Lecture 2

Part D

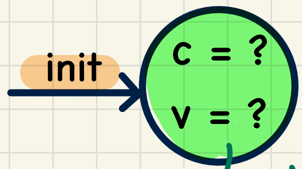
Case Study on Reactive Systems -
Bridge Controller
Initial Model: Invariant Establishment

Initializing the System → ASM

Analogy to Induction:

Base Cases ≈ Establishing Invariants

→ P(1)
P(2)
⋮

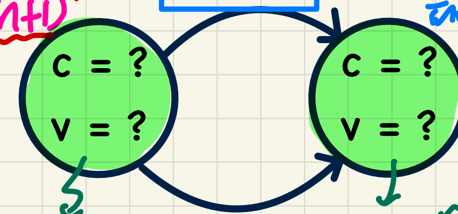


Initial state when the system is first launched.

Analogy to Induction:

Inductive Cases ≈ Preserving Invariants

invariant
P(n) ⇒ P(n+1)
pre-state
post-state



→ ML_out/
INV0_1/INV

assume (pre-state)

prove (post-state)

The Initialization Event

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

variable

no notion of pre-state value

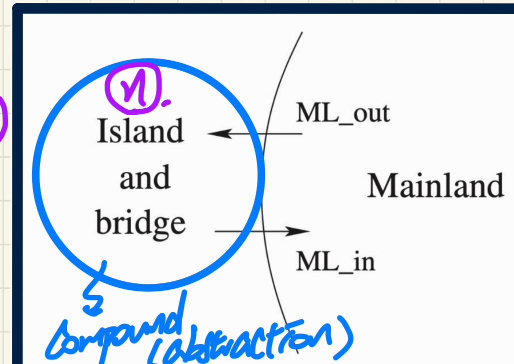
n := n+1 ✗

value in post-state

PRINCIPLES

BAP: n′ = 0

1. init has no guards (unconditional)
   (no pre-state constraints)
2. only use constants to specify the post-state value



compound (abstraction)

PO of Invariant Establishment

m0

constants: d

variables: n

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

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

init
  begin
    $n := 0$
  end

BAP: $n' = 0$

Components

$K(c)$: effect of init's actions

constants

specified w.r.t. constants or literals.

$v' = K(c)$: BAP of init's actions

only the notion of post-state is applicable.

Rule of Invariant Establishment

$$\frac{A(c) \vdash I_i(c, \boxed{K(c)})}{}\ \text{INV}$$

invariant satisfied at the pre-state is not relevant here.

single invariant condition.

post-state values of variables w.r.t. init's actions.

Exercise:

Generate Sequents from the INV rule.

init/inv0_1/INV

$$d \in \mathbb{N} \vdash \cancel{n'} \in \mathbb{N}$$
$$0$$

init/inv0_2/INV

$$d \in \mathbb{N} \vdash \cancel{n'} \leq d$$
$$0$$

Discharging PO of Invariant Establishment

$$d \in \mathbb{N} \vdash 0 \in \mathbb{N}$$

init/**inv0_1**/INV MON $\boxed{\vdash 0 \in \mathbb{N}}$ ✓ P1

$$d \in \mathbb{N} \vdash 0 \leq d$$

init/**inv0_2**/INV P3 ✓

$$\frac{}{\vdash 0 \in \mathbb{N}} \quad \textbf{P1}$$

$$\frac{}{n \in \mathbb{N} \vdash 0 \leq n} \quad \textbf{P3}$$

d instantiates n

Lecture 2

Part E

Case Study on Reactive Systems -
Bridge Controller
Initial Model: Deadlock Freedom

PO Rule: Deadlock Freedom

init
not relevant.

REQ4	Once started, the system should work for ever.

m0

constants: d

variables: n

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

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

H

$$\boxed{\begin{array}{l} A(c) \\ I(c, v) \end{array}} \vdash G_1(c, v) \vee \cdots \vee G_m(c, v) \quad \text{DLF}$$

pre-state values

- c: list of ***constants*** $\langle d \rangle$
- $A(c)$: list of ***axioms*** $\langle$**axm0_1**$\rangle$
- v and v': list of ***variables*** in ***pre***- and ***post***-states $v \mathrel{\widehat{=}} \langle n \rangle$, $v' \mathrel{\widehat{=}} \langle n' \rangle$
- $I(c, v)$: list of ***invariants*** $\langle$**inv0_1**, **inv0_2**$\rangle$
- $G(c, v)$: the event's ***guard***

$G(\langle d \rangle, \langle n \rangle)$ of $ML\_out \mathrel{\widehat{=}} n < d$, $G(\langle d \rangle, \langle n \rangle)$ of $ML\_in \mathrel{\widehat{=}} n > 0$

② Instead, we're concerned about if there's even a transition in the first place.

Exercise: Generate Sequent from the DLF rule.

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$\vdash$
$n < d \vee n > 0$

pre-state values

2. before-after pred. of event actions irrelevant

∵ ① we're not concerned about effects of event actions

PO	pre-state	post-state
INV est.	n.a.	✓
INV pre.	✓	✓
DLF	✓	n.a.

Example Inference Rules

To prove the consequent, it's sufficient to prove nothing. (i.e. consequent proved auto.)

