EECS3342 System Specification and Refinement

Lecture Notes

Winter 2022

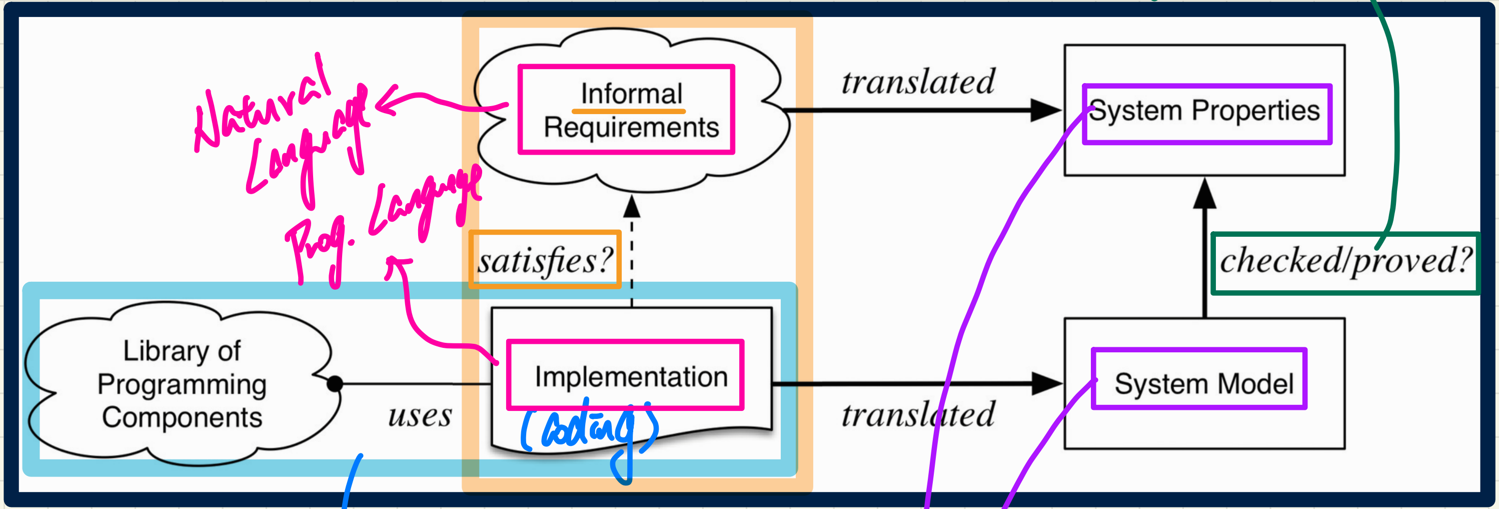
Jackie Wang

Lecture 1a

Introduction

Building the product right?

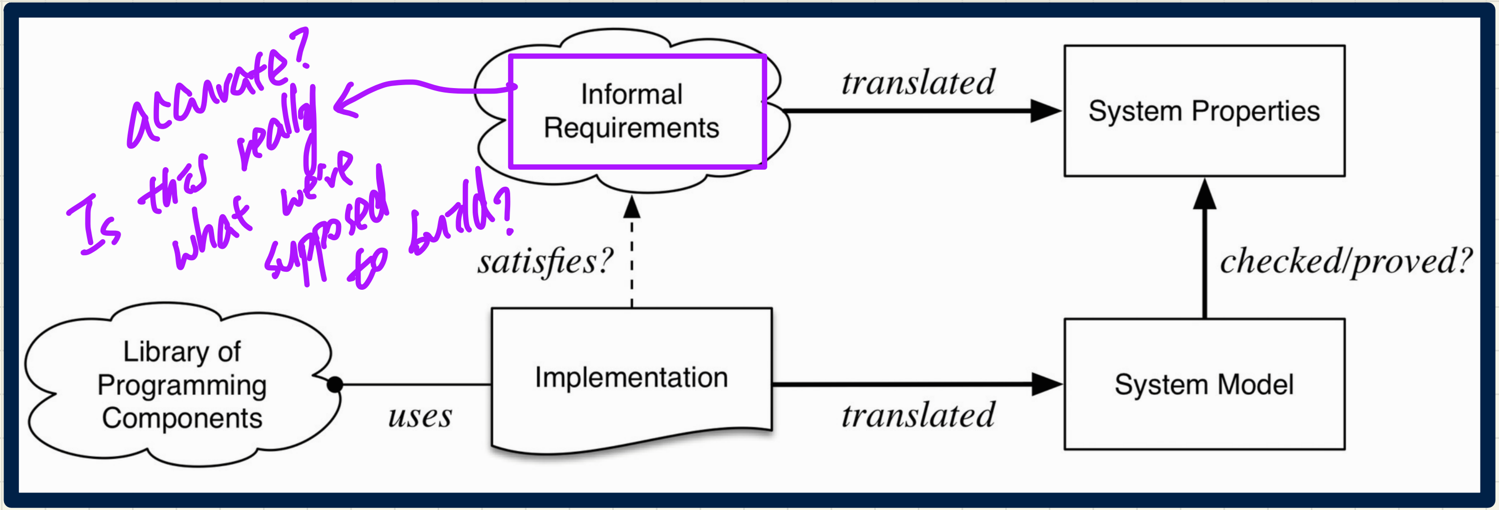
success means the right product to build? Not necessarily.



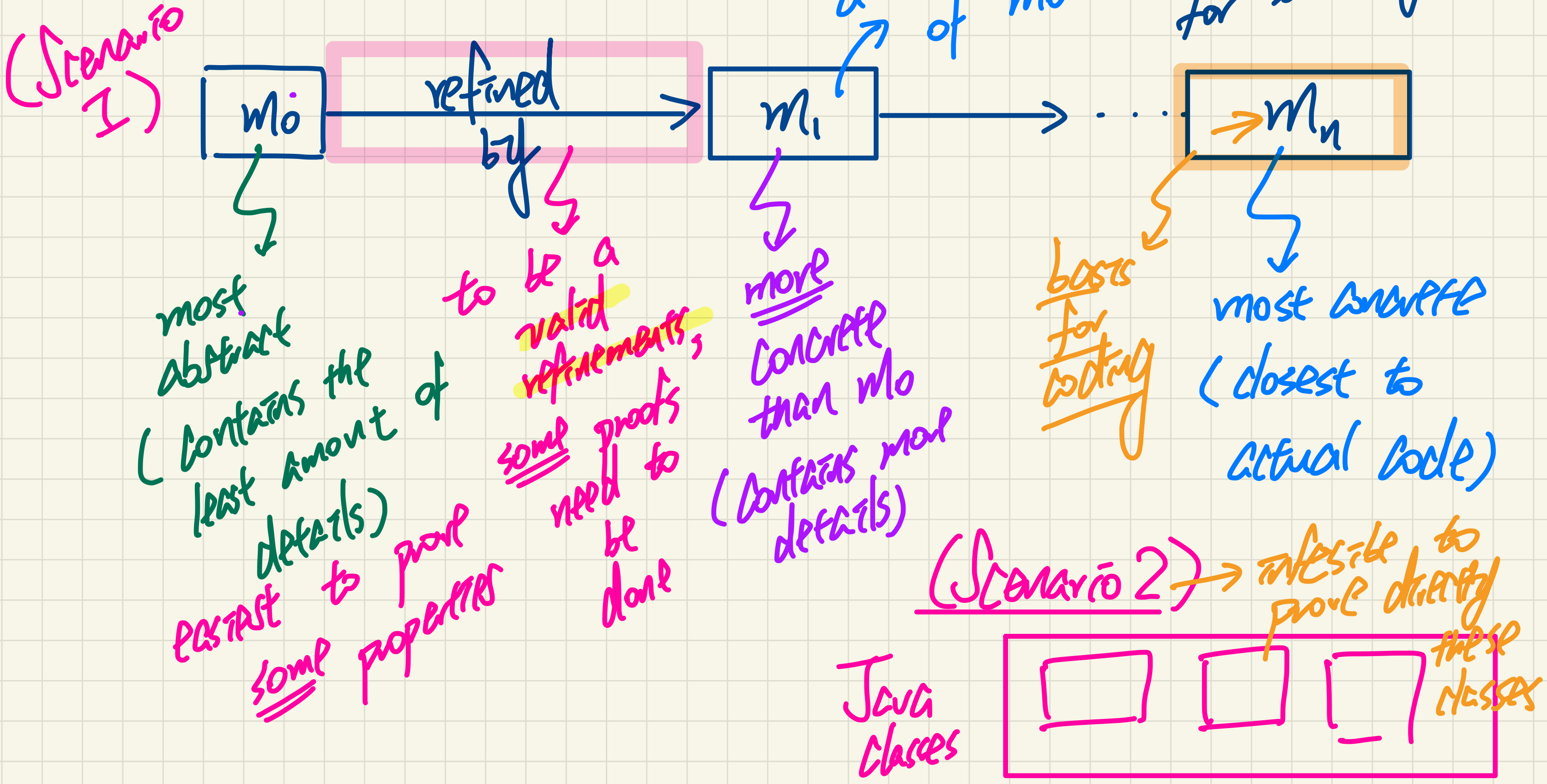
e.g. using Java API

specified using the same formal language.

Building the right product?



Model - Based Development



a refinement of m_0

(n+1) models for same system.

m_0: most abstract (contains the least amount of details)

easiest to prove some properties

refined by: to be a valid refinements, some proofs need to be done

m_1: more concrete than m_0 (contains more details)

m_n: most concrete (closest to actual code)

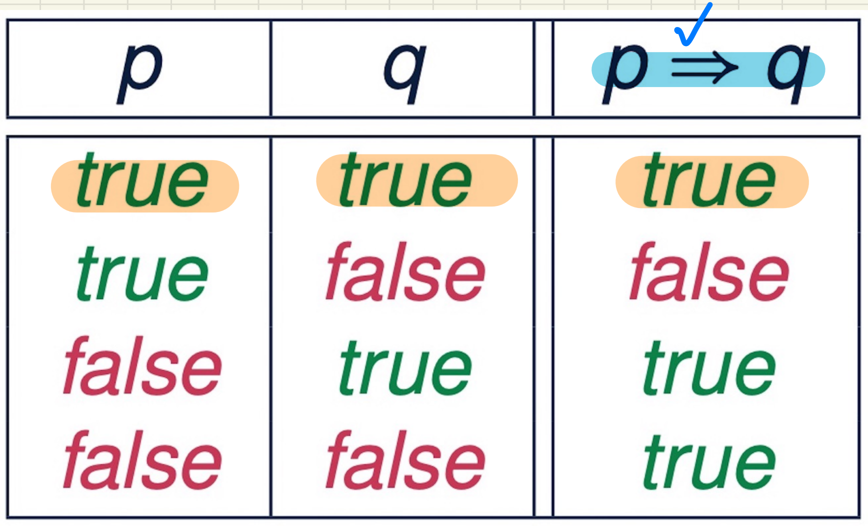
basis for coding

(Scenario 2) → infeasible to prove directly these classes

Java classes

Lecture 1b

Review on Math



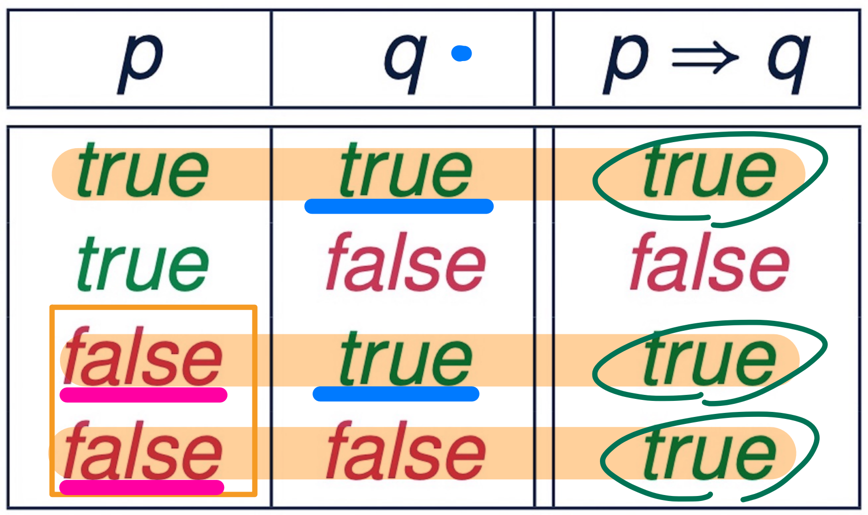
p	q	$p \overset{\checkmark}{\Rightarrow} q$
true	true	true
true	false	false
false	true	true
false	false	true

p only if q

↳ p holds, then the only way for ⇒ to hold is if q holds

q is necessary for p

↳ p holds, then it's necessary for q to hold s.t. ⇒ holds.



p	q	$p \Rightarrow q$
true	true	true
true	false	false
false	true	true
false	false	true

When is $p \Rightarrow q$ true?

1. both p and q hold

2. p does not hold

q unless $\neg p$

$$p \Rightarrow q \equiv \neg p \vee q$$

$\forall$ universal quantification ("for all")

$\exists$ existential quantification ("there exists")

$$\exists i, j \bullet \underline{(i \in \mathbb{Z} \wedge j \in \mathbb{Z} \wedge i < j)} \underline{\vee} \underline{i > j}$$

how the predicate is evaluated ∵ ∧ binds more tightly than ∨

Precedence of Logical Ops.

