Lecture 19 - Nov 13

Bridge Controller

New Events: IL_in, IL_out
Simulation of New Events: skip
Livelock/Divergence: Example

Announcements/Reminders

- Today's class: notes template posted
- **Lab4** to be released

Events

n M_0

ML_out
ML_in

old events, abstract version

refines

a, b, c M_1

ML_out
ML_in

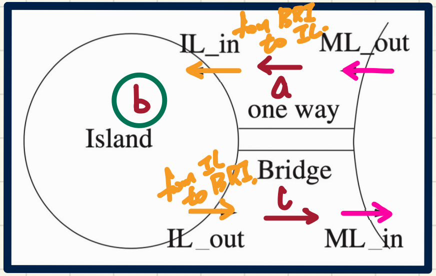
old events, concrete version

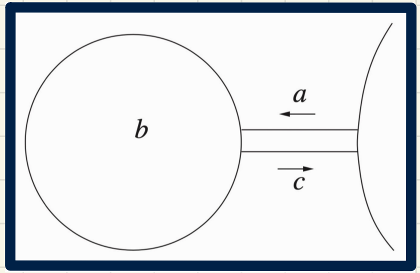
New events: IL_in, IL_out

Island.

* Each new event in a concrete model should be simulated by an event of no change in abstract model.

Bridge Controller: Guarded Actions of "new" Events in 1st Refinement





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 : $c \in \mathbb{N}$
- $\boxed{\text{inv1\_4} : a + b + c = n}$
- inv1_5 : $a = 0 \vee c = 0$

IL_in: A car enters island (getting off the bridge).

skip

```
IL_in
  when
    ??
  then
    ??
  end
```

① $a > 0$

② $\boxed{c = 0}$

$\boxed{\begin{array}{l} a := a - 1 \\ b := b + 1 \end{array}}$

necessary?

$\boxed{\begin{array}{l} a > 0 \\ a = 0 \vee c = 0 \\ \vdash c = 0 \end{array}}$

pre: $a + b + c = n$

post: $(a-1) + (b+1) + c = n$

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

as if nothing happened in M0.

```
IL_out
  when
    ??
  then
    ??
  end
```

① $b > 0$

② $\boxed{a = 0}$

$\boxed{\begin{array}{l} b := b - 1 \\ c := c + 1 \end{array}}$

$\boxed{\begin{array}{l} \text{all Inv} \\ b > 0 \\ \vdash a = 0 \end{array}}$

pre: $a + b + c = n$

post: $a + (b-1) + (c+1) = n$

Before-After Predicates of Event Actions: 1st Refinement

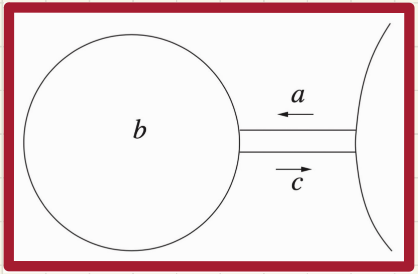
IL_in
  when
    $a > 0$
  then
    $\boxed{\begin{array}{l} a := a - 1 \\ b := b + 1 \end{array}}$
  end

$a' = a - 1$
$\wedge\ b' = b + 1$
$\wedge\ c' = c$

IL_out
  when
    $b > 0$
    $a = 0$
  then
    $b := b - 1$
    $c := c + 1$
  end

- Pre-State
- Post-State
- Sate Transition

Concrete State Space



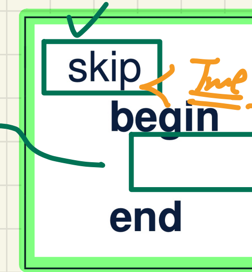
$\boxed{p \Rightarrow True \equiv True}$ guard strengthening trivially proved for new events.

