Lecture 17 - Nov. 7

Bridge Controller

Interpreting Unprovable DLF PO
First Refinement: Abstraction, State Space

Announcements/Reminders

- **ProgTest2** results to be released by Monday, Nov 18
- **Lab5** to be released on Friday, Nov 15

L R

$n = d \vdash n < d \lor n > 0$

EQ_RL

replace in the goal every free occurrence of d by n.

$n = d \vdash n < d \lor n > 0$

Now

$\vdash F$

$\boxed{n < n} \lor n > 0$

$ARI_{\emptyset}$: $n < n \equiv F$

② $F \lor P \equiv P$

$\vdash \boxed{n > 0}$

unprovable

but may not be a good idea to fix the model accordingly

∵ n represents the # of cars on the bridge & island, which should be allowed to be zero when system is initialized.

Understanding the Failed Proof on DLF

constants: d

variables: n

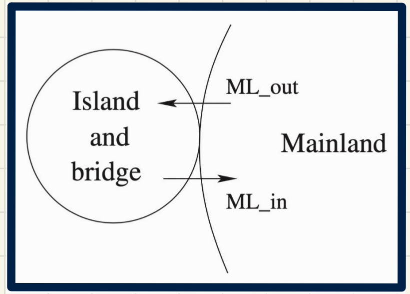
axioms:
axm0_1 : $\boxed{d \in \mathbb{N}}$

$d \in \mathbb{N}$, $d > 0$

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

ML_out
when
$\boxed{n < d}$
then
$n := n + 1$
end

ML_in
when
$\boxed{n > 0}$
then
$n := n - 1$
end



Unprovable Sequent: $\vdash d > 0$

$\boxed{\vdash d > 0}$

$\underbrace{0 < 0}_{F} \lor \underbrace{0 > 0}_{F}$

$\equiv \text{F}$ deadlocks right after init

Not being able to prove $d > 0$

↳ current model may violate it : $\neg(d > 0)$ is true.

① $d \leq 0$

② $d \in \mathbb{N}$ $(d \geq 0)$

$\boxed{d = 0}$ allowed by current model

Say $d = 0$

After init : $n = 0$

$G(\text{ML\_out}) \lor G(\text{ML\_in})$

$n < d$ (0 < 0) $\qquad$ $n > 0$ (0 > 0)

Discharging PO of DLF: Second Attempt

$$\begin{array}{l} d \in \mathbb{N} \quad d>0 \\ n \in \mathbb{N} \\ n \le d \\ \vdash \\ n < d \lor n > 0 \end{array}$$

new axiom added (as a fix).

$\equiv$

$$\begin{array}{l} d \in \mathbb{N} \quad d>0 \\ n \in \mathbb{N} \\ n < d \lor n = d \\ \vdash \\ n < d \lor n > 0 \end{array}$$

MON

$$\begin{array}{l} d>0 \\ n < d \lor n = d \\ \vdash \\ n < d \lor n > 0 \end{array}$$

OR_L

$$\begin{array}{l} d>0 \\ n < d \\ \vdash \\ n < d \lor n > 0 \end{array} \quad \textbf{OR\_R1} \quad \begin{array}{l} n < d \\ \vdash \\ n < d \end{array} \quad \textbf{HYP}$$

$$\begin{array}{l} d>0 \\ n = d \\ \vdash \\ n < d \lor n > 0 \end{array} \quad \textbf{EQ\_LR, MON} \quad \begin{array}{l} d>0 \\ \vdash \\ d < d \lor d > 0 \end{array} \quad \textbf{OR\_R2} \quad \boxed{\begin{array}{l} d>0 \\ \vdash \\ d > 0 \end{array}} \quad ?$$