$\neg$

$\wedge$

$\vee$

$\Leftrightarrow \quad \equiv$

$$\exists i, j \bullet \underbrace{\boxed{i \in \mathbb{Z} \wedge j \in \mathbb{Z}}}_{R} \wedge \underbrace{\boxed{(i < j \vee i > j)}}_{P}$$

Conversions between ∀ and ∃

1. $(\forall i \cdot i \in S \Rightarrow i > 0) \Leftrightarrow \neg(\exists i \cdot i \in S \wedge \neg(i > 0))$

2. $(\exists \underline{\underline{i}} \cdot i \in S \wedge i > 0) \Leftrightarrow \neg(\forall i \cdot i \in S \Rightarrow \neg(i > 0))$

$\in$ membership

$e \notin S \equiv \neg(e \in S)$

$S = \{1, 2, 3\}$

$T = \{2, 3, 1\}$

$U = \{3, 2\}$

$(i \le j \wedge j \le i) \iff i = j$

$(S \subseteq T \wedge T \subseteq S) \iff S = T$

$S \subseteq T$ ✓ $S \subset S$ ✗

$T \subseteq S$ ✓ $S \subset T$ ✗

$U \subseteq S$ ✓ $S \subset U$ ✗

$U \subseteq T$ ✓

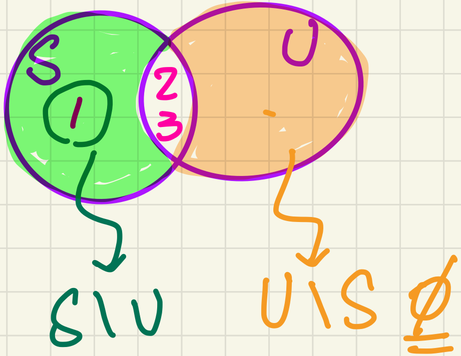
$U \subset S$ ✓

$U \subset T$ ✓

not commutative

$S \boxed{\setminus} U = \{1\}$

$U \setminus S = \varnothing$



Power Set

$$\binom{3}{1} = \underline{\underline{3}}$$

$$\binom{3}{2} = \binom{3}{1} = 3$$

$$\mathbb{P}(\underline{\{\dot{1}, \dot{2}, \dot{3}\}})$$

$$= \{ s \mid s \subseteq \{1, 2, 3\} \}$$

$$= \{ \underline{\emptyset}\ 0,\ \boxed{\{1\},\ \{2\},\ \{3\}},\ \boxed{\{1,2\},\ \{2,3\},\ \{1,3\}},\ \underline{\{1,2,3\}}\ 3 \}$$

subsets of card. 1

subsets of card. 2

int i ;

i stores a single integer

type: set of 2^{32} values

:

$$\boxed{i \in int}$$

$$r \overset{\in}{:} \mathbb{P}(\boxed{S \times T})$$

declared to store the value of a single relation, which is a subset of S x T

each member is a subset of S x T (which will be a relation smaller than or equal to S x T)

max relation on S and T

$$r : \mathbb{P}(S \times T)$$

$$\equiv$$

$$r : S \leftrightarrow T$$

$$\boxed{\text{Enumerate : } \{a, b\} \Longleftrightarrow \{1, 2, 3\}}$$

$$\mathbb{P}(\{a, b\} \times \{1, 2, 3\})$$

relations of card. 2

$$\binom{6}{2} = \frac{6 * 5}{2!} = 15.$$

card. of max relation

relation of card. 0

relations of card. 1

$\binom{6}{1} = 6$

relations of card. 2 15

max relation of card. 6

$$\Big\{ \emptyset,$$

$$\{(a,1)\}, \{(a,2)\}, \{(a,3)\}, \{(b,1)\}, \{(b,2)\}, \{(b,3)\},$$

$$\{(a,1), (a,2)\}, \{(a,2), (a,3)\}, \dots$$

card 3.

4.

5.

$$\{(a,1), (a,2), (a,3), (b,1), (b,2), (b,3)\} \Big\}$$

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

$dom(r) = \{a, b, c, d, e, f\}$

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

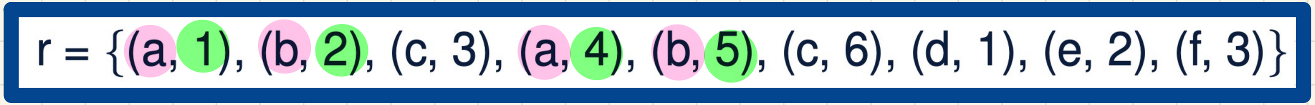
$ran(r) = \{1, 2, 3, 4, 5, 6\}$

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

① $dom(r^{-1}) = ran(r)$ ② $ran(r^{-1}) = dom(r)$

$r^{-1} = \{(1, a), (2, b), (3, c), (4, a), (5, b), (6, c),$
$(1, d), (2, e), (3, f)\}$

$r : S \leftrightarrow T$



r = {(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)}

$$r[\underbrace{\{a, b\}}_{\subseteq S}] = \{1, 2, 4, 5\}$$

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

$\{a, b\} \lhd r = \{(a, 1), (b, 2), (a, 4), (b, 5)\}$

r domain-restricted to {a, b}

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

$r \rhd \{1, 2\} = \{(a, 1), (b, 2), (d, 1), (e, 2)\}$

r range-restricted to {1, 2}

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

$\{a, b\} \mathbin{⩤} r = \{(c, 3), (c, 6), (d, 1), (e, 2), (f, 3)\}$

r domain-subtracted by {a, b}

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

$r \mathbin{⩥} \{1, 2\} = \{(c, 3), (a, 4), (b, 5), (c, 6), (f, 3)\}$

r range-subtracted by {1, 2}

Lecture 1b

Review on Math (continued)

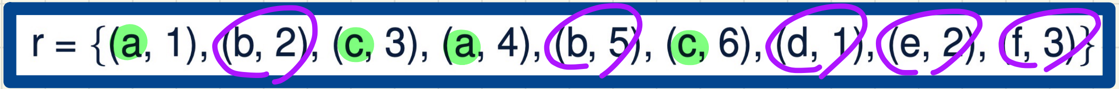
$$\underset{\text{set}}{s} \mathbin{⩤} \underset{\text{relation}}{r}$$

domain subtraction

$$\underset{\text{relation}}{r} \overset{\text{overridden by}}{\mathbin{\lhd\!\!\!+}} \underset{\text{relation}}{t}$$

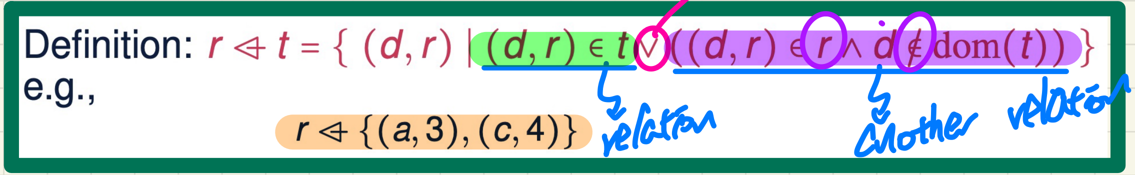
overriding

✓



$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

✓



Definition: $r \mathbin{\lhd\!\!\!-} t = \{ (d,r) \mid \underbrace{(d,r) \in t}_{\text{relation}} \overset{\text{union}}{\vee} \underbrace{((d,r) \in r \wedge d \notin \text{dom}(t))}_{\text{another relation}} \}$

e.g.,

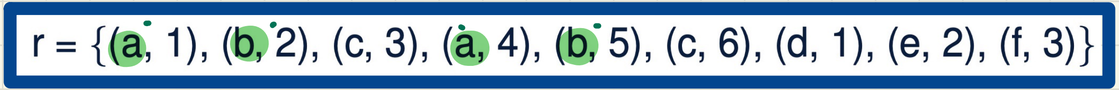
$r \mathbin{\lhd\!\!\!-} \{(a,3),(c,4)\}$

$$r \mathbin{\lhd\!\!\!-} \underline{\{(a,3),(c,4)\}}$$

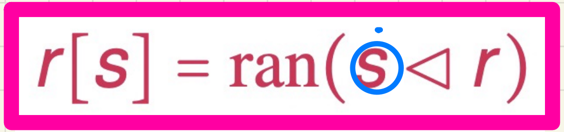
$$\text{dom}(t) = \{a, c\}$$

$$\{(a,3),\ (c,4)\} \cup \{(b,2),\ (b,5),\ (d,1),\ (e,2),\ (f,3)\}$$

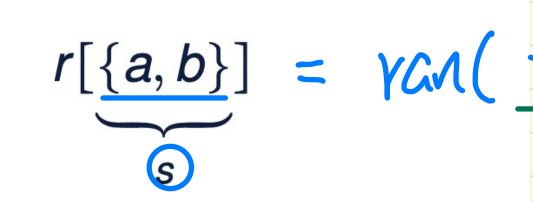
$$= \{ \underline{\hspace{6cm}} \}$$



$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$



$$r[s] = \mathrm{ran}(s \lhd r)$$



$$\underbrace{r[\{a, b\}]}_{s} = \mathrm{ran}(\underline{\{a, b\} \lhd r})$$

$$= \mathrm{ran}(\{(a, 1), (b, 2), (a, 4), (b, 5)\})$$

$$= \{1, 2, 4, 5\}$$

Side note:

databases

↳ relational databases (SQL queries)

↳ relational algebra

$$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$$

$$\boxed{r \mathbin{\lhd\!\!\!\!+} t = t \cup \underline{(\mathrm{dom}(t) \mathbin{\lhd\!\!\!\!-} r)}}$$ → algebraic property.

$$a + b = b + a$$

$$r \mathbin{\lhd\!\!\!\!+} \underbrace{\{(a, 3), (c, 4)\}}_{t}$$

$$r \mathbin{\lhd\!\!\!\!+} \{\underbrace{(a,3), (c,4)}_{t}\}$$

$$= \{(a,3), (c,4)\} \cup \underbrace{(\{a, c\} \mathbin{\lhd\!\!\!\!-} r)}_{\{(b,2),\ (b,5),\ (d,1),\ (e,2),\ (f,3)\}}$$

$$= \{\ \underline{\qquad\qquad\qquad\qquad}\ \}$$

$$
\boxed{
\begin{array}{l}
\texttt{isFunctional}(r) \\
\Longleftrightarrow \\
\forall s, t_1, t_2 \bullet \underline{(s \in S \land t_1 \in T \land t_2 \in T)} \Rightarrow \boxed{\overset{\checkmark}{(}(s, t_1) \overset{\checkmark}{\in} r \overset{\checkmark}{\land} (s, t_2) \in r \Rightarrow t_1 = t_2)}
\end{array}
}
$$

to disprove, find witness (satisfying antecedent but violating consequence)

≡ Contrapositive $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$

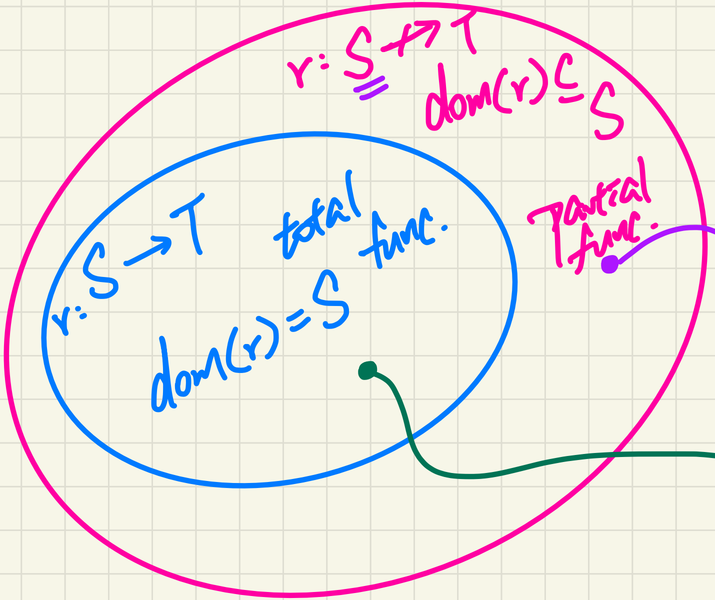
$$t_1 \neq t_2 \Rightarrow (s, t_1) \notin r \lor (s, t_2) \notin r$$

What is the smallest relation satisfying the functional property?

↳ $\emptyset$ ∵ we cannot find any witness to disprove that it violates the functional property ∴ $\emptyset$ is a functio

$\mathrm{dom} = \{2,1\}$ $\subseteq$ $\subset$ $\mathrm{dom} = \{2,3,1\}$ $\subseteq$ $\subset$ × ✓✓ ✓✓

$$\text{e.g., } \{ \underbrace{\{(\mathbf{2}, a), (\mathbf{1}, b)\}}_{\text{function}}, \underbrace{\{(\mathbf{2}, a), (\mathbf{3}, a), (\mathbf{1}, b)\}}_{\text{function}} \} \boxed{\subseteq} \underline{\{1,2,3\}} \underset{\text{the set of possible partial functions}}{\not\rightarrow} \underline{\{a,b\}}$$



partial, not total

s ∈ S s.t. r(s) undefined

total & partial

	injective	surjective	bijective
partial			✗
total			

Injective Functions

$isInjective(f)$

$\iff$

$\forall s_1, s_2, t \bullet (s_1 \in S \land s_2 \in S \land t \in T) \Rightarrow ((s_1, t) \in f \land (s_2, t) \in f \Rightarrow s_1 = s_2)$

$b = b \Rightarrow 1 = 3$ False

If f is a ***partial injection***, we write: $f \in S \rightarrowtail\!\!\!\!\!\!\!+\; T$

$\rightarrowtail\!\!\!\!\!\!\!+$

$\emptyset$: 1. not total

2. injective

∵ no witnesses of violation

- e.g., $\{\emptyset, \{(1,\mathbf{a})\}, \{(2,\mathbf{a}), (3,\mathbf{b})\}\} \subseteq \{1,2,3\} \rightarrowtail\!\!\!\!\!\!\!+\; \{a,b\}$
- e.g., $\{(1,\mathbf{b}), (2,a), (3,\mathbf{b})\} \notin \{1,2,3\} \rightarrowtail\!\!\!\!\!\!\!+\; \{a,b\}$
- e.g., $\{(1,\mathbf{b}), (3,\mathbf{b})\} \notin \{1,2,3\} \rightarrowtail\!\!\!\!\!\!\!+\; \{a,b\}$

partial, not inj.

If f is a ***total injection***, we write: $f \in S \rightarrowtail T$

$\rightarrowtail$

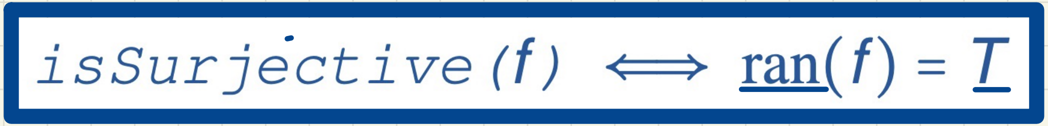
- e.g., $\{1,2,3\} \rightarrowtail \{a,b\} = \emptyset$ → $\{(1,a),(2,b),(3,a)\}$
- e.g., $\{(2,d), (1,a), (3,c)\} \in \{1,2,3\} \rightarrowtail \{a,b,c,d\}$
- e.g., $\{(2,d), (1,c)\} \notin \{1,2,3\} \rightarrowtail \{a,b,c,d\}$ → not total, inj
- e.g., $\{(2,\mathbf{d}), (1,c), (3,\mathbf{d})\} \notin \{1,2,3\} \rightarrowtail \{a,b,c,d\}$ → total, not inj.

