Lecture 19 - March 30

Reactive System: Bridge Controller

Announcements

- ProgTest1: Andy (eMail, Zoom); Jackie (Office Hour)
  - 2:30 - 3:30
  - 1:30pm - 3:30pm
- Lab3 due soon
- ProgTest2
  - guide

Lecture

Reactive System: Bridge Controller

First Refinement:
Relative Deadlock Freedom

Example Inference Rules

$$\frac{H, \neg P \vdash Q}{H \vdash P \vee Q} \quad \textbf{OR\_R}$$

Justify:

$H \Rightarrow P \vee Q \equiv H \wedge \neg P \Rightarrow Q$

$$\frac{H \vdash P}{H \vdash P \vee Q} \quad \text{OR\_R1}$$

$H \vdash P \vee Q$ ARI $H \vdash Q \vee P$ OR_R $H, \neg Q \vdash P$

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

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

Idea of Relative Deadlock Freedom

m_0 : DLF
m_1 : relative DLF

m_0 – m_1 – m_2

$A(c)$ — axioms
$I(c, v)$ — abs. I.
$J(c, v, w)$ — con. I.

$G_1(c, v) \vee \cdots \vee G_m(c, v)$ — disjunction of guards of all abstract events

↓ circumstance for abs. model not to deadlock

$\vdash \Rightarrow$

$H_1(c, w) \vee \cdots \vee H_n(c, w)$ — disjunction of guards of all con. events

DLF → relative DLF

Guard Strengthening

⋮
H_i
$\vdash$
G_i

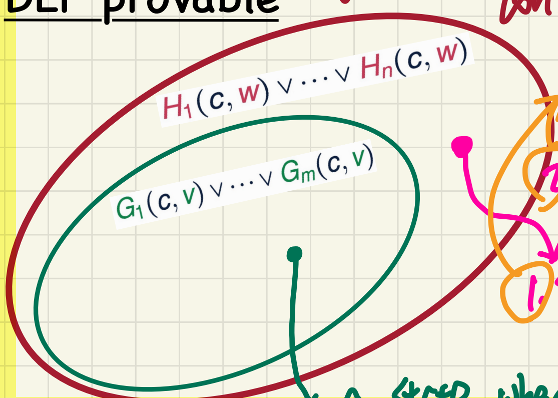
→ if concrete evt is enabled, then the abstract counterpart is enabled.

PRINCIPLES

1. DL is bad!
2. a refinement should not introduce a bad scenario

(unacceptable if there's a state where the con. model DL but the abs. model does not)

DLF provable

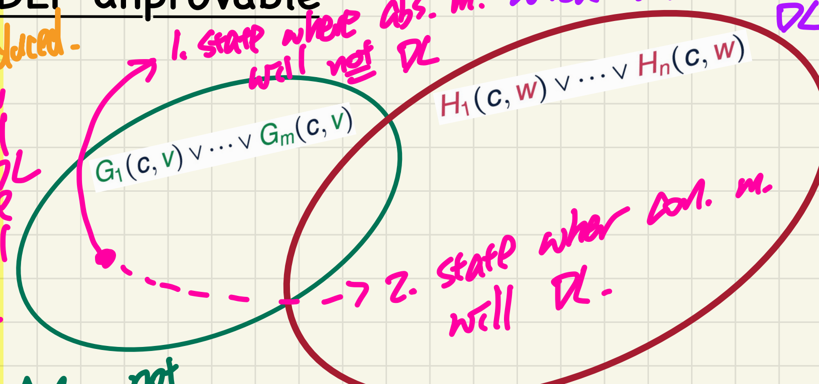


nothing bad introduced.

1. state where abs. model DL.
2. state where con. model not DL

→ a state where abs. & con. models will not DL.

DLF unprovable



1. state where abs. m. will not DL
2. state where con. m. will DL.