HYP

Summary of the Initial Model: Provably Correct

static

constants: d

axioms:
axm0_1 : $d \in \mathbb{N}$
$\boxed{\textbf{axm0\_2} : d > 0}$

fix for the potential deadlock

dynamic

variables: n

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

init
  begin
    $n := 0$
  end

ML_out
  when
    $\boxed{n < d}$
  then
    $n := n + 1$
  end

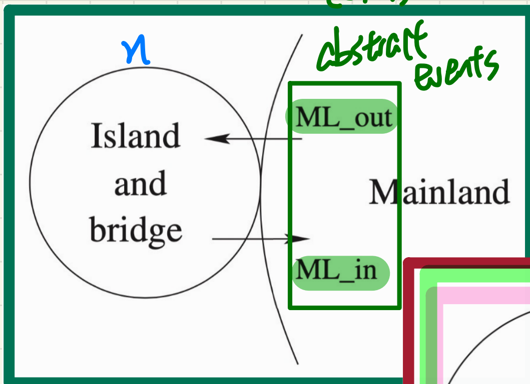
ML_in
  when
    $\boxed{n > 0}$
  then
    $n := n - 1$
  end

added for inv. pres.

Correctness Criteria:
+ Invariant Establishment
+ Invariant Preservation
+ Deadlock Freedom

Bridge Controller: Abstraction in the 1st Refinement

(old)
abstract events



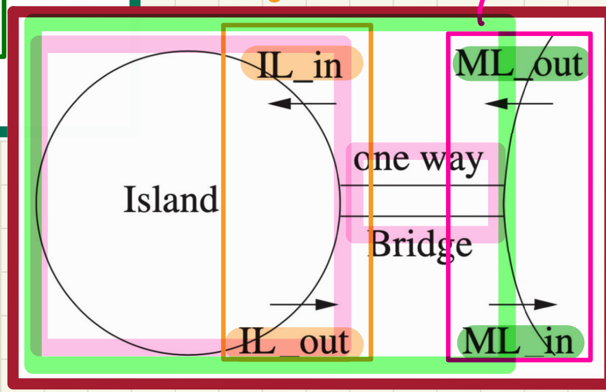
m0:

initial, most abstract

added "concreteness": one-way bridge

new events

old concrete events

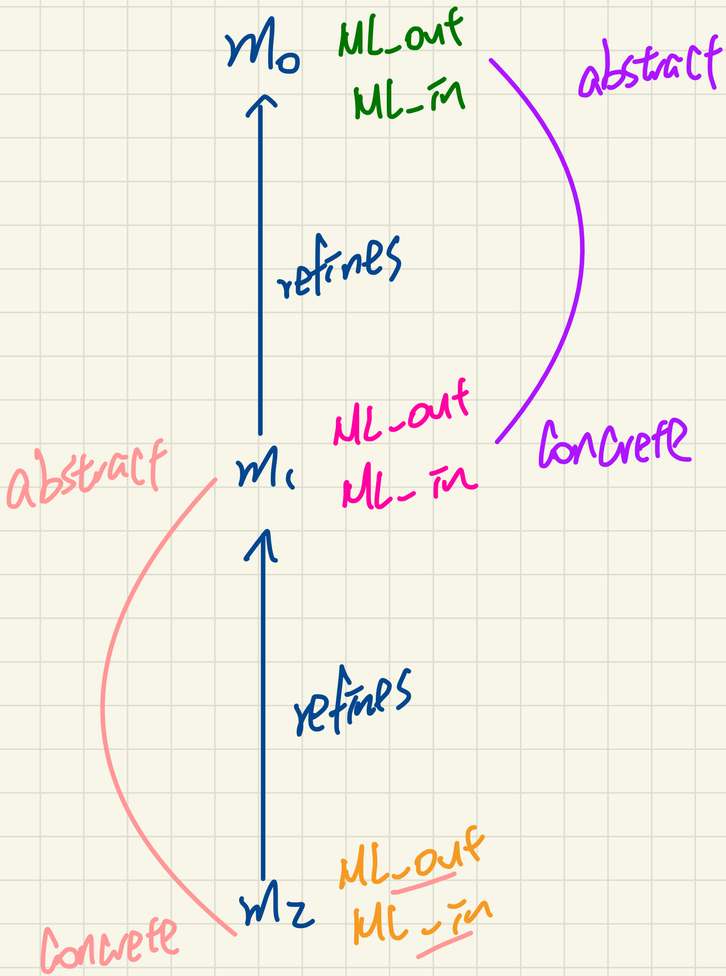


m1:

second, more concrete

traffic light still abstracted away.