$(2,d), (3,d)$

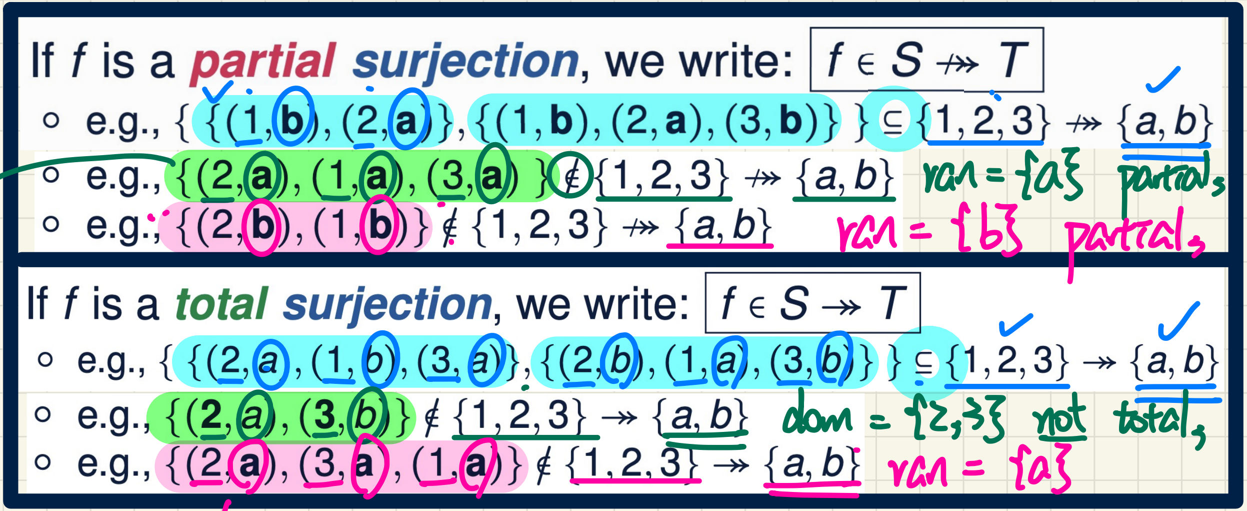
$d = d \Rightarrow 2 = 3$ False.

the set of all possible total injections

Surjective Functions



$\mathtt{isSurjective}(f) \iff \underline{\mathrm{ran}}(f) = \underline{T}$



If f is a ***partial surjection***, we write: $\boxed{f \in S \mathrel{+\!\!\!\twoheadrightarrow} T}$ $\mathrel{+\!\!\!\twoheadrightarrow}$

- e.g., $\{ \overset{✓}{\{(1,b),(2,a)\}}, \{(1,b),(2,a),(3,b)\} \} \subseteq \{1,2,3\} \mathrel{+\!\!\!\twoheadrightarrow} \overset{✓}{\{a,b\}}$
- e.g., $\{(2,a),(1,a),(3,a)\} \notin \{1,2,3\} \mathrel{+\!\!\!\twoheadrightarrow} \{a,b\}$ ran = {a} partial, not sur.
- e.g., $\{(2,b),(1,b)\} \notin \{1,2,3\} \mathrel{+\!\!\!\twoheadrightarrow} \{a,b\}$ ran = {b} partial, not sur.

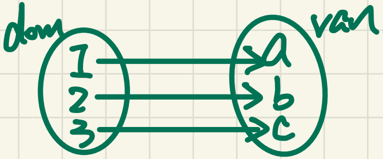
not ≡ inj.

If f is a ***total surjection***, we write: $\boxed{f \in S \twoheadrightarrow T}$ $\twoheadrightarrow$

- e.g., $\{ \{(2,a),(1,b),(3,a)\}, \{(2,b),(1,a),(3,b)\} \} \subseteq \overset{✓}{\{1,2,3\}} \twoheadrightarrow \overset{✓}{\{a,b\}}$ total, sur.
- e.g., $\{(2,a),(3,b)\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$ dom = {2,3} not total, sur.
- e.g., $\{(2,a),(3,a),(1,a)\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$ ran = {a}

total, not sur.

Bijective Functions



f is ***bijective***/***a bijection***/*one-to-one correspondence* if *f* is ***total***, ***injective***, and ***surjective***.

$\rightarrowtail\mathrel{\mkern-14mu}\twoheadrightarrow$

- e.g., $\{1,2,3\} \rightarrowtail\mathrel{\mkern-14mu}\twoheadrightarrow \{a,b\} = \varnothing$ $\{(1,a),(2,b),(3,\overset{a}{?})\}$
- e.g., $\{\{(1,a),(2,b),(3,c)\},\{(2,a),(3,b),(1,c)\}\} \subseteq \{1,2,3\} \rightarrowtail\mathrel{\mkern-14mu}\twoheadrightarrow \{a,b,c\}$
- e.g., $\{(2,b),(3,c),(4,a)\} \notin \{1,2,3,4\} \rightarrowtail\mathrel{\mkern-14mu}\twoheadrightarrow \{a,b,c\}$ not total, inj, sur.
- e.g., $\{(1,a),(2,b),(3,c),(4,a)\} \notin \{1,2,3,4\} \rightarrowtail\mathrel{\mkern-14mu}\twoheadrightarrow \{a,b,c\}$ total, not inj, sur.
- e.g., $\{(1,a),(2,c)\} \notin \{1,2\} \rightarrowtail\mathrel{\mkern-14mu}\twoheadrightarrow \{a,b,c\}$ ran = $\{a,c\}$

total ✓

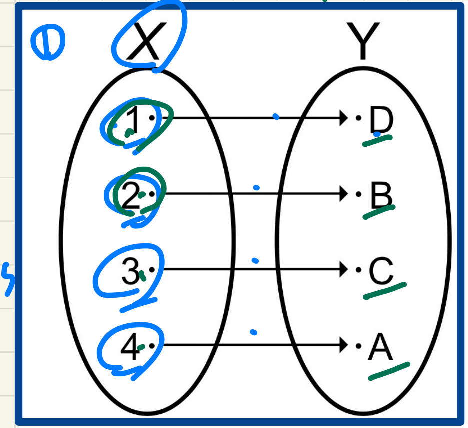
inj. ✓

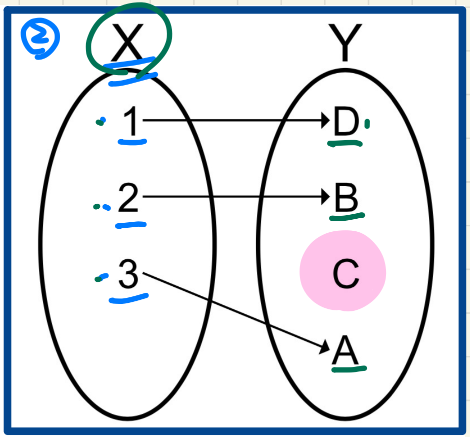
sur. ✗

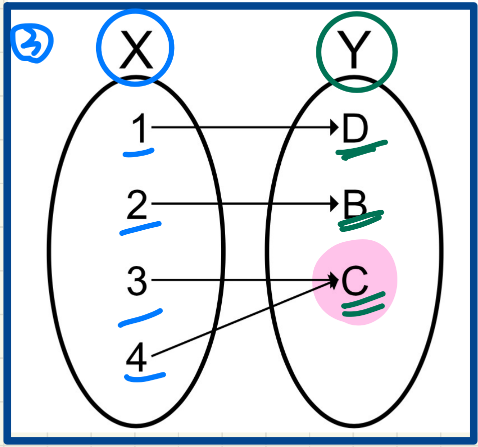
Exercise

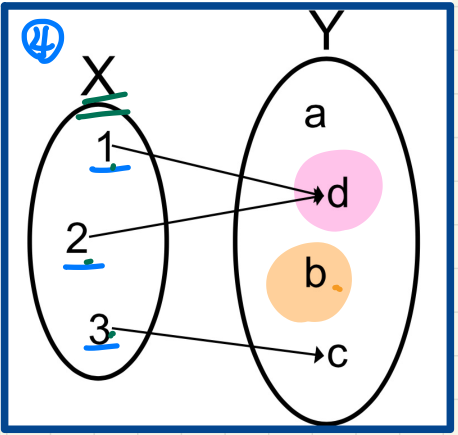
dom(①) = X
ran(①) = Y

✓ Exercise
Make a function that's partial but not total.









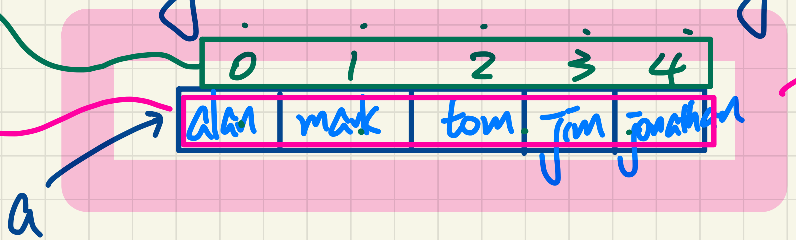
	①	②	③	④
partial	✓	✓	✓	✓
total	✓	✓	✓	✓
inj.	✓	✓	✗	✗
sur.	✓	✗	✓	✗
bij.	✓	✗	✗	✗

Formalizing Arrays as Functions

String[] a = new String[5];

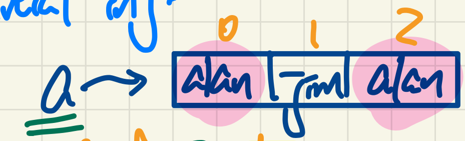
indices ≈ domain

content ≈ range



programming

Not partial inj.



a = {(0, alan), (1, jim), (2, alan)}

Strings 0 "" 1 "_" 2 "_ _"

$a = \{(0, \text{"alan"}), (1, \text{"mark"}), (2, \text{"tom"}), (3, \text{"jim"}), (4, \text{"jona."})\}$

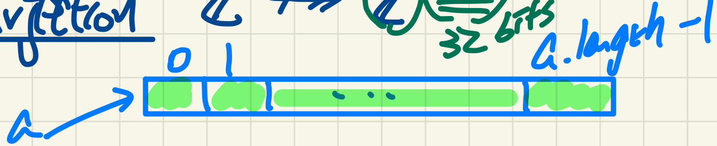
formalization in math.

Should a be formalized/modeled as a relation?

No. ∵ {(0, alan), (0, jim)} → in reality, only one element may be stored at each index

$\mathbb{Z} \leftrightarrow String$

Partial Surjection $\mathbb{Z} \mathrel{+\!\!\twoheadrightarrow} \mathbb{Z}$ int 32 bits



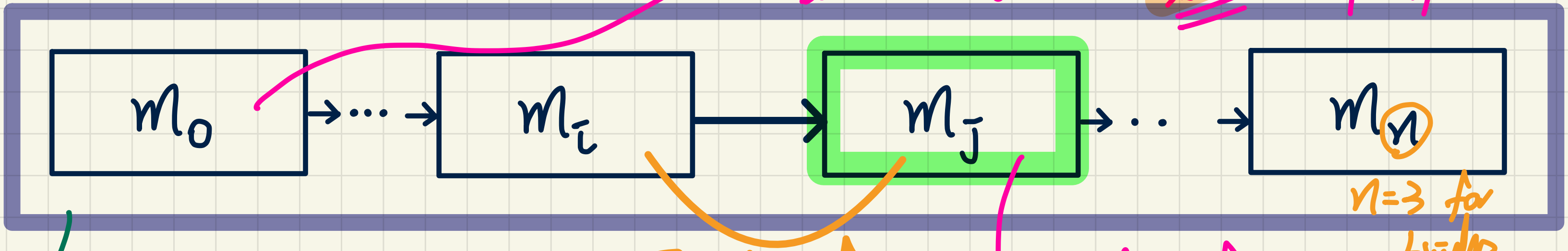
Lecture 2

Part A

Case Study on Reactive Systems - Bridge Controller Introduction, State Space, Req. Doc.

Correct by Construction

most abstract, but we can expect to encode & prove some constraints & properties from RD.



$n=3$ for bridge controller

Each model formalizes the public view of the system under construction.

PO of refinement:

m_j behaves consistently w.r.t. m_i.

1. m_j refines m_i (by intro. extra state variable and/or events)

RD (requirements document)

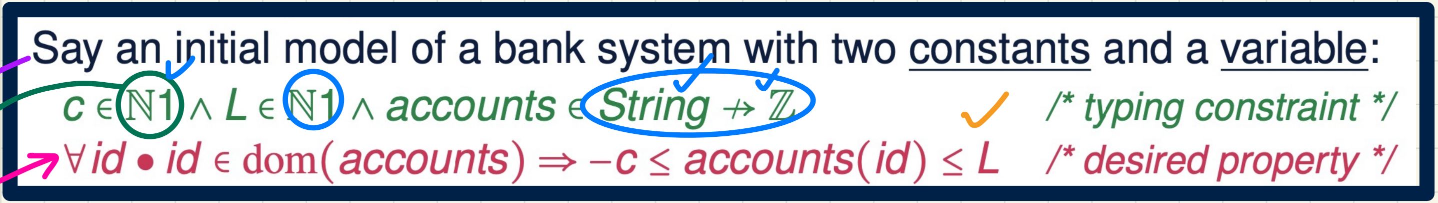
- E-descriptions
- R-description

State Space of a Model

Invalid Configuration/Valuation: witness of violation
(C=4000, L=175,000, {("id1", -4500)})

Definition: The state space of a model is the set of all possible valuations of its declared constants and variables, subject to declared constraints.

typing, properties
axioms theorems


Say an initial model of a bank system with two constants and a variable:

$c \in \mathbb{N}1 \land L \in \mathbb{N}1 \land accounts \in String \nrightarrow \mathbb{Z}$ ✓ /* typing constraint */

$\forall id \bullet id \in \mathrm{dom}(accounts) \Rightarrow -c \leq accounts(id) \leq L$ /* desired property */

pos. num.

theorem proving

Q1. Give some example configurations of this initial model's state space.

Ex.1. (C=3000, L=150,000, accounts = ∅)
↳ empty bank
∈ state space

Ex.2. (C=3500, L=200,000, accounts = {("id1", 150), ("id2", 1750)})
∈ state space

Q2. How large exactly is this initial model's state space?