PO of Relative Deadlock Freedom

Abstract m0

variables: n

invariants:
- **inv0_1** : $n \in \mathbb{N}$
- **inv0_2** : $n \leq d$

```
ML_out
  when
    n < d
  then
    n := n + 1
  end
```

```
ML_in
  when
    n > 0
  then
    n := n − 1
  end
```

Concrete m1

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

```
① ML_out
  when
    a + b < d
    c = 0
  then
    a := a + 1
  end
```

```
② ML_in
  when
    c > 0
  then
    c := c − 1
  end
```

```
③ IL_in
  when
    a > 0
  then
    a := a − 1
    b := b + 1
  end
```

```
④ IL_out
  when
    b > 0
    a = 0
  then
    b := b − 1
    c := c + 1
  end
```

$$A(c)$$
$$I(c, v)$$
$$J(c, v, w)$$
$$G_1(c, v) \vee \cdots \vee G_m(c, v)$$
$$\vdash$$
$$H_1(c, w) \vee \cdots \vee H_n(c, w)$$

DLF

$d \in \mathbb{N}$
$d > 0$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \vee c = 0$

$n < d \vee n > 0 \vdash$

① $(a+b<d \wedge c=0)$
$\vee$
② $(c>0)$
$\vee$
③ $(a>0)$
$\vee$
④ $(b>0 \wedge a=0)$

Discharging POs of m1: Relative Deadlock Freedom

Part 1

$$\frac{H1 \vdash G}{H1, H2 \vdash G} \quad \textbf{MON}$$

$$\frac{H(F), E = F \vdash P(F)}{H(E), E = F \vdash P(E)} \quad \textbf{EQ\_LR}$$

$$\frac{H, \neg P \vdash Q}{H \vdash P \vee Q} \quad \textbf{OR\_R}$$

$d \in \mathbb{N}$
$d > 0$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \vee c = 0$
$n < d \vee n > 0$
$\vdash$
$\quad a + b < d \wedge c = 0$
$\vee \quad c > 0$
$\vee \quad a > 0$
$\vee \quad b > 0 \wedge a = 0$

$d > 0$
$b = 0 \vee b > 0$
$\vdash$
$\quad b < d \wedge 0 = 0$
$\vee \quad b > 0 \wedge 0 = 0$

Discharging POs of m1: Relative Deadlock Freedom

Part 2

$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, P \lor Q \vdash R} \quad \textbf{OR\_L}$$

$$\frac{H \vdash P}{H \vdash P \lor Q} \quad \textbf{OR\_R1}$$

$$\frac{}{P \vdash E = E} \quad \textbf{EQ}$$

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

$$\frac{H \vdash Q}{H \vdash P \lor Q} \quad \textbf{OR\_R2}$$

$$\frac{}{H, P \vdash P} \quad \textbf{HYP}$$

$$\begin{array}{l} d > 0 \\ b = 0 \lor b > 0 \\ \vdash \\ \qquad b < d \land 0 = 0 \\ \lor \quad b > 0 \land 0 = 0 \end{array}$$

Initial Model and 1st Refinement: Provably Correct

Abstract m0

constants: d

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

variables: n

invariants:
- inv0_1 : $n \in \mathbb{N}$
- inv0_2 : $n \leq d$

```
init
  begin
    n := 0
  end
```

```
ML_out
  when
    n < d
  then
    n := n + 1
  end
```

```
ML_in
  when
    n > 0
  then
    n := n − 1
  end
```