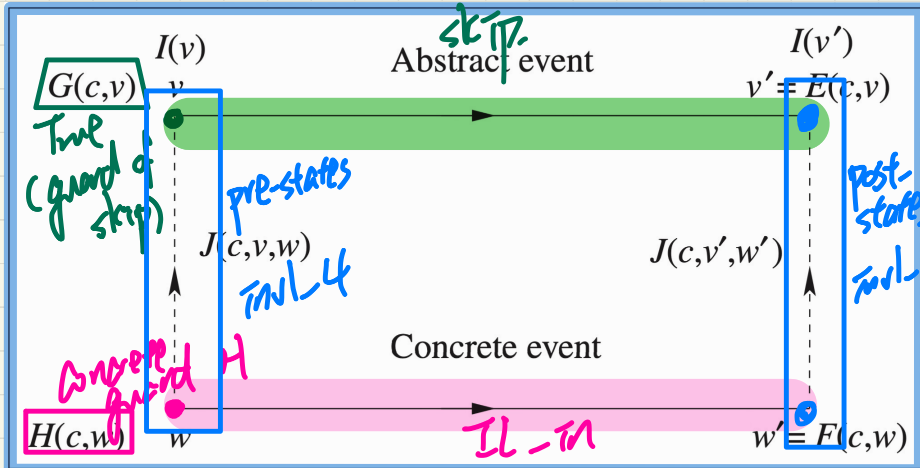
Visualizing Invariant Preservation in Refinement

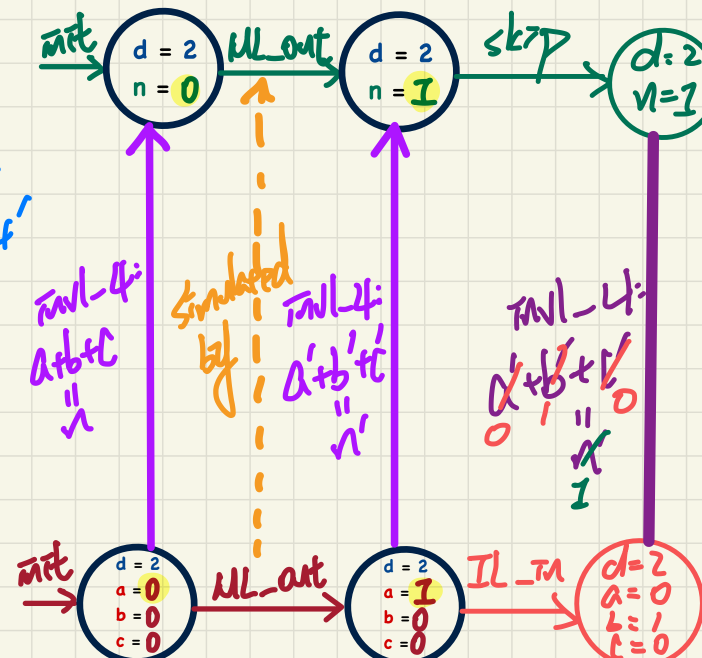
IL_in, IL_out

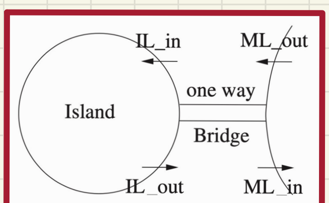
Each **new state transition** (from w to w′)
should be simulated by
an **abstract dummy state transition** (from v to v′)

skip $v' = v$









trace: ⟨init, ML_out, IL_in, ...⟩

old and new events

PO/VC Rule of Invariant Preservation: Sequents

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 : $n \leq d$

$A(c)$
$I(c, v)$
$J(c, v, w)$
$H(c, w)$
$\vdash$
$J_i(c, E(c, v))$ $F(c, w)$

effect on abstract model

each concrete inv. in post-state

effect on concrete model

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 : $a = 0 \vee c = 0$

IL_in
when
$a > 0$
then
$a := a - 1$
$b := b + 1$
end

corresponded by skip event in m0: $n' = n$

BAP: $a' = a - 1 \wedge b' = b + 1 \wedge c' = c$

$2 * 5 = 10$

IL_out
when
$b > 0$
$a = 0$
then
$b := b - 1$
$c := c + 1$
end

IL_in/INV1_4/INV

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

$(a-1)$ $(b+1)$ c n

$\vdash$ $\boxed{\cancel{a} + \cancel{b} + \cancel{c} = \cancel{n}}$

$(a-1) + (b+1) + c = n$

IL_in/INV1_5/INV

(exercise)

Q. How many PO/VC rules for model m1?

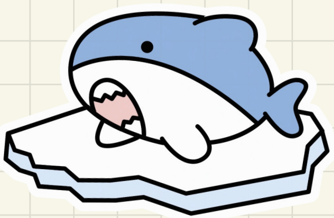
Discharging POs of m1: Invariant Preservation in Refinement

IL_in/inv1_4/INV

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

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

$d \in \mathbb{N}$
$d > 0$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$a > 0$
$\vdash$
$(a - 1) + (b + 1) + c = n$



