# Lecture 17 - March 21

## Reactive System: Bridge Controller

# Announcements

- Lab3 ✓ released
- Review Q&A Session 7pm on Wednesday, March 22

  Zoom

# Lecture

## Reactive System: Bridge Controller

### *First Refinement: Inv. Establishment*

# PO of Invariant Establishment in Refinement

**constants:** $d$

**variables:** $a, b, c$

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

init
**begin**
$a := 0$
$b := 0$
$c := 0$
**end**

$a' = 0 \quad b' = 0 \quad c' = 0$

## Components

K(c): effect of abstract init

L(c): effect of concrete init

## Rule of Invariant Establishment

$$\frac{A(c) \quad \vdash \quad \cancel{J(c, v, w)}}{J_i(c, K(c), L(c))}$$

× ∵ no pre-state
✓ when init. the system

abs. init.    con. init. steps.

## Exercise:

Generate Sequents from the INV rule.

init/inv1_4/INV

$d \in \mathbb{N}$
$d > 0$
$\vdash$
$0 + 0 + 0 = 0$

$* \quad a' + b' + c' = n$

$0 + 0 + 0 = 0$

exercise:

init/inv1_5/INV : formulate + prove

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

↳ 5 (init, 5 inv.)

ARI : simplification

# Discharging PO of Invariant Establishment in Refinement

$$d \in \mathbb{N}$$
$$d > 0$$
$$\vdash$$
$$0 + 0 + 0 = 0$$

init/inv1_4/INV

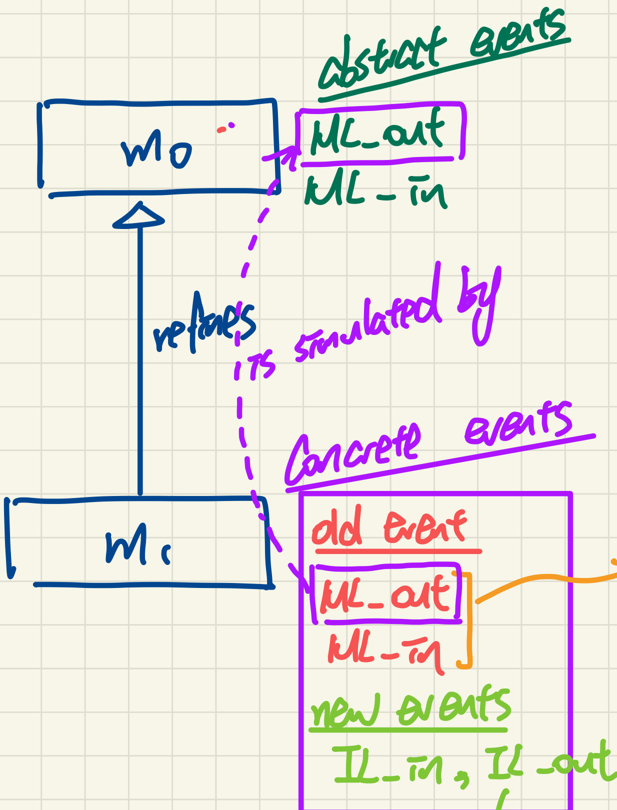
$$\frac{H1 \vdash G}{H1, H2 \vdash G} \quad \textbf{MON}$$

$$\frac{}{P \vdash \top} \quad \textbf{TRUE\_R}$$

$$d \in \mathbb{N}$$
$$d > 0$$
$$\vdash$$
$$0 = 0 \lor 0 = 0$$

init/inv1_5/INV

# Events




1. guard strengthening
2. inv. pre.
3. inv. est.

(3) divergence
(4) variant of model

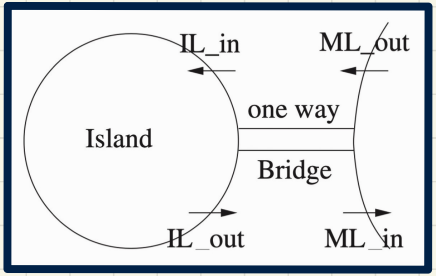
(1) guards, actions
(2) IL_in is simulated by which abstract evt. ?