Concrete m1

constants: d

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

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 : $a = 0 \lor c = 0$

variants: $2 \cdot a + b$

measure interleaving of new events

- NAT
- VAR

```
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

Lecture

Reactive System: Bridge Controller

2nd Refinement: State and Events

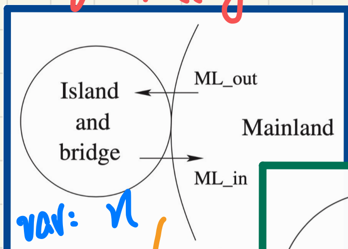
Bridge Controller: Abstraction in the 2nd Refinement

ENV1	The system is equipped with two traffic lights with two colors: green and red.
ENV2	The traffic lights control the entrance to the bridge at both ends of it.
ENV3	Cars are not supposed to pass on a red traffic light, only on a green one.

→ E-descriptions

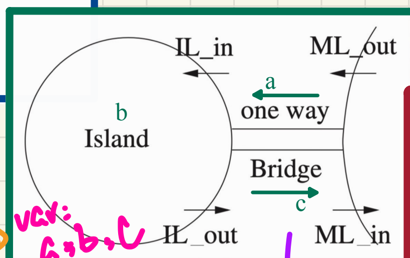
without this assumptions, m2 would have to be much more complicated (e.g. red-light camera)

m0: more abstract than m1



abs. vars replaced by con. vars

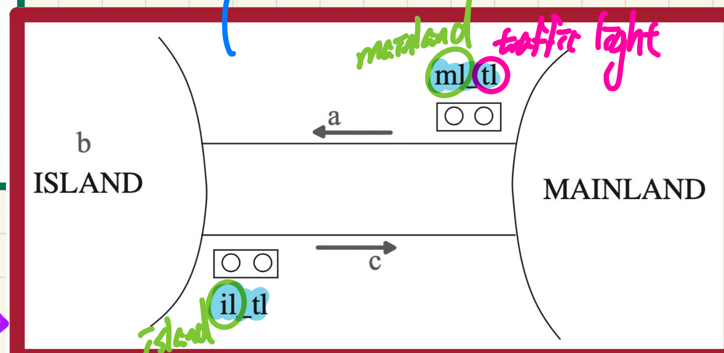
m1: more concrete than m0, more abstract than m2



superposition
1. abs. vars inherited
2. new con. variables

1. inv. est. & preservation
2. guard strengthening
3. relative DLF
4. convergence

m2: more concrete than m1



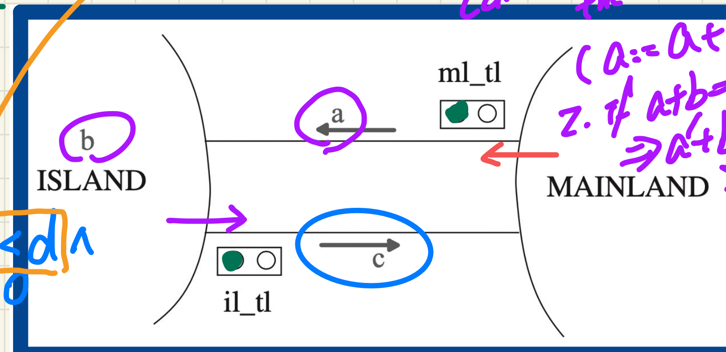
Bridge Controller: State Space of the 2nd Refinement

ENV1	The system is equipped with two traffic lights with two colors: green and red.
ENV2	The traffic lights control the entrance to the bridge at both ends of it.
ENV3	Cars are not supposed to pass on a red traffic light, only on a green one.

Dynamic Part of Model

variables:
a, b, c
ml_tl
il_tl

invariants:
inv2_1 : $ml\_tl \in COLOUR$
inv2_2 : $il\_tl \in COLOUR$
inv2_3 : ??
inv2_4 : ??



Static Part of Model

sets: $COLOR$

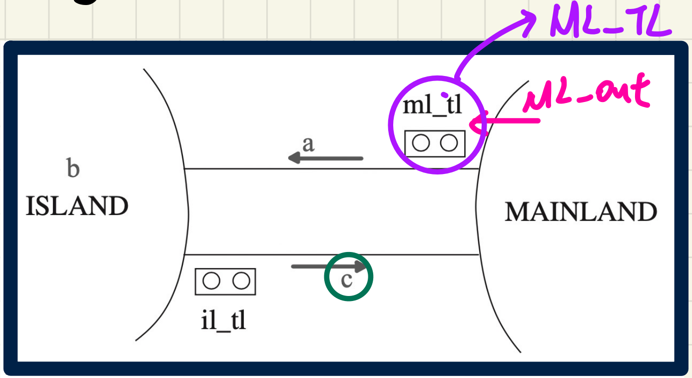
constants: $red, green$

axioms:
axm2_1 : $COLOR = \{green, red\}$
axm2_2 : $green \neq red$

Exercises

inv2_3: being allowed to exit ML means limited cars & no crash
inv2_4: being allowed to exit IL means some car in IL & no crash

Bridge Controller: Guards of "old" Events 2nd Refinement



sets:	COLOR

constants:	*red*, *green*

axioms:

axm2_1 : $COLOR = \{green, red\}$

axm2_2 : $green \neq red$

variables: a, b, c, ml_tl, il_tl

invariants:

inv2_1 : $ml\_tl \in COLOUR$

inv2_2 : $il\_tl \in COLOUR$

inv2_3 : $ml\_tl = green \Rightarrow a + b < d \land c = 0$

inv2_4 : $il\_tl = green \Rightarrow b > 0 \land a = 0$

ML_out: A car exits **mainland** (getting onto the **bridge**).

```
ML_out
  when
    ??
  then
    a := a + 1
  end
```

old event

for driver to follow

$ml\_tl = green$

abstract guard from ml: $c = 0 \land a + b < d$

guard for new event ML_TL

IL_out: A car exits **island** (getting onto the **bridge**).

```
IL_out
  when
    ??
  then
    b := b - 1
    c := c + 1
  end
```