REQ1	The system is controlling cars on a bridge connecting the mainland to an island.
REQ3	The bridge is one-way or the other, not both at the same time.


m0
ML_out
ML_in
abstract
refines
m1
ML_out
ML_in
abstract
concrete
refines
m2
ML_out
ML_in
concrete

$* \; a=0 \lor c=0 \equiv \neg(a \neq 0 \land c \neq 0)$

Bridge Controller: State Space of the 1st Refinement

REQ1	The system is controlling cars on a bridge connecting the mainland to an island.
REQ3	The bridge is one-way or the other, not both at the same time.

Dynamic Part of Model

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** : ??
- **inv1_5** : ??

linking/gluing invariant

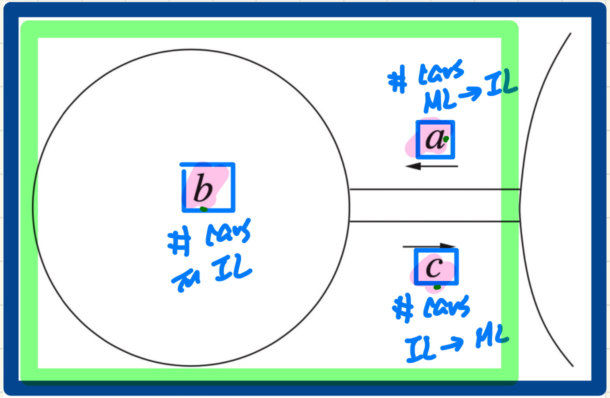
$n = a + b + c$

$a = 0 \lor c = 0$

df $a * c = 0$

may be hard for prover:

n



Static Part of Model

constants: d

axioms:

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

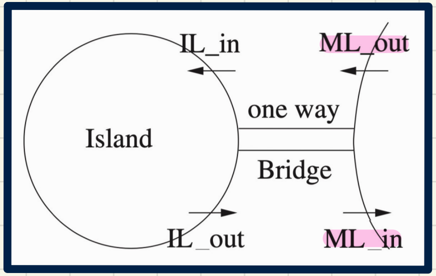
Exercises

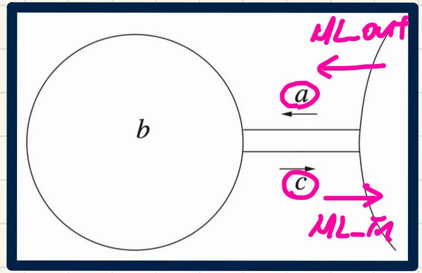
n a, b, c

inv1_4: linking abstract & concrete states

inv1_5: bridge is one-way

Bridge Controller: Guards of "old" Events 1st Refinement





constants: d

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

(2) $a+b<d$

** (1) (2) (3)

$n<d$

$a+b+\cancel{c}\,0<d$

variables: a, b, c

invariants:
- inv1_1 : $a \in \mathbb{N}$
- inv1_2 : $b \in \mathbb{N}$
- inv1_3 : $c \in \mathbb{N}$
- $\boxed{\text{inv1\_4} : a + b + c = n}$
- inv1_5 : $a = 0 \lor c = 0$

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

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

(1) $c=0$

abstract guard from M0 : $\boxed{n<d}$

(1) From M0, abstract guard of ML_out: $n<d$

(2) taking abst. & con. states: $a+b+c=n$

(3) first concrete grd: $c=0$

ML_in: A car enters **mainland** (getting off the **bridge**).

```
ML_in
  when
    ??
  then
    c := c - 1
  end
```

Before-After Predicates of Event Actions: 1st Refinement

Events	ML_in **when** $0 < c$ **then** $c := c - 1$ **end**	ML_out **when** $a + b < d$ $c = 0$ **then** $a := a + 1$ **end**
Before–after predicates	$a' = a \ \wedge \ b' = b \ \wedge$ $c' = c - 1$	$a' = a + 1 \ \wedge \ b' = b \ \wedge$ $c' = c$

- Pre-State
- Post-State
- Sate Transition