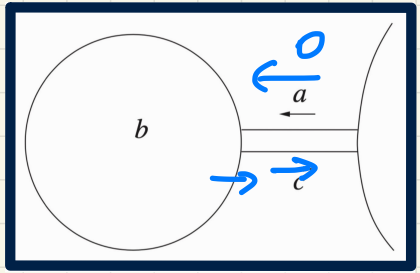
# Lecture

## Reactive System: Bridge Controller

### *First Refinement: Invariant Preservation*
### *New Events*

# 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 : $c \in \mathbb{N}$
inv1_4 : $a + b + c = n$
inv1_5 : $a = 0 \lor c = 0$

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

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

$a > 0$

$c = 0$ not necessary ∵ inv1_5

Q. Guard: $a + b < d$

not necessary: ① ML_out already checks it ② $a' + b' = (a-1) + (b+1) = a + b$

$a := a - 1$
$b := b + 1$

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

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

$b > 0$
$a = 0$

$b := b - 1$
$c := c + 1$

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

# Before-After Predicates of Event Actions: 1st Refinement

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

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

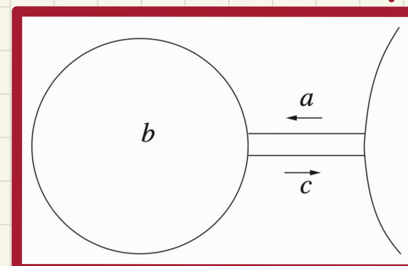
```
IL_out
  when
    b > 0
    a = 0
  then
    b := b − 1
    c := c + 1
  end
```

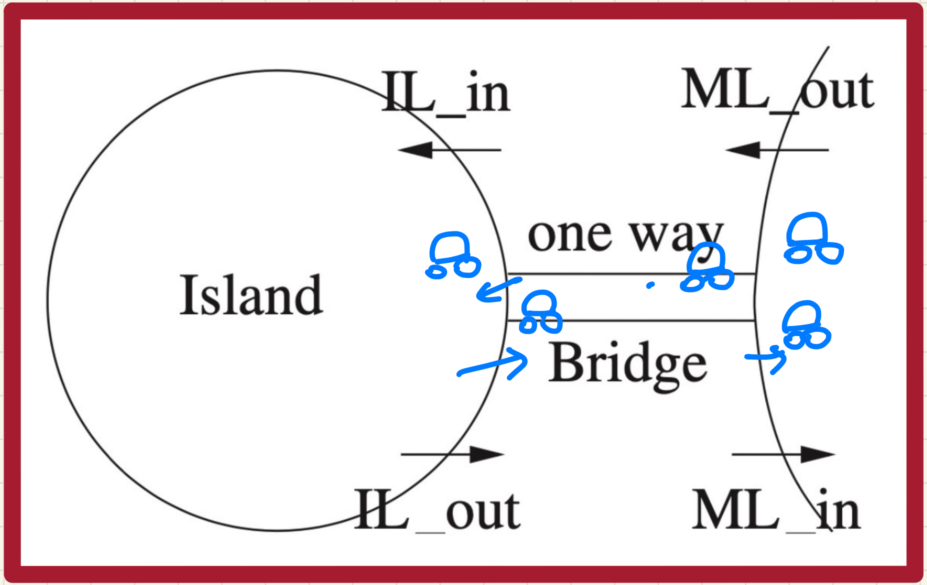
BAP: $b' = b - 1 \land c' = c + 1 \land a' = a$