↳ Combinatorial explosion

$|\mathbb{N}1| \times |\mathbb{N}1| \times |String \nrightarrow \mathbb{Z}|$

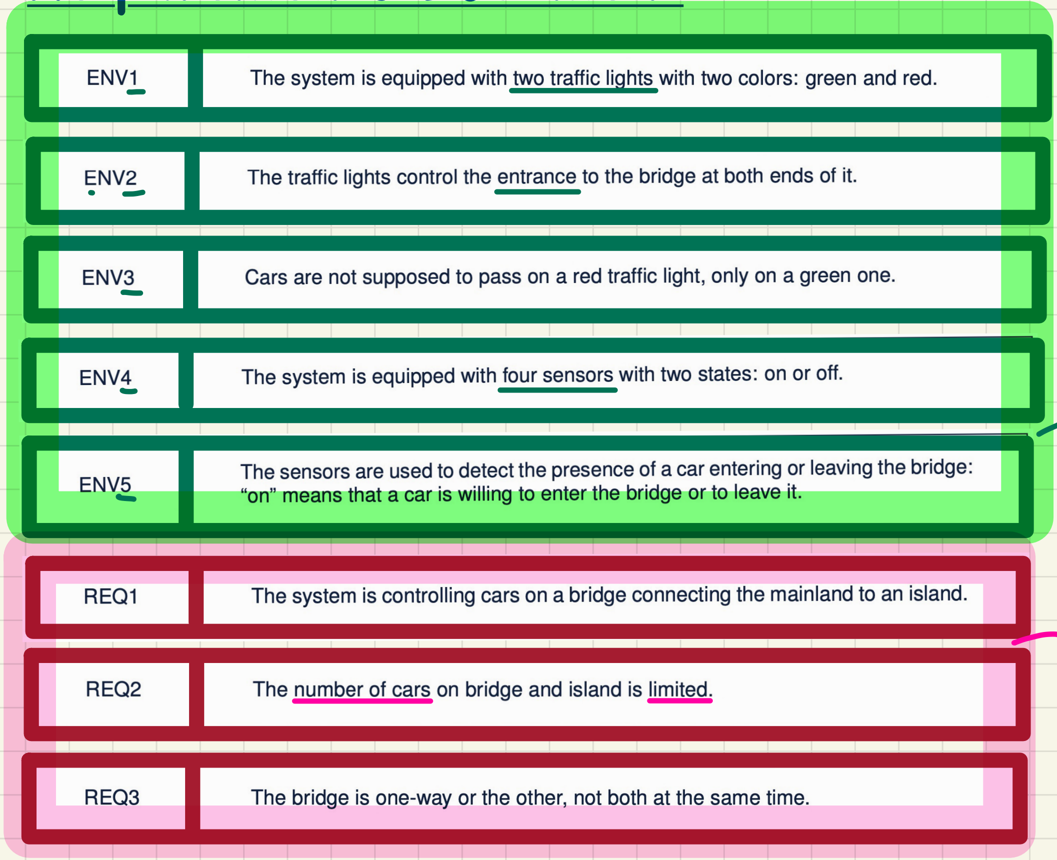
↳ infinite

① infeasible to test all possible values
② theorem proving can address this.

⇒ manipulation of symbols & predicates/props. at the abstract level (vs. concrete valuations for individual JUnit tests)

Bridge Controller:

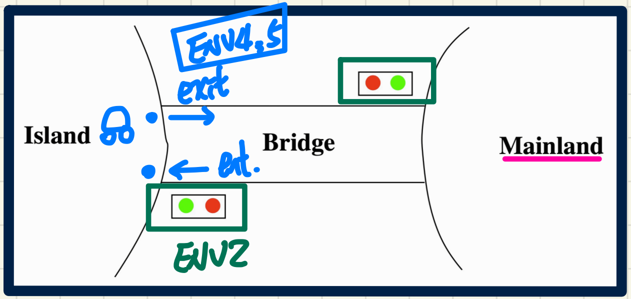
Requirements Document



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.
ENV4	The system is equipped with four sensors with two states: on or off.
ENV5	The sensors are used to detect the presence of a car entering or leaving the bridge: "on" means that a car is willing to enter the bridge or to leave it.
REQ1	The system is controlling cars on a bridge connecting the mainland to an island.
REQ2	The number of cars on bridge and island is limited.
REQ3	The bridge is one-way or the other, not both at the same time.

E-descriptions (working environment)

R-descriptions (functionalities, properties)

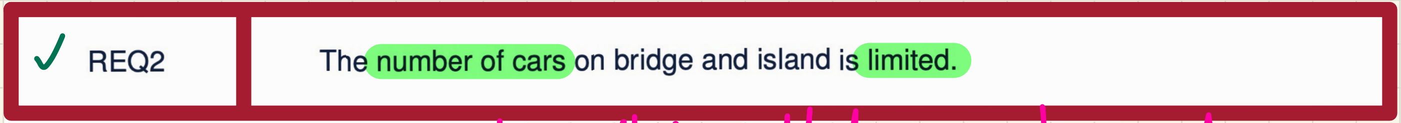


Lecture 2

Part B

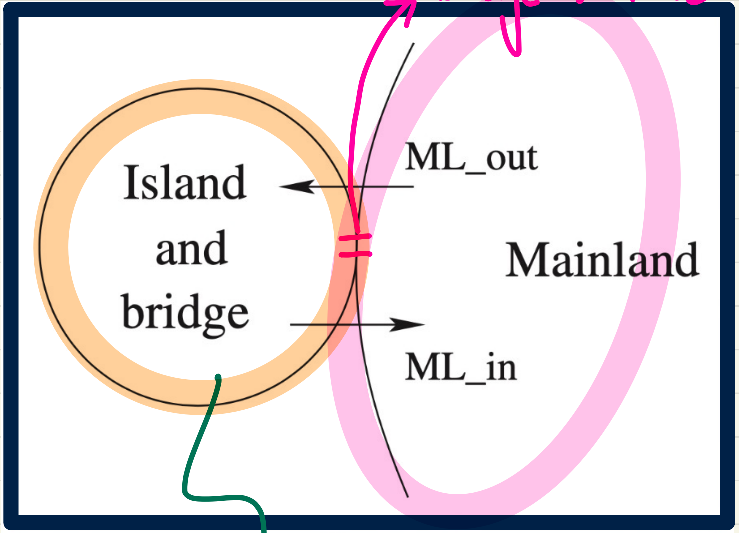
Case Study on Reactive Systems - Bridge Controller
Initial Model: State and Events

Bridge Controller: Abstraction in the Initial Model



✓ REQ2	The number of cars on bridge and island is limited.

bridge will be added at a later refinement.



abstraction: abstract away the bridge existing between the island & mainland.

Bridge Controller: State Space of the Initial Model

thm: theorem

REQ2	The number of cars on bridge and island is limited.

n $n \le d$

Static Part of Model

Context in Rodin

constants: d

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

assumed to be true

initial model m0

axiom

1st axiom of m0

Dynamic Part of Model

$I \triangleq n \in \mathbb{N} \land n \le d$

m0

variables: n

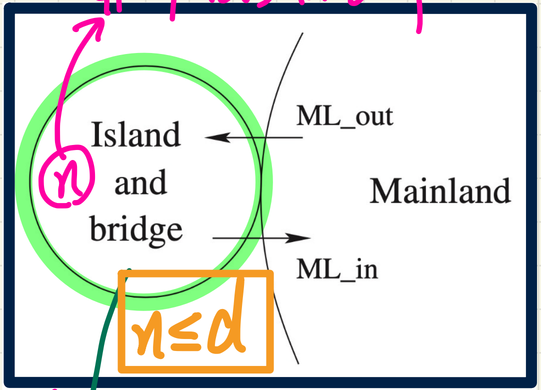
invariants: I
inv0_1 : $n \in \mathbb{N}$
inv0_2 : $n \le d$

typing constraint

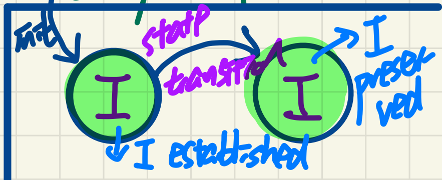
property

Interactions between system and users continue "forever"

# of cars in compound currently



max d cars in the compound



Bridge Controller: State Transitions of the Initial Model

Trace: ⟨ML_out, ML_out, ML_out⟩

abstraction w.r.t. REQ2.

REQ2	The number of cars on bridge and island is limited.

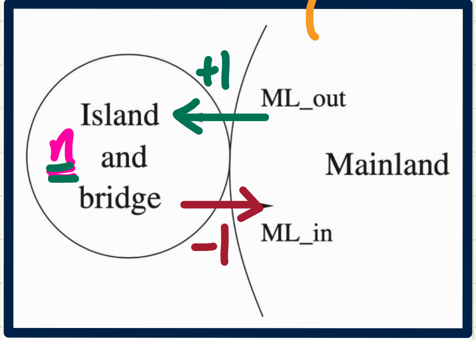
state space

constants: d

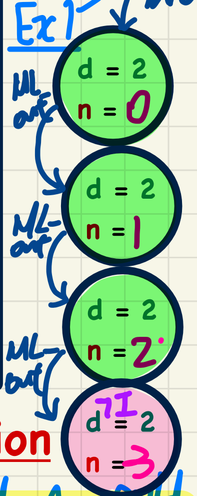
axioms:
axm0_1 : $d \in \mathbb{N}$

variables: n

invariants: $I \triangleq n \in \mathbb{N} \wedge n \leq 2$
inv0_1 : $n \in \mathbb{N}$
inv0_2 : $n \leq d$



Ex 1



guards → events enabled if they evaluate to true.

mainland

```
ML_out
  begin
    n := n + 1
  end
```

as if: when True

```
ML_in
  begin
    n := n − 1
  end
```

when True

always enabled

becomes

actions of events

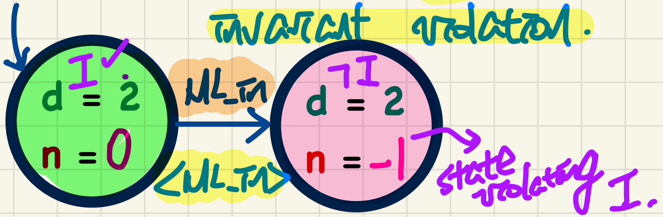
State Transition Diagram on an Example Configuration

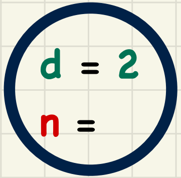
d = 2

n initialized to 0

Are ML_in and ML_out specified correctly s.t. there's not a trace leading to invariant violation.

Ex 1





Before-After Predicates of Event Actions

- Pre-State
- Post-State
- State Transition

post-state → X variable assignment

Events	ML_out	ML_in
	$n := n + 1$	$n := n - 1$
before–after predicates	$n' = n + 1$	$n' = n - 1$

specification

post-state value

actions

becomes

∀ n in the post-state becomes its pre-state value +1

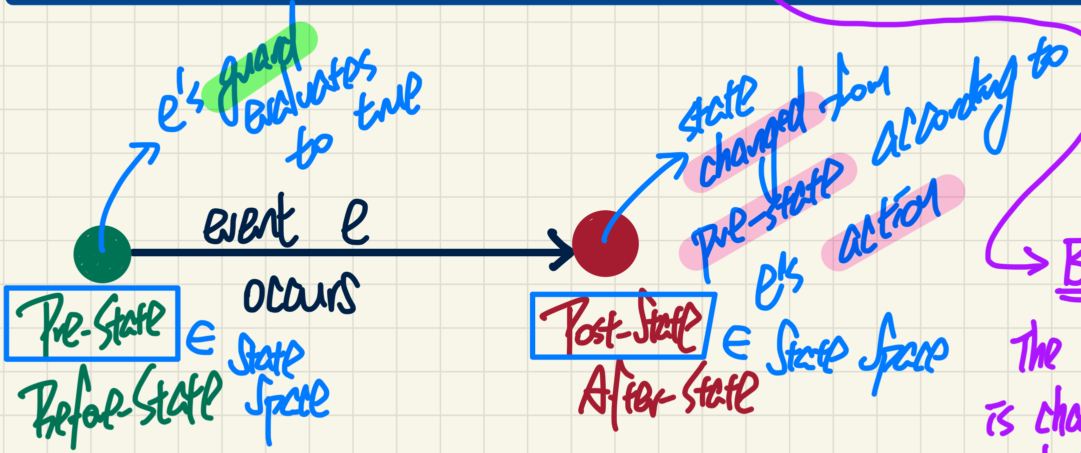
pre-state value

↳ POs related to events.

Variable x:

Unprimed version x denotes its value in pre-state.

Primed version x' denotes its value in post-state.



BAP

$$\underset{\text{post-state}}{n'} = \underset{\text{pre-state}}{n} + 1$$

The effect of ML_out's occurrence is characterized as a relation between its pre-state and post-state.

Lecture 2

Part C

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

Design of Events: Invariant Preservation

variables: n

dynamic part
↳ values might change via actions of events

```
ML_out
  begin
    $n := n + 1$
  end
```

```
ML_in
  begin
    $n := n - 1$
  end
```

always enabled

guard: true

guard: true

enabled
↳ guards evaluating to true

$\forall s \bullet s \in StateSpace \Rightarrow invariants(s)$

$\equiv$

$\neg(\exists s \bullet s \in StateSpace \land \neg invariants(s))$

witness for disproving

✓

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

✓ important properties of the system that must always hold true

the state space being consistent

may or may not be consistent

✓ state space : configurations
↳ variable values
constant values
↳ invariants

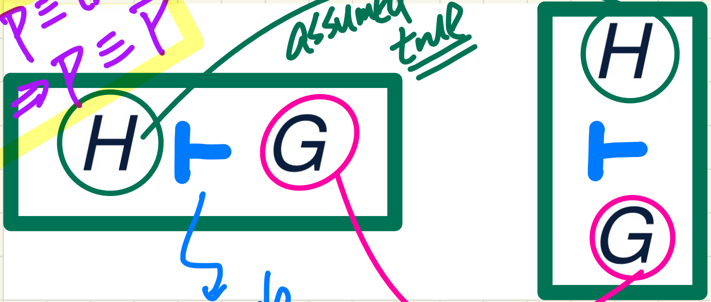
inconsistent s.s. if some combination of var. and c. violates the invariant. ✓

Sequents: Syntax and Semantics

Syntax

Zero of ⇒: $false \Rightarrow P \equiv true$

Identity of ⇒: $true \Rightarrow P \equiv P$



$H \vdash G$

$\begin{array}{c} H \\ \vdash \\ G \end{array}$

hypotheses/assumptions
(a set of predicates)
↳ might be empty

goal (a set of predicates)
↳ should not be empty.