Discharging POs of m1: Invariant Preservation in Refinement

ML_in/inv1_5/INV

$$\frac{}{\bot \vdash P}\ \textbf{FALSE\_L}$$

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

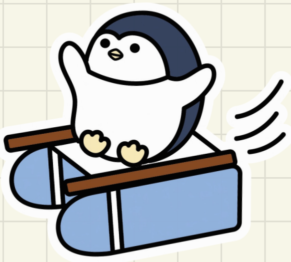
$$\frac{H \vdash Q}{H \vdash P \vee Q}\ \textbf{OR\_R2}$$

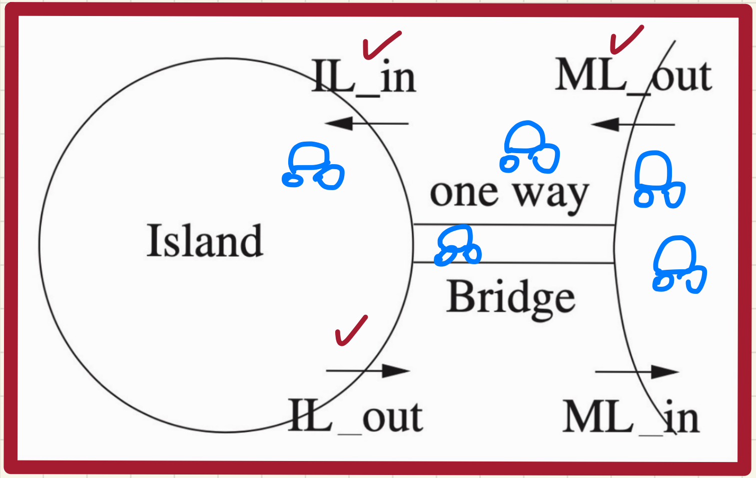
$$\frac{}{H, P \vdash P}\ \textbf{HYP}$$

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

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

$$\begin{array}{l} d \in \mathbb{N} \\ d > 0 \\ n \in \mathbb{N} \\ n \leq d \\ a \in \mathbb{N} \\ b \in \mathbb{N} \\ c \in \mathbb{N} \\ a + b + c = n \\ a = 0 \vee c = 0 \\ a > 0 \\ \vdash \\ (a - 1) = 0 \vee c = 0 \end{array}$$





Exercise Show the abstract & concrete transitions of:

abstract transitions: <init, ML_out, skip, skip, ML_in>

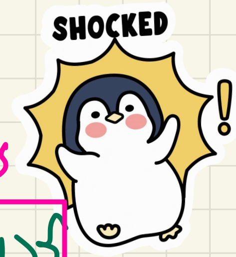
concrete transitions: <init, ML_out, IL_in, IL_out, ML_in>

Livelock Caused by New Events Diverging

- livelock
- divergence

sys. diverges

```
while(1){
}
```



An alternative m1 (for demonstration)

constants: d

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

variables: a, b, c

invariants:
inv1_1 : $a \in \mathbb{Z}$
inv1_2 : $b \in \mathbb{Z}$
inv1_3 : $c \in \mathbb{Z}$

old events

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

ML_in
when
$c > 0$
then
$c := c - 1$
end

new events (always enabled)

IL_in
begin
$a := a - 1$
$b := b + 1$
end

IL_out
begin
$b := b - 1$
$c := c + 1$
end

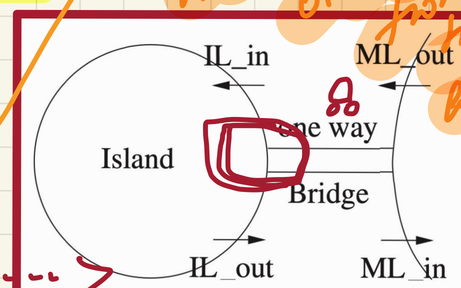
not contributing something useful, while preventing other events from happening.

A possible scenario that's problematic:

Abstract Transitions: ⟨ init, ML_out, skip* ⟩

Concrete Transitions: ⟨ init, ML_out, IL_in, IL_out, IL_in, IL_out, ... ⟩

infinite interleaving of new events



Livelock / Divergence

→ caused by an infinite interleaving of new events (≈ busy looping in the abstract model).

→ (system) variant ($\in \mathbb{N}$)

~ not a solution to livelock

~ a measure of the $\boxed{\text{\# of times}}$ new events might interleave

make sure it's not ∞

two POs to discharge

→ if unprovable
↳ fix model