$a' + b' + c' = a + b + c = n$

- Pre-State
- Post-State
- Sate Transition

## Concrete State Space






Trace: 1 car travelling

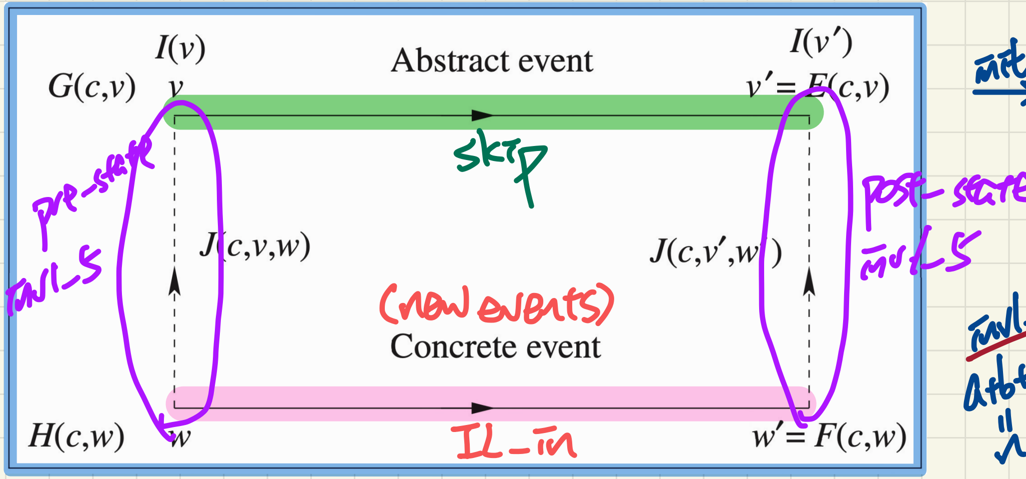
⟨init, ML_out, IL_in, IL_out, ML_in⟩

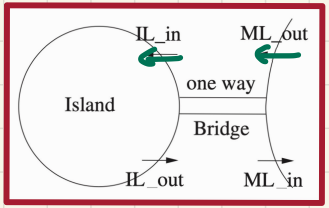
Exercise 2 cars travelling

# Visualizing Invariant Preservation in Refinement

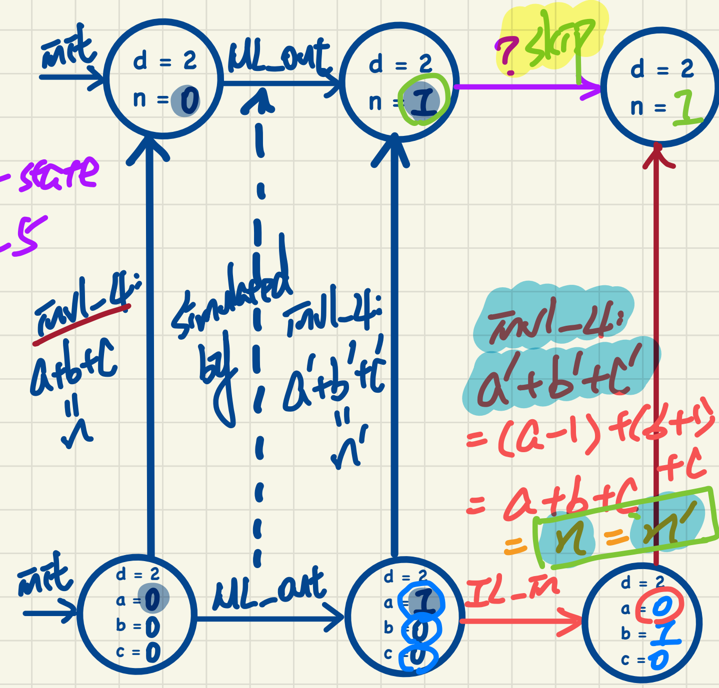
Each **new state transition** (from w to w′) should be simulated by an **abstract dummy state transition** (from v to v′)

```
skip
  begin
    (n′ = n)
  end
```




inv_4: $a+b+c=n$



# PO/VC Rule of Invariant Preservation: Sequents

## Abstract m0

| constants: $d$ | variables: $n$ |
|---|---|
| axioms: <br> axm0_1 : $d \in \mathbb{N}$ <br> axm0_2 : $d > 0$ | invariants: <br> inv0_1 : $n \in \mathbb{N}$ <br> inv0_2 : $n \le d$ |

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

## 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
```

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

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

IL_in/INV1_4/INV

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

$\vdash (a-1)+(b+1)+c = n$ ?

? $a'+b'+c' = n'$

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

IL_in/INV1_5/INV

(Exercise: formulate & prove).

skip event

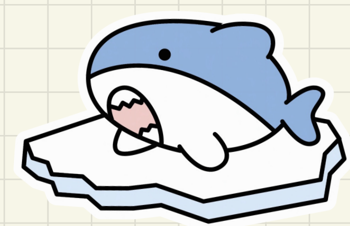
# Discharging POs of m1: Invariant Preservation in Refinement

**IL_in/inv1_4/INV**

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

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

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



# Discharging POs of m1: Invariant Preservation in Refinement

**ML_in/inv1_5/INV**

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

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

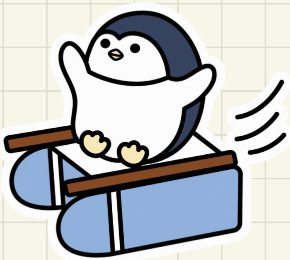
$$\frac{H \vdash Q}{H \vdash P \lor Q}\ \textbf{OR\_R2}$$

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

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

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

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



# Lecture

## Reactive System: Bridge Controller

***First Refinement: Convergence***
***New Events***

# Livelock Caused by New Events Diverging

## 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** : $c \in \mathbb{Z}$

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

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

while(true){
waiting
}

system is under-specified:
(1) safety properties are missing (e.g. $a = 0 \vee c = 0$)
(2) linking invariants are missing (e.g. $a + b + c = n$)

Abstract Transitions: ⟨init, ML_out, skip, skip, skip, skip, ...⟩

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

but since invariants are incomplete, the notion of correctness is weak.

**Exercise**

POs related to INV preservation can be discharged. How?