Semantics

$\boxed{H \vdash G}$ → a predicate
↓
proved or disproved.

provable assuming H

$\boxed{H \vdash G \Leftrightarrow H \Rightarrow G}$

✓

Q. What does it mean when **H** is empty/absent?

$\boxed{\vdash G}$ $\boxed{\begin{array}{c} \vdash \\ G \end{array}}$ $\overset{?}{\equiv}$ ✗ $\boxed{false \vdash G}$ ✓ $\boxed{true \vdash G}$

↳ $false \Rightarrow G \equiv True$

↳ $true \Rightarrow G \equiv G$

PO/VC Rule of Invariant Preservation

Identity of $\wedge$: $P \wedge true \equiv P$
zero of $\wedge$: $P \wedge false \equiv false$
model m0

substitute n' by expression on n according to the BAP

constants: d

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

Pre-State
$n \in \mathbb{N} \wedge n \leq d$

variables: n

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

→ Post-State
$n' \in \mathbb{N} \wedge n' \leq d$
$n+1$ $n-1$

guard: true ←
ML_out
begin
$n := n + 1$
end
BAP: $n' = n+1$

guard: true ←
ML_in
begin
$n := n - 1$
end
BAP: $n' = n-1$

Variable
n before-state
n' after-state

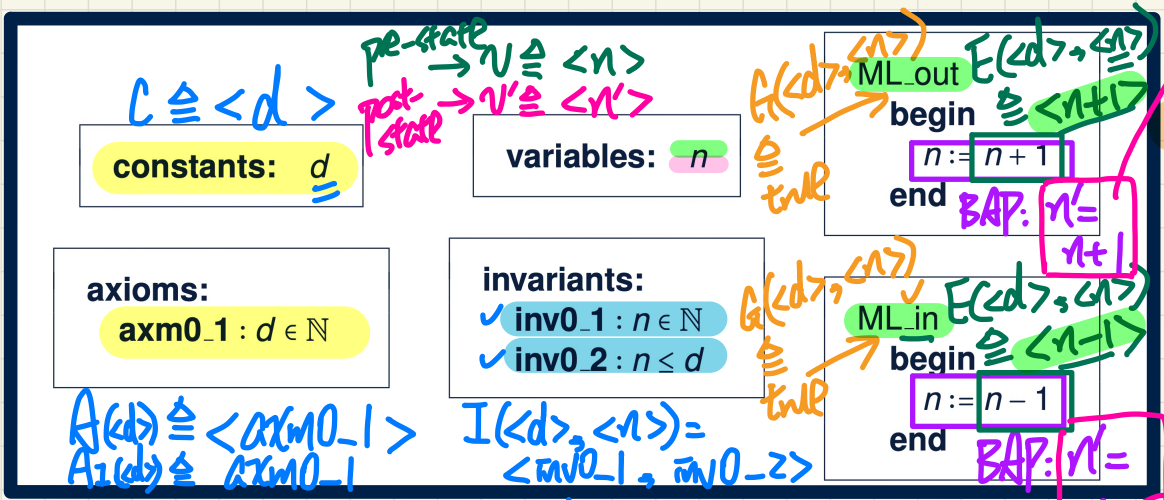
$H1 \wedge \cancel{H2} \vdash G$ (True)
$= H1 \wedge \cancel{H2} \Rightarrow G$ (True)

Axioms
Invariants Satisfied at *Pre-State*
↳ before state transition occurs
Guards of the Event
↳ true
⊢
Invariants Satisfied at *Post-State*

INV → name of rule

PO/VC rule of invariant preservation
for a single event

PO/VC Rule of Invariant Preservation: Components



constants: d

axioms:

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

variables: n

invariants:

✓ inv0_1 : $n \in \mathbb{N}$

✓ inv0_2 : $n \leq d$

ML_out
begin
$n := n + 1$
end

ML_in
begin
$n := n - 1$
end

$c \triangleq \langle d \rangle$

pre-state → $v \triangleq \langle n \rangle$

post-state → $v' \triangleq \langle n' \rangle$

$A(\langle d \rangle) \triangleq \langle axm0\_1 \rangle$

$A_1(\langle d \rangle) \triangleq axm0\_1$

$I(\langle d \rangle, \langle n \rangle) = \langle inv0\_1, inv0\_2 \rangle$

$G(\langle d \rangle, \langle n \rangle) \triangleq true$

$E(\langle d \rangle, \langle n \rangle) \triangleq \langle n+1 \rangle$

BAP: $n' = n+1$ → $\langle n' \rangle = \langle n+1 \rangle$

$G(\langle d \rangle, \langle n \rangle) \triangleq true$

$E(\langle d \rangle, \langle n \rangle) \triangleq \langle n-1 \rangle$

BAP: $n' = n-1$ → $\langle n' \rangle = \langle n-1 \rangle$

Each PO rule should be instantiated for every event.

$I_1(\langle d \rangle, \langle n \rangle) = inv0\_1$

$I_2(\langle d \rangle, \langle n \rangle) = inv0\_2$

c: list of constants

A(c): list of axioms

v and v′: variables in pre- and post-state

I(c, v): list of invariants

G(c, v): guards of an event

↳ determines enabledness of event

E(c, v): effect of an event's actions

↳ values of variables in post-state i.t.o pre-state exp.

v′ = E(c, v): BAP of an event's actions

PO/VC Rule of Invariant Preservation: Sequents

for a single inv. condition for a single event

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
begin
$n := n + 1$
end

ML_in
begin
$n := n - 1$
end

true guard

true guard

all invariants

hypotheses

event enabled

post-state: $n' \in \mathbb{N}$ → $n+1 \in \mathbb{N}$

$n' \leq d$ → $n+1 \leq d$

Rule of PO (I.P.)

$$\begin{array}{l} A(c) \\ I(c, v) \\ G(c, v) \\ \vdash \\ I_i(c, E(c, v)) \end{array}$$

pre-state

guard & effect of the event under consideration

goal

index of an inv. condition

post-state value i.t.o pre-state exp. specified in the event's actions

↳ BAP.

Q. How many PO/VC rules for model m0?

1. # of events (state transitions) $|\{ML\_out, ML\_in\}|$
2. # of invariant conditions $|\{inv0\_1, inv0\_2\}|$

$|\{ML\_out, ML\_in\}| \times |\{inv0\_1, inv0\_2\}| = 4$

event / inv. cond. / kind of PO

① ML_out/inv0_1/INV

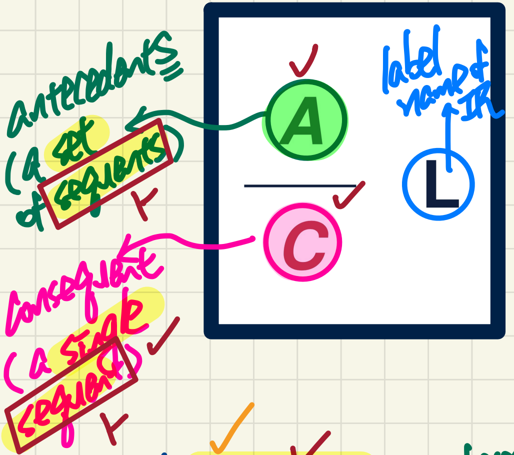
② ML_out/inv0_2/INV

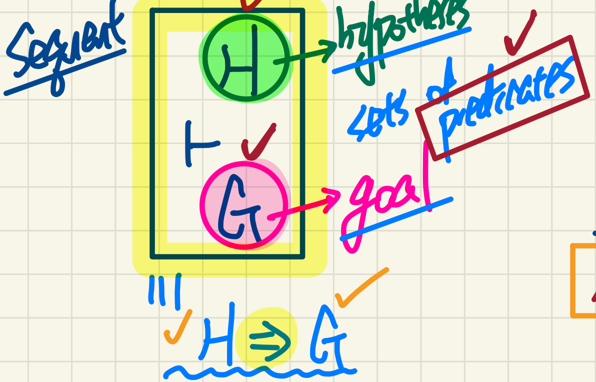
① $d \in \mathbb{N}$, $n \in \mathbb{N}$, $n \leq d$ $\vdash$ $n+1 \in \mathbb{N}$

② $d \in \mathbb{N}$, $n \in \mathbb{N}$, $n \leq d$ $\vdash$ $n+1 \leq d$

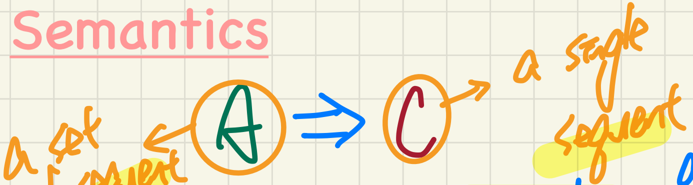
Inference Rule: Syntax and Semantics

Syntax


antecedents ≡ (a set of sequents) ⊢
A
C
L
label name of IR
consequent (a single sequent) ⊢

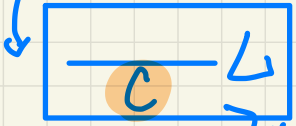


Semantics



Think of an IR is stating that an implication whose antecedent & consequence are both sets of predicates and that

Q. What does it mean when A is empty/absent?



implication is an axiom ready to use to prove C, nothing else to prove

Examples

IR1 → $(H1 \Rightarrow G) \Rightarrow (H1 \land H2 \Rightarrow G)$

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

$H1 \Rightarrow G$

$H1 \land H2 \Rightarrow G$

monotonicity

IR2 True $\Rightarrow (n \in N \Rightarrow n+1 \in N)$

axiom $\equiv (n \in N \Rightarrow n+1 \in N)$

$$\frac{}{n \in N \vdash n+1 \in N}\ \text{P2}$$

$n \in N \Rightarrow n+1 \in N$

Proof of Sequent: Steps and Structure

Outstanding Sequent to Prove

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \le d \\ \vdash \\ n+1 \in \mathbb{N} \end{array}$$

ML_out/inv0_1/INV

Known Inference Rules

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

$$\frac{}{n \in \mathbb{N} \vdash n+1 \in \mathbb{N}}\ \textbf{P2}$$

Ⓒ

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \quad H1 \\ n \le d \\ \vdash \\ n+1 \in \mathbb{N} \end{array}$$

H2

MON

Ⓐ

$$\begin{array}{l} n \in \mathbb{N} \\ \vdash \\ n+1 \in \mathbb{N} \end{array}$$

P2

to prove the original, outstanding sequent, it's sufficient to prove this instead.

Justifying Inference Rule: OR_L

$$\frac{\boxed{\cancel{H}, P \vdash R \qquad \cancel{H}, Q \vdash R}^{\wedge}}{\cancel{H}, P \lor Q \vdash R}\ \checkmark\ \text{OR\_L}$$

$$\boxed{(P \Rightarrow R) \land (Q \Rightarrow R)} \overset{\checkmark}{\Rightarrow} ((P \lor Q) \Rightarrow R)$$

$(p \Rightarrow R) \land (Q \Rightarrow R)$

$\equiv$ < def. of imp: $p \Rightarrow q \equiv \neg p \lor q$ >

$(\neg p \lor \underline{\underline{R}}) \land (\neg Q \lor \underline{\underline{R}})$

$\equiv$ < def. of dist. $\lor$ over $\land$: $p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$ >

$R \lor (\neg p \land \neg \overset{Q}{q})$

$\equiv$ < de morgan : $\neg(p \lor q) \equiv \neg p \land \neg q$ >

$\neg(\underline{p \lor \overset{Q}{q}}) \lor \underline{R}$ $\equiv$ < def. of imp. > $p \lor \overset{Q}{q} \Rightarrow R$

Example Inference Rules

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

$$\frac{}{n \in \mathbb{N}\ \vdash\ n+1 \in \mathbb{N}}\ \textbf{P2}$$

$$\frac{}{n < m\ \vdash\ n+1 \leq m}\ \textbf{INC}$$

$$\frac{}{0 < n\ \vdash\ n-1 \in \mathbb{N}}\ \textbf{P2'}$$

$$\frac{}{n \leq m\ \vdash\ n-1 < m}\ \textbf{DEC}$$

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

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

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

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

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

Discharging POs of original m0: Invariant Preservation

ML_out/inv0_1/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$\vdash$
$n + 1 \in \mathbb{N}$

MON

$n \in \mathbb{N}$
$\vdash$
$n+1 \in \mathbb{N}$

PZ

ML_in/inv0_1/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$\vdash$
$n - 1 \in \mathbb{N}$

MON

$n \in \mathbb{N}$
$\vdash$
$n-1 \in \mathbb{N}$

?

ML_out/inv0_2/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$\vdash$
$n + 1 \leq d$

MON

$n \leq d$
$\vdash$
$n+1 \leq d$

?

ML_in/inv0_2/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$\vdash$
$n - 1 \leq d$

MON

$n \leq d$
$\vdash$
$\boxed{n-1 \leq d}$

$\downarrow$

$\underline{n-1<d} \lor n-1=d$

OR_R1

$n \leq d$
$\vdash$
$n-1 < d$

DEC

Discharging POs of revised m0: Invariant Preservation

ML_out/inv0_1/INV

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ n < d \\ \vdash \\ n + 1 \in \mathbb{N} \end{array}$$

Exercise

Conclusion

m0 as is

is correct

w.r.t.

Invariant

Preservation

ML_in/inv0_1/INV

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ n > 0 \\ \vdash \\ n - 1 \in \mathbb{N} \end{array}$$

MON

$$\boxed{\begin{array}{l} n>0 \\ \vdash \\ n-1\in\mathbb{N} \end{array}}$$

P2'

ML_out/inv0_2/INV

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ n < d \\ \vdash \\ n + 1 \leq d \end{array}$$

MON

$$\boxed{\begin{array}{l} n<d \\ \vdash \\ n+1\leq d \end{array}}$$

INC

ML_in/inv0_2/INV

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ n > 0 \\ \vdash \\ n - 1 \leq d \end{array}$$

Exercise

Lecture 2

Part D

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

Initializing the System → ASM

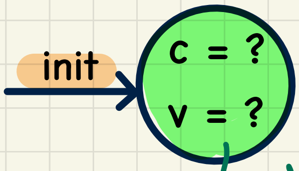
Analogy to Induction:

Base Cases ≈ Establishing Invariants

P(1)

P(2)

⋮



initial state when the system is first launched.

