq d$	**guards of** *ML_out_1* **in** m_2
$\lor\ ml\_tl = green \land a + b + 1 = d$	**guards of** *ML_out_2* **in** m_2
$\lor\ il\_tl = green \land b \neq 1$	**guards of** *IL_out_1* **in** m_2
$\lor\ il\_tl = green \land b = 1$	**guards of** *IL_out_2* **in** m_2
$\lor\ a > 0$	**guards of** *ML_in* **in** m_2
$\lor\ c > 0$	**guards of** *IL_in* **in** m_2

Abstract m1

variables: a, b, c

```
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 m2

```
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 ≠ 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
```

Discharging POs of m2: Relative Deadlock Freedom

$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$ml\_tl = red \lor il\_tl = red$
$ml\_pass \in \{0, 1\}$
$il\_pass \in \{0, 1\}$
$ml\_tl = red \Rightarrow ml\_pass = 1$
$il\_tl = red \Rightarrow il\_pass = 1$
$\quad a + b < d \land c = 0$
$\lor \; c > 0$
$\lor \; a > 0$
$\lor \; b > 0 \land a = 0$
$\vdash$
$\quad ml\_tl = red \land a + b < d \land c = 0 \land il\_pass = 1$
$\lor \; il\_tl = red \land b > 0 \land a = 0 \land ml\_pass = 1$
$\lor \; ml\_tl = green$
$\lor \; il\_tl = green$
$\lor \; a > 0$
$\lor \; c > 0$

⋮

$d \in \mathbb{N}$
$d > 0$
$b \in \mathbb{N}$
$ml\_tl = red$
$il\_tl = red$
$ml\_tl = red \Rightarrow ml\_pass = 1$
$il\_tl = red \Rightarrow il\_pass = 1$
$\vdash$
$\quad b < d \land ml\_pass = 1 \land il\_pass = 1$
$\lor \; b > 0 \land ml\_pass = 1 \land il\_pass = 1$

…

$d \in \mathbb{N}$
$d > 0$
$b \in \mathbb{N}$
$ml\_tl = red$
$il\_tl = red$
$ml\_pass = 1$
$il\_pass = 1$
$\vdash$
$\quad b < d \land ml\_pass = 1 \land il\_pass = 1$
$\lor \; b > 0 \land ml\_pass = 1 \land il\_pass = 1$

…

$d > 0$, $b \in \mathbb{N}$ $\vdash$ $b < d \lor b > 0$

ARI

$d > 0$, $b > 0 \lor b = 0$ $\vdash$ $b < d \lor b > 0$

OR_L

- $d > 0$, $b > 0$ $\vdash$ $b < d \lor b > 0$ **OR_R2** $d > 0$, $b > 0$ $\vdash$ $b > 0$ **HYP**
- $d > 0$, $b = 0$ $\vdash$ $b < d \lor b > 0$ **EQ_LR, MON** $d > 0$ $\vdash$ $0 < d \lor 0 > 0$ **OR_R1** $d > 0$ $\vdash$ $0 < d$ **HYP**

1st Refinement and 2nd Refinement: Provably Correct

Abstract m1

variables: a, b, c

constants: d

axioms:
- **axm0_1** : $d \in \mathbb{N}$
- **axm0_2** : $d > 0$

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** : $a = 0 \lor c = 0$

variants: $2 \cdot a + b$

```
init
  begin
    a := 0
    b := 0
    c := 0
  end

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
```

Correctness Criteria:
- Guard Strengthening
- Invariant Establishment
- Invariant Preservation
- Convergence
- Relative Deadlock Freedom

Concrete m2

variables: a, b, c, ml_tl, il_tl, ml_pass, il_pass

constants: d

sets: $COLOR$

axioms:
- **axm0_1** : $d \in \mathbb{N}$
- **axm0_2** : $d > 0$
- **axm2_1** : $COLOR = \{green, red\}$
- **axm2_2** : $green \neq red$

invariants:
- **inv2_1** : $ml\_tl \in COLOUR$
- **inv2_2** : $il\_tl \in COLOUR$
- **inv2_3** : $ml\_tl = green \Rightarrow a + b < d \land c = 0$
- **inv2_4** : $il\_tl = green \Rightarrow b > 0 \land a = 0$
- **inv2_5** : $ml\_tl = red \lor il\_tl = red$
- **inv2_6** : $ml\_pass \in \{0, 1\}$
- **inv2_7** : $il\_pass \in \{0, 1\}$
- **inv2_8** : $ml\_tl = red \Rightarrow ml\_pass = 1$
- **inv2_9** : $il\_tl = red \Rightarrow il\_pass = 1$

variants: $ml\_pass + il\_pass$

```
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 ≠ d
  then
    a := a + 1
    ml_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_1
  when
    il_tl = green
    b ≠ 1
  then
    b := b − 1
    c := c + 1
    il_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

ML_in
  when
    c > 0
  then
    c := c − 1
  end

IL_in
  when
    a > 0
  then
    a := a − 1
    b := b + 1
  end
```