$\perp$ T

HYP

$$\frac{\boxed{}}{H, P \vdash P}$$

$H \wedge P \Rightarrow P$

axiom IRs

theorem without further justification

FALSE_L

$$\frac{}{\boxed{\perp \vdash P}}$$

false, "bottom"

$\perp \Rightarrow P \equiv T$ (zero of $\Rightarrow$)

TRUE_R

$$\frac{}{P \vdash \top}$$

true, "top"

$P \Rightarrow T \equiv T$ (zero of $\Rightarrow$)

EQ

$$\frac{}{P \vdash \boxed{E = E}}$$

T

EQ_LR

$$\frac{H(F), E = F \vdash P(F)}{H(E), \boxed{E = F} \vdash P(E)} \quad E=F$$

hypothesis: E and F are interchangeable → replace occurrence of L by R

from left to right

EQ_RL

$$\frac{H(E), \boxed{E = F} \vdash P(E)}{H(F), \boxed{E = F} \vdash P(F)} \quad E=F$$

from R to L

replace F by E

Discharging PO of DLF: First Attempt

* $d > 0 \rightarrow$ max # cars ≥ 1

* $n > 0 \rightarrow$ # cars ≥ 1

max $= 0$ should be avoided

not reasonable to impose on model

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

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

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

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

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

Mo may not be sufficient

$\cancel{d \in \mathbb{N}}$
$n \in \mathbb{N}$ $\quad n \geq 0$
$n \leq d$ $\quad$ upper bound of n
$\vdash$
$n < d \vee n > 0$

guard of ML_out $\qquad$ guard of ML_in

$\equiv$
$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n < d \vee n = d$
$\vdash$
$n < d \vee n > 0$

MON

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

OR_L

$n < d \vdash n < d \vee n > 0$ $\quad$ ~~MON~~ OR_R1 $\quad$ $n < d \vdash n < d$ $\quad$ HYP

✓ $n = d \vdash n < d \vee n > 0$ $\quad$ EQ_LR $\quad$ $n = d \vdash d < d \vee d > 0$ $\quad$ MON $\quad$ $\vdash d < d \vee d > 0$ * ✓ ?

alternatively

EQ_RL $\quad$ $n = d \vdash n < n \vee n > 0$ $\quad$ MON $\quad$ $\vdash n < n \vee$ * $n > 0$ X

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

Understanding the Failed Proof on DLF

① $d=0$: max 0 cars on the IB comp.

② $n=0$ by init

constants: d

variables: n

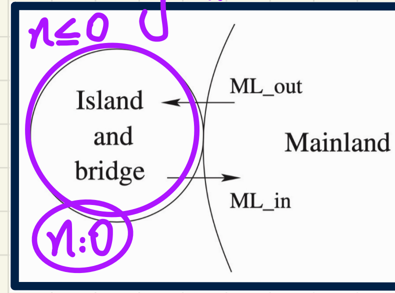
axioms:
- axm0_1 : $d \in \mathbb{N}$ ($d \geq 0$)
- axm0_2 : $d > 0$

↳ revision on model based on

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



Unprovable Sequent: $\boxed{\vdash d > 0}$ ✓

$\neg(d > 0)$ is possible for m_0

① $d \leq 0$

② axiom0_1: $d \in \mathbb{N}$ ($d \geq 0$)

↳ $\boxed{d = 0}$ (counter scenario for deadlock freedom)

$d=0$: deadlock happens

Init: $n'=0$

$\boxed{\cancel{n}_{0} < \cancel{d}_{0} \lor \cancel{n}_{0} > 0}$ → false $\perp$

both events are disabled

↳ deadlock!!

Discharging PO of DLF: Second Attempt

added axiom:
axm0_2: $d > 0$

PO of DLF

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

$\equiv$

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

$$\overline{H, P \vdash P} \quad \text{HYP}$$

MON

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

OR_L

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

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

drops: $n = d$

not yet ready to be applied HYP rule!

Summary of the Initial Model: Provably Correct

invariant establishment

invariant preservation

constants: d

variables: n

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

deadlock freedom (non-blocking property).

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
```

Correctness Criteria:

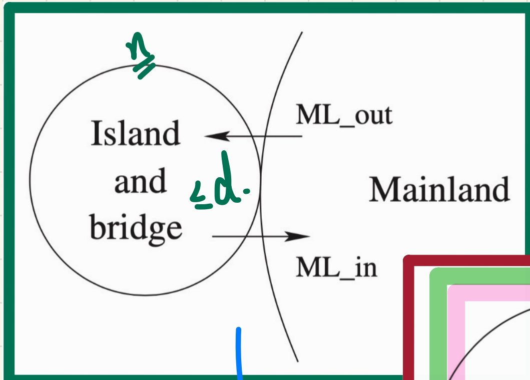
+ Invariant Establishment
+ Invariant Preservation
+ Deadlock Freedom

Lecture 2

Part F

Case Study on Reactive Systems -
Bridge Controller
First Refinement: State and Events

Bridge Controller: Abstraction in the 1st Refinement



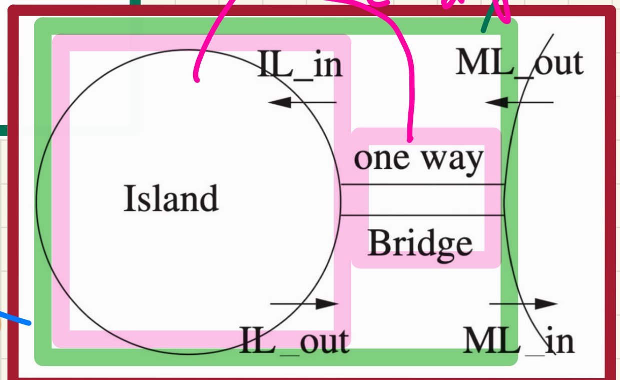
m0:

initial, most abstract

m1 abs. more concrete than m0 abs.

m1: abstraction of 1st refinement (island vs. bridge)

abstraction of initial model m0 (IB compound)



m1:

second, more concrete

m0 state space: abstract state

m1 state space: concrete state

① both models are specifying the same system with diff. levels of details

② these two levels of details must be proved consistent

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.