Analogy to Induction:

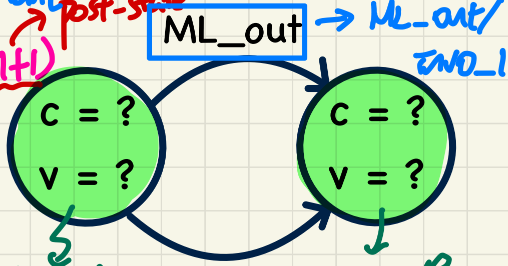
Inductive Cases ≈ Preserving Invariants

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
```

~~n := n+1~~

no notion of pre-state value

variable

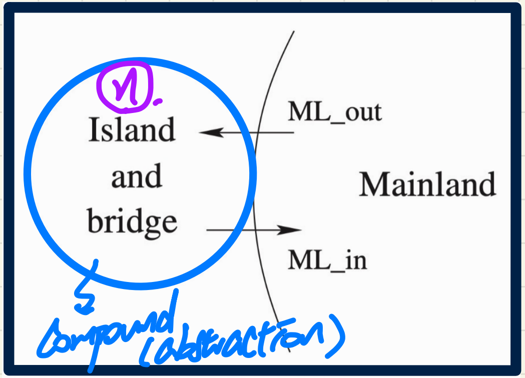
value in post-state

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 : $n \le d$ ✓

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

BAP: $n' = 0$

Components

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

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

constants

specified t.t.o. constants or literals.

only the notion of post-state is applicable.

Rule of Invariant Establishment

$$\frac{\begin{array}{l} A(c) \\ \vdash \\ I_i(c, \boxed{K(c)}) \end{array}}{} \quad \text{INV}$$

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

$$\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ \overset{0}{\cancel{n}} \in \mathbb{N} \end{array}$$

init/inv0_2/INV

$$\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ \overset{0}{\cancel{n}} \le d \end{array}$$

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 ✓

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

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

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

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

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.

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

1. 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.)

$\bot$ $\top$

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

$H \land P \Rightarrow P$

↳ theorem without further justification

axiom IRs

$$\frac{}{\boxed{\bot \vdash P}}\ \text{FALSE\_L}$$

false, "bottom"

$\bot \Rightarrow P \equiv \top$ (zero of $\Rightarrow$)

$$\frac{}{P \vdash \top}\ \text{TRUE\_R}$$

true, "top"

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

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

$\top$

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

$E=F$

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

from left to right

$$\frac{H(E), \boxed{E = F} \vdash P(E)}{H(F), \boxed{E = F} \vdash P(F)}\ \text{EQ\_RL}$$

$E=F$

from R to L ↑

replace F by E

Discharging PO of DLF: First Attempt

* $\underline{d > 0}$ → max # cars ≥ 1

* $n > 0$ → # cars ≥ 1

max = 0 should be avoided

not reasonable to impose on model

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

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

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

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

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

Mo may not be sufficient

$\cancel{d \in \mathbb{N}}$

$n \in \mathbb{N}$ $n \geq 0$

$n \leq d$ upper bound of n

$\vdash$

$n < d \vee n > 0$

guard of ML_out

guard of ML_in

≡

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

MON

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

OR_L

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

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

alternatively

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

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

Understanding the Failed Proof on DLF

① $d=0$: max 0 cars on the IB comp.

② $n=0$ by init

constants: d

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

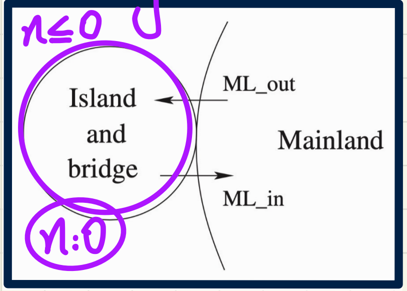
↳ revision on model based on

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
```



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

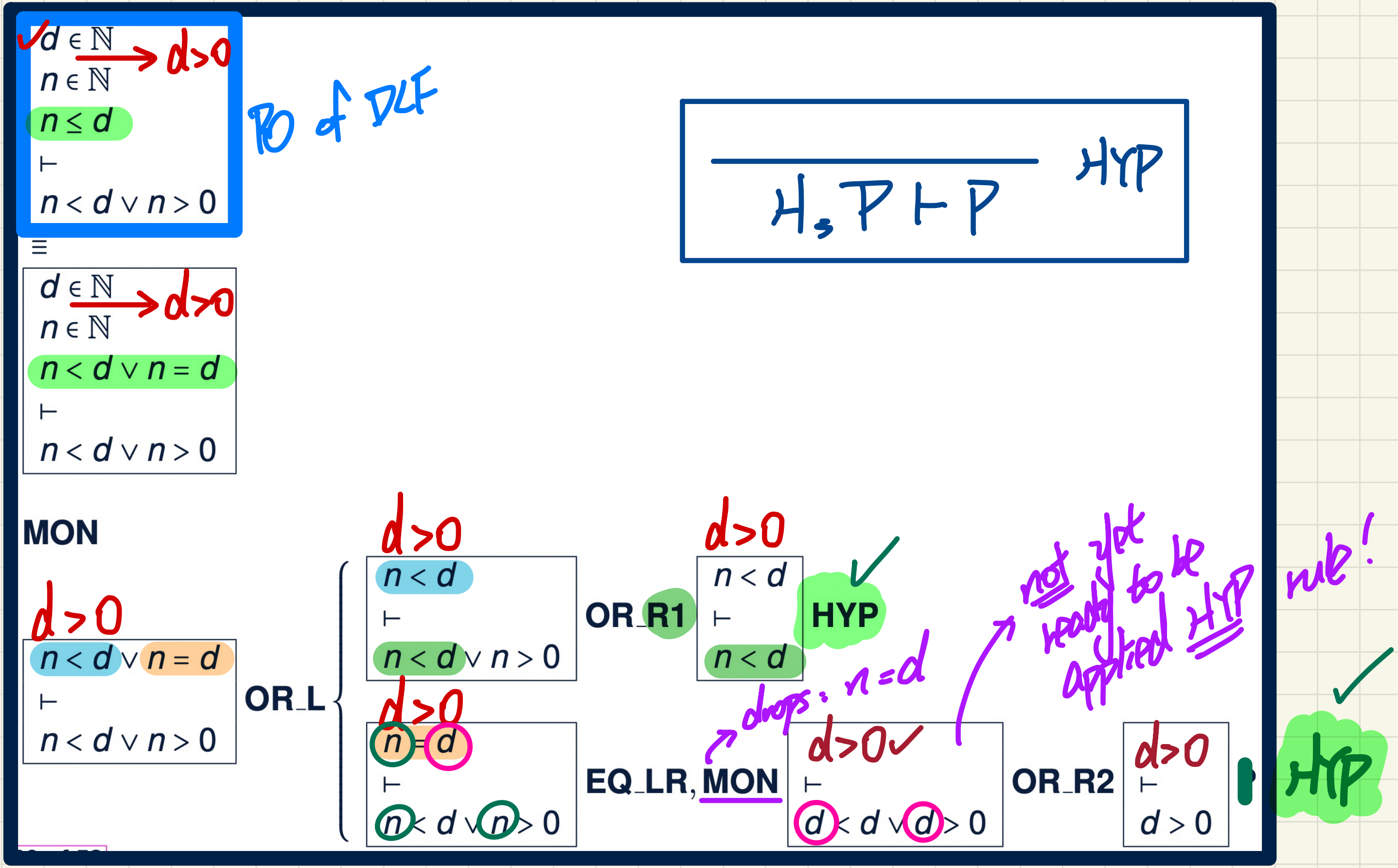
init : $n' = 0$

$$\boxed{\underset{0}{\cancel{n}} < \underset{0}{\cancel{d}} \vee \underset{0}{\cancel{n}} > 0} \rightarrow \text{false } \bot$$

both events are disabled
↳ deadlock!!

Discharging PO of DLF: Second Attempt

added axiom:
axm0_2: $d > 0$



PO of DLF

$\checkmark d \in \mathbb{N} \rightarrow d>0$
$n \in \mathbb{N}$
$n \leq d$
$\vdash$
$n < d \lor n > 0$

$\equiv$

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

MON

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

OR_L

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

OR_R1

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

HYP ✓

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

EQ_LR, MON

drops: $n=d$

$d>0 \checkmark$
$\vdash$
$d < d \lor d > 0$

not yet ready to be applied HYP rule!

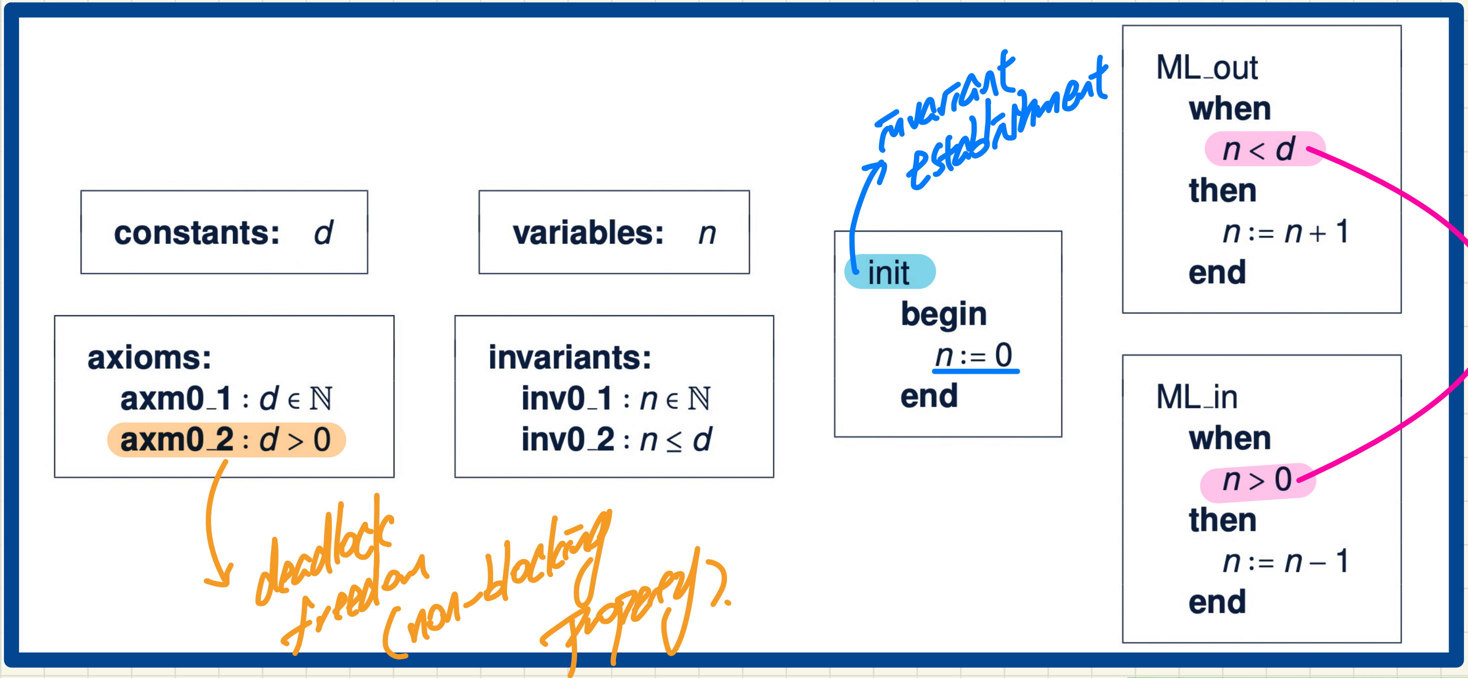
OR_R2

$d>0$
$\vdash$
$d > 0$

HYP ✓

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

Summary of the Initial Model: Provably Correct



constants: d

variables: n

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

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:

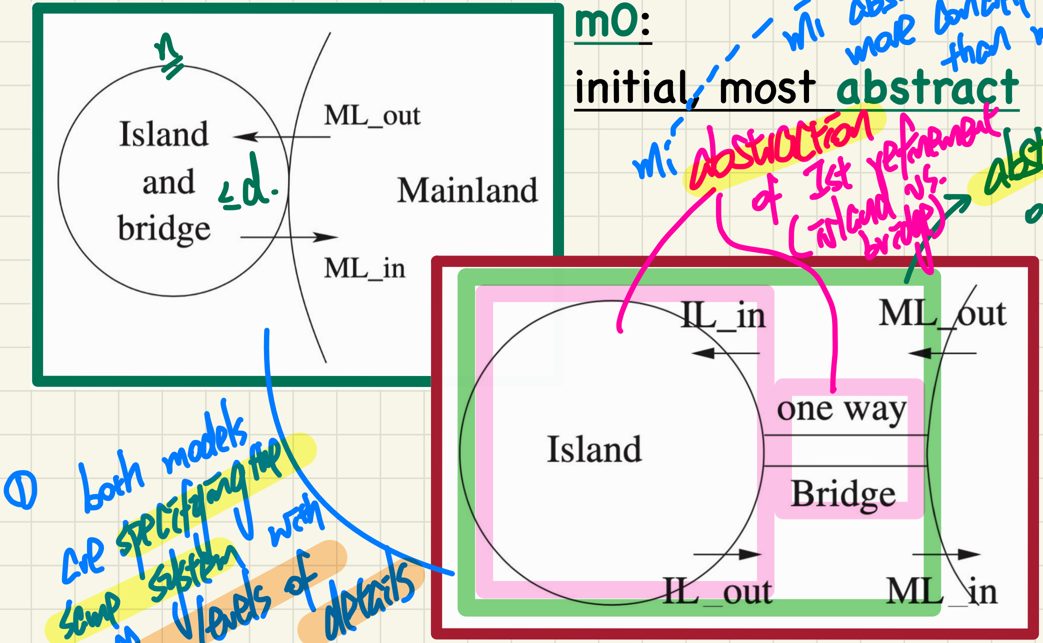
- + Invariant Establishment
- + Invariant Preservation
- + Deadlock Freedom

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:

second, more concrete

m1 abs. more concrete than m0 abs.

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

abstraction of m0 initial model (IB compound).

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.

Lecture 2

Part F

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

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

Counter example to violate this safety inv.

unsafe

$a = 2$
$c = 1$
$b = ?$

↳ crash ↓ need inv1_5 to disallow.

variables: a, b, c

$c = 0 \lor a = 0$

flow to IL   flow to ML

invariants:

- inv1_1 : $a \in \mathbb{N}$
- inv1_2 : $b \in \mathbb{N}$
- inv1_3 : $c \in \mathbb{N}$
- inv1_4 : ??
- inv1_5 : ??

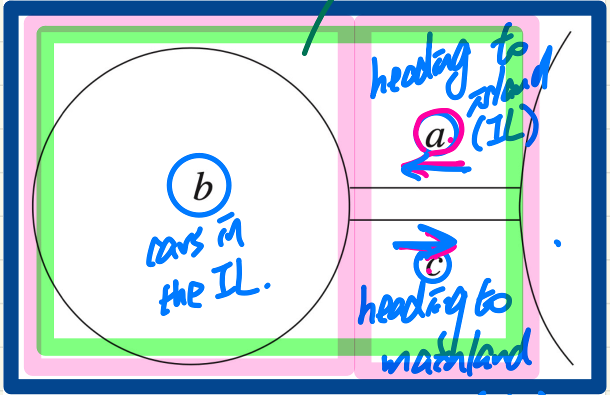
abstract state

$n = a+b+c$

concrete state

1st refinement m1

n : ≙ IB compound.



(ML)

Static Part of Model

constants: d

axioms:

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

axm0_2: $d > 0$

Exercises

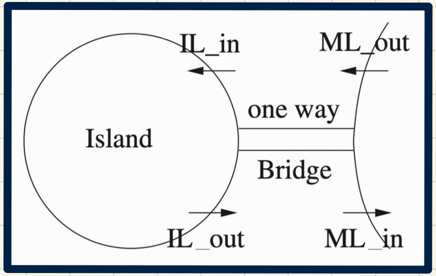
n a, b, c

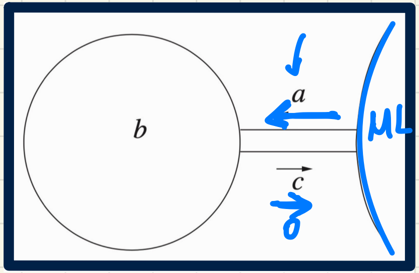
inv1_4: linking abstract & concrete states

inv1_5: bridge is one-way

safety invariant

Bridge Controller: Guards of “old” Events 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 : $a = 0 \lor c = 0$

② $\therefore n \le d$

① 0

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

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

abstract: BAP: $n' = n + 1$ ✓

BAP: $a' = a + 1$

$b' = b$

$c' = c$

G1: $c = 0$ ✓

G2: $a + b < d$

Post-state
$n' \le d$
$a' + b' + c' = n'$
$c = 0$

$n + 1 \le d$

$(a + 1) + b + 0 = n + 1$

$\therefore n < d$

$a + b = n < d$

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

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

G1: $c > 0$

$n \le d$ not relevant

unnecessary: $\therefore a = 0$

$\Rightarrow a = 0$

inv1_5: $a = 0 \lor c = 0$
G1: $c > 0$

Bridge Controller: Abstract vs. Concrete State Transitions

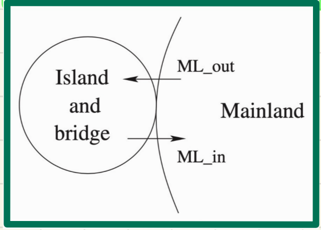
Abstract m0

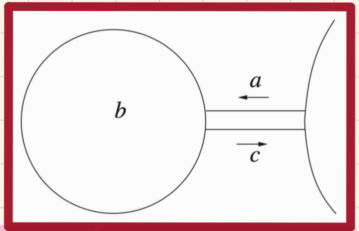
variables: n

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

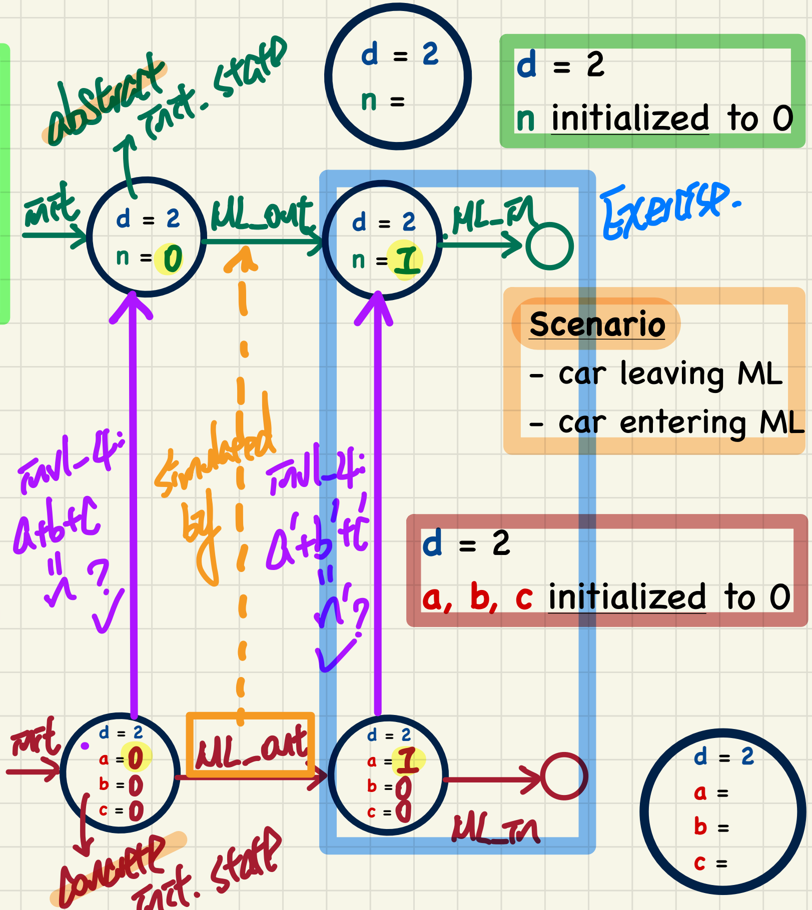
invariants involving both abs. & con. variables

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 : $a = 0 \lor c = 0$

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

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



d = 2
n initialized to 0

Scenario
- car leaving ML
- car entering ML

d = 2
a, b, c initialized to 0

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

actions

post-state value

becomes

evaluation of pre state value

as if: $a := a \quad b := b$

- Pre-State
- Post-State
- Sate Transition

States, Invariants, Events: Abstract vs. Concrete

Abstract m0

abstract version

abs. variables (abstract state)

```
variables:   n
```

```
invariants:
  inv0_1 : n ∈ ℕ
  inv0_2 : n ≤ d
```

abs. invariants (involving abs. state only)

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

abs. state

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

```
constants:   d
```

```
axioms:
  axm0_1 : d ∈ ℕ
  axm0_2 : d > 0
```

Concrete m1

concrete version

concrete variables (concrete state)

```
variables:   a, b, c
```

```
invariants:
  inv1_1 : a ∈ ℕ
  inv1_2 : b ∈ ℕ
  inv1_3 : c ∈ ℕ
  inv1_4 : a + b + c = n
  inv1_5 : a = 0 ∨ c = 0
```

special kind of con. inv. involving both state spaces

concrete invariants (involving at least con. vars.)

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

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

con. state

PO Rule of Invariant Preservation in Refinement: Components

Abstract m0

abs. guards

```
variables: n

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

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

ML_in
    when
        $n > 0$
    then
        $n := n - 1$
    end
```

v

Concrete m1

concrete guards

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

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

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

w

v and v′: abstract variables in pre-/post-states

w and w′: concrete variables in pre-/post-states

G(c, v): an abstract event's guards

H(c, w): a concrete event's guards

I(c, v): list of abstract invariants

J(c, v, w): list of concrete invariants

abs. variables

concrete vars.

E(c, v): an abstract event's effect

F(c, w): a concrete event's effect

$E(c, v)$ of ML_out: $\langle n+1 \rangle$

$F(c, \underline{w})$ of ML_out: $\langle a+1, b, c \rangle$

Lecture 2

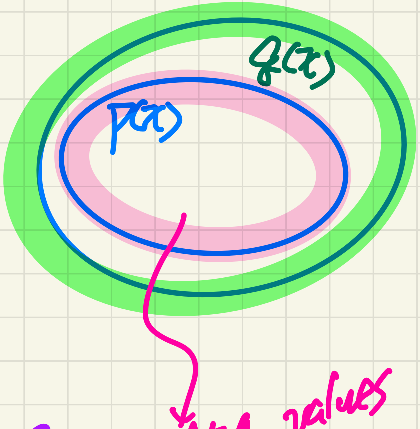
Part G

Case Study on Reactive Systems - Bridge Controller First Refinement: Guard Strengthening

satisfying values

$P \Rightarrow q$

$\{x \mid P(x)\} \subseteq \{x \mid q(x)\}$



satisfying values of a stronger predicate

"P is stronger than q"

"q is weaker than P"

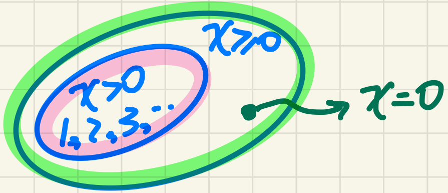
$x > 0$

$x \geq 0$

$x > 0$ is stronger than $x \geq 0$

$x \geq 0$ is weaker than $x > 0$

$$\boxed{x > 0 \Rightarrow x \geq 0}$$



PO/VC Rule of Guard Strengthening: Sequents

Abstract m0

variables: n

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

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

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

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

$$A(c) \quad I(c, v) \quad J(c, v, w) \quad H(c, w) \quad \vdash \quad G_i(c, v)$$

$A(c)$ → event-independent

$I(c, v)$ → abs. inv.

$J(c, v, w)$ → con. inv.

$H(c, w)$ → con. guard

$G_i(c, v)$ → abs. guard

i → single cond.

depends on event under consideration

$d \in \mathbb{N}$	axm0_1
$d > 0$	axm0_2
$n \in \mathbb{N}$	inv0_1
$n \le d$	inv0_2
$a \in \mathbb{N}$	inv1_1
$b \in \mathbb{N}$	inv1_2
$c \in \mathbb{N}$	inv1_3
$a+b+c=n$	inv1_4
$a=0 \lor c=0$	inv1_5

$a+b<d$
$c=0$ } concrete grds of ML_out

ML_out/GRD

$\vdash n<d$ abstract guard of ML_out

Exercise: Formulate ML_in/GRD

② → # abstract guard conditions

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

Discharging POs of m1: Guard Strengthening in Refinement

ML_out/GRD

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

when applying MON IR, guide yourself by the goal to see which hypotheses to drop.

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

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

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

MON

$a+b+c=n$
$a+b<d$
$c=0$
$\vdash$
$n<d$

EQ_LR

$a+b+0=n$
$a+b<d$
$c=0$
$\vdash$
$n<d$

MON

$a+b+0=n$
$a+b<d$
$\vdash$
$n<d$

ARI & arithmetic (basic)

$a+b=n$
$a+b<d$
$\vdash$
$n<d$

EQ_LR, MON

$n<d$
$\vdash$
$n<d$

HYP ✓

Discharging POs of m1: Guard Strengthening in Refinement

ML_in/GRD

$d \in \mathbb{N}$

$d > 0$

$n \in \mathbb{N}$

$n \leq d$

$a \in \mathbb{N}$

$b \in \mathbb{N}$

$c \in \mathbb{N}$

$a + b + c = n$

$a = 0 \lor c = 0$

$c > 0$

$\vdash$

$n > 0$

Con. Inv.

Con. guard

$b \in \mathbb{N}$ $b \geq 0$ 0 1

$n = b + c$ 0 1 1

$c > 0$

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

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

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

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

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

MON

$b \in \mathbb{N}$, $a+b+c=n$, $a=0 \lor c=0$, $c>0$ $\vdash$ $n>0$

OR_L

$b \in \mathbb{N}$, $a+b+c=n$, $a=0$, $c>0$ $\vdash$ $n>0$

EQ_LR, MON

$b \in \mathbb{N}$, $0+b+c=n$, $c>0$ $\vdash$ $n>0$

ARI

$b \in \mathbb{N}$, $b+c=n$, $c>0$ $\vdash$ $n>0$

ARI

$n>0$ $\vdash$ $n>0$ ✓ HYP

$b \in \mathbb{N}$, $a+b+c=n$, $c=0$, $c>0$ $\vdash$ $n>0$

EQ_LR, MON

$0>0$ $\vdash$ $n>0$

ARI

$\bot$ $\vdash$ $n>0$

✓ FALSE_L

Lecture 2

Part H

Case Study on Reactive Systems -
Bridge Controller
First Refinement: Invariant Preservation

PO/VC Rule of Invariant Preservation: Sequents

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
```

BAP: $n' = n+1$

```
ML_in
  when
    $n > 0$
  then
    $n := n - 1$
  end
```

BAP: $n' = n-1$

$$
\begin{array}{l}
A(c) \\
I(c, \mathbf{v}) \\
J(c, \mathbf{v}, \mathbf{w}) \\
✓ H(c, \mathbf{w}) \\
\vdash \\
✓ J_i(c, \boxed{E(c, \mathbf{v})}, \boxed{F(c, \mathbf{w})})
\end{array}
$$

a single concrete inv. cond.

Effect of abs vars.

Effect of con. vars.

Concrete m1

* $\cancel{a'} + \cancel{b'} + \cancel{c'} = n'$

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

** $\cancel{a'} = 0 \vee \cancel{c'} = 0$

$a \qquad (c-1)$

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

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

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

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

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

$2 * 5 = \textcircled{10}$

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

ML_out/inv1_4/INV

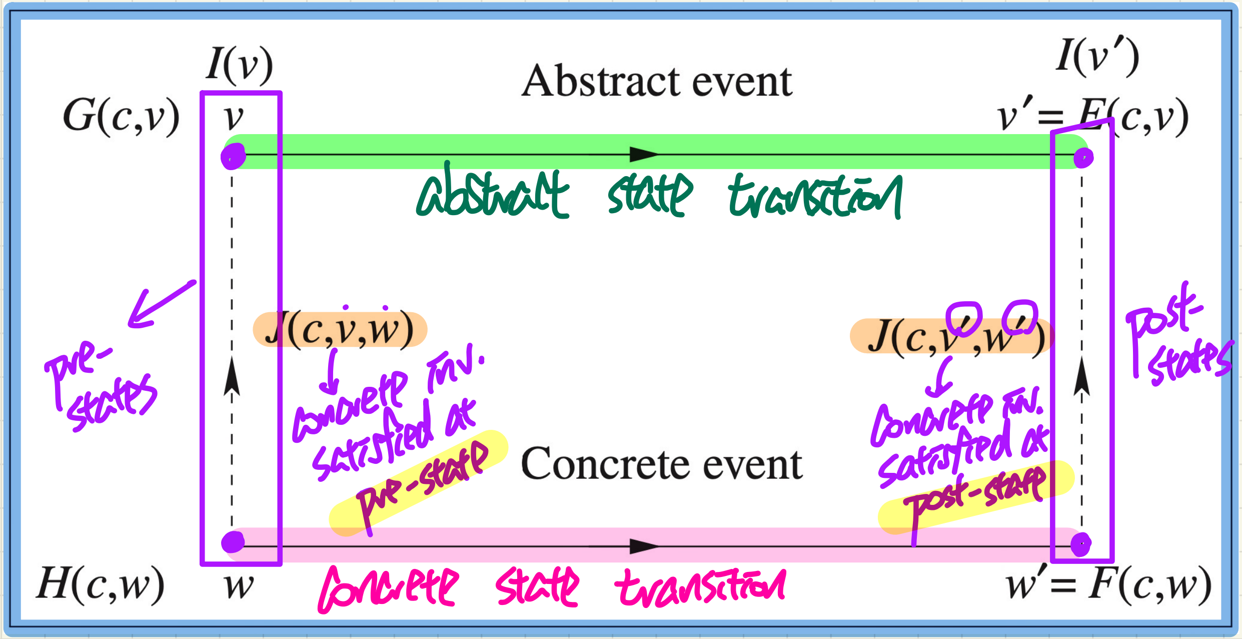
$d \in \mathbb{N}$ axm0_1
$d > 0$ axm0_2
$n \in \mathbb{N}$ inv0_1
$n \leq d$ inv0_2
$a \in \mathbb{N}$ inv1_1
$b \in \mathbb{N}$ inv1_2
$c \in \mathbb{N}$ inv1_3
$a+b+c=n$ inv1_4
$a=0 \vee c=0$ inv1_5
$a+b<d$
$c=0$
$\vdash$ ✓ *
$\boxed{(a+1)+b+c=n+1}$

ML_in/inv1_5/INV

$d \in \mathbb{N}$ axm0_1
$d > 0$ axm0_2
$n \in \mathbb{N}$ inv0_1
$n \leq d$ inv0_2
$a \in \mathbb{N}$ inv1_1
$b \in \mathbb{N}$ inv1_2
$c \in \mathbb{N}$ inv1_3
$a+b+c=n$ inv1_4
$a=0 \vee c=0$ inv1_5
$c>0$
$\vdash$ **
$\boxed{a=0 \vee (c-1)=0}$

Visualizing Invariant Preservation in Refinement

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



Discharging POs of m1: Invariant Preservation in Refinement

ML_out/inv1_4/INV

Exercise

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

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

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

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

Discharging POs of m1: Invariant Preservation in Refinement

ML_in/inv1_5/INV

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

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

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

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

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

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

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

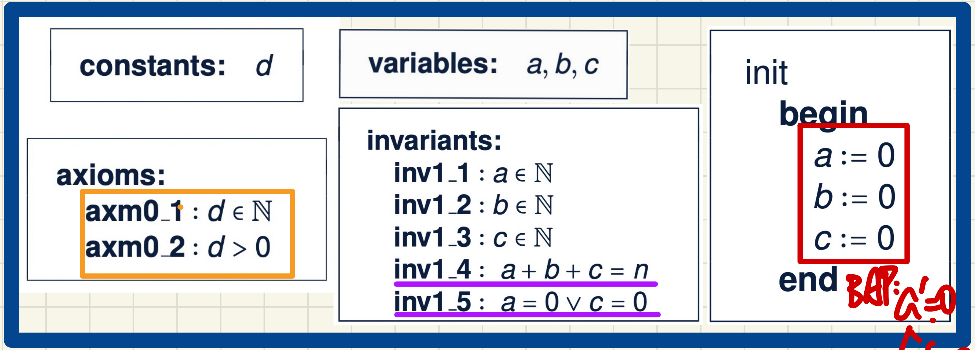
Exercise

Lecture 2

Part I

Case Study on Reactive Systems - 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 : $a = 0 \vee c = 0$

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

BAP: $a' = 0 \wedge b' = 0 \wedge c' = 0$

Components

K(c): effect of **abstract** init

L(c): effect of **concrete** init

$* \; \overset{}{\underset{0}{\cancel{a'}}} + \underset{0}{\cancel{b'}} + \underset{0}{\cancel{c'}} = \cancel{n'}\, 0$

$** \; \underset{0}{\cancel{a'}} = 0 \vee \underset{0}{\cancel{c'}} = 0$

Rule of Invariant Establishment

$$A(c) \vdash J_i(c, K(c), L(c))$$

post-state conc. inv.

conc. inv. cond. ⑤

Exercise: Generate Sequents from the INV rule.

init/inv1_4/INV

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

init/inv1_5/INV

$d \in \mathbb{N}$
$d > 0$
$\vdash **$
$\boxed{0 = 0 \vee 0 = 0}$

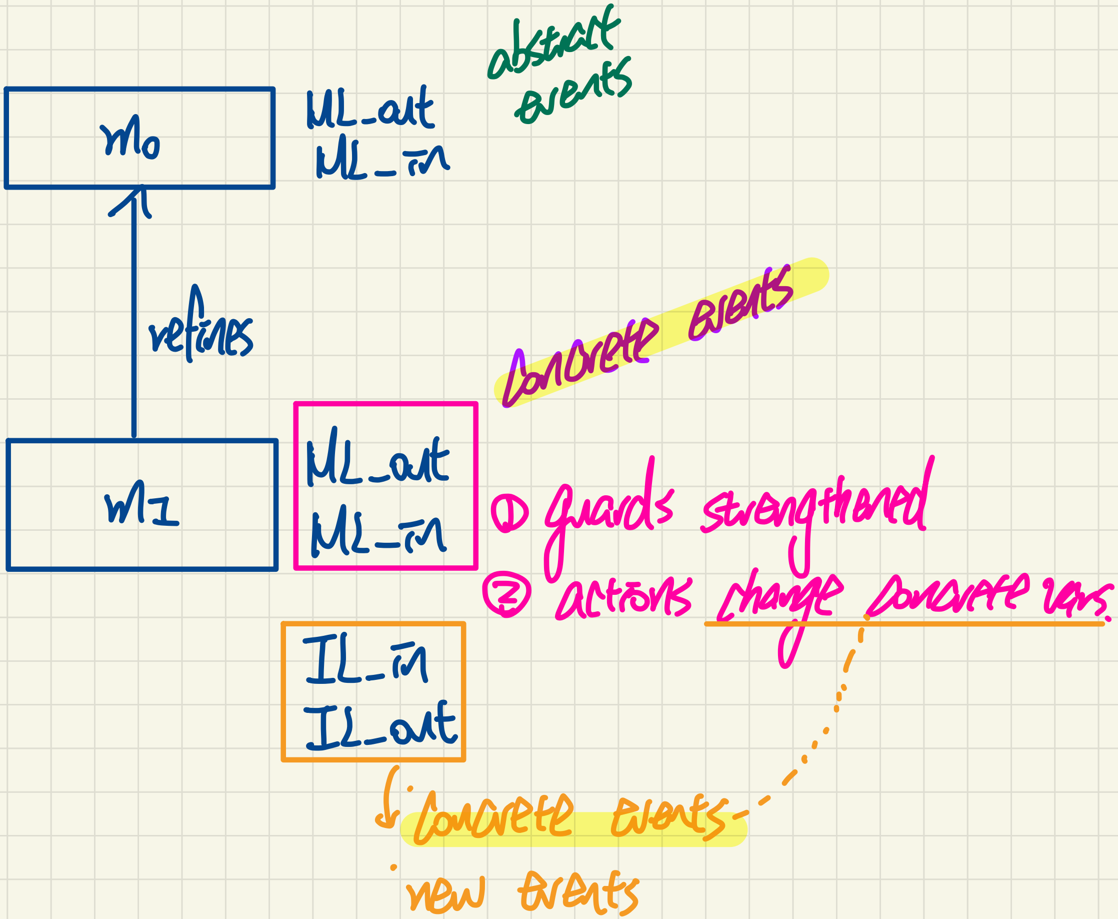
Q. How many PO/VC rules for model m1?

Lecture 2

Part J

Case Study on Reactive Systems - Bridge Controller First Refinement: Invariant Preservation New Events

Events

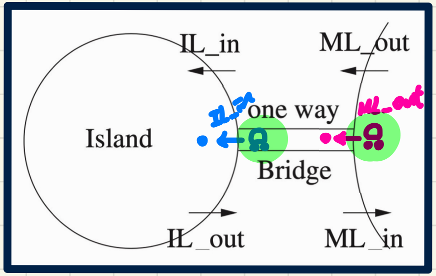


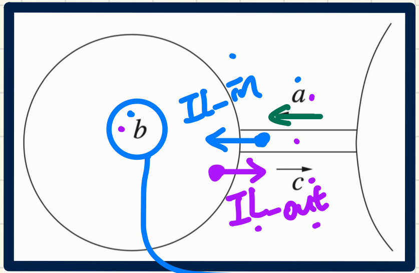
① guards strengthened

② actions change concrete vars.

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

m0 $\boxed{n \leq d \quad \text{IB} \quad \text{ML}}$





IL_in
but b=d
which will
violate: n≤d

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

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

$c = 0$

$\boxed{a + b < d}$? unnecess.

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

① $a' + b' = (a-1) + (b+1) = a + b$

② ML_out earlier for the same car already checked it

constants: d

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

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

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

$b > 0$
$a = 0$

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

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 : $a = 0 \lor c = 0$

Before-After Predicates of Event Actions: 1st Refinement

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

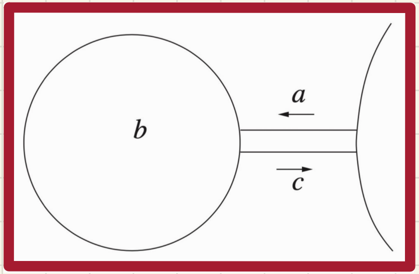
$$a' = a - 1$$
$$\land$$
$$b' = b + 1$$
$$\land$$
$$c' = c$$

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

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

- Pre-State
- Post-State
- Sate Transition

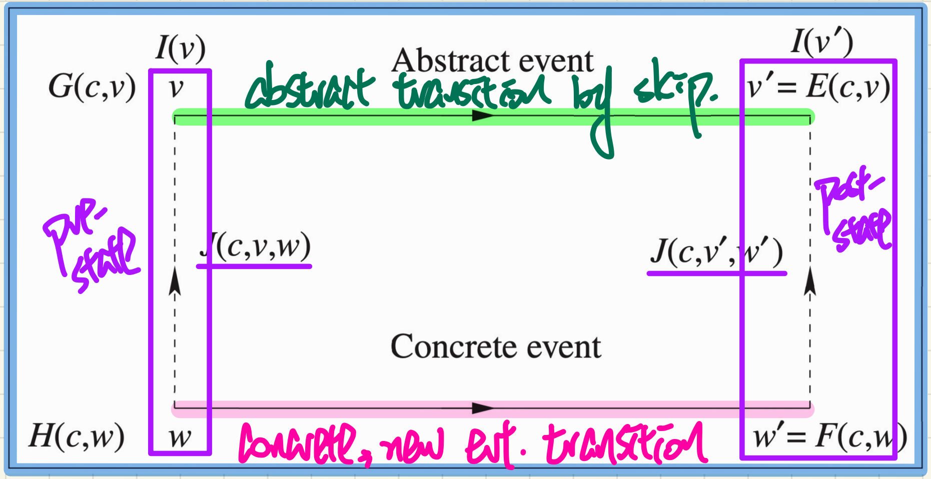
Concrete State Space



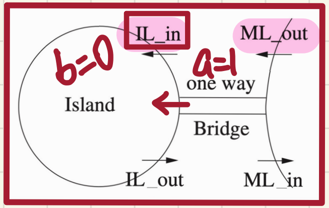
Visualizing Invariant Preservation in Refinement

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

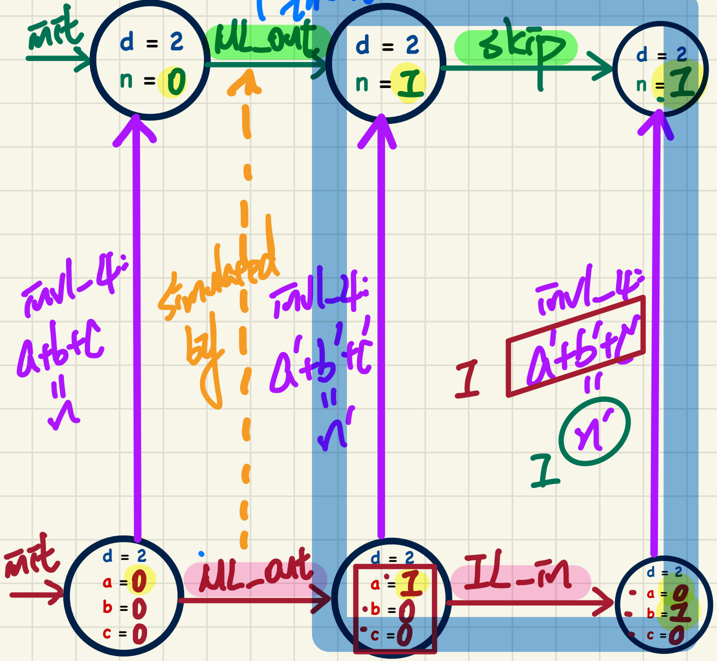
skip
begin
n′=n
end



(IL_in & IL_out)



abs. version simulating the con. version



PO/VC Rule of Invariant Preservation: Sequents

Abstract m0

skip $(n'=n)$

constants: d

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

variables: n

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

$A(c)$

$I(c, v)$ abs. inv.

$J(c, v, w)$ con. inv

$H(c, w)$ con. guard

$\vdash$

$J_i(c, E(c, v), F(c, w))$

effect of skip

effect of new evt.

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 : $a = 0 \lor c = 0$

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

BAP: $a' = a - 1$, $b' = b + 1$, $c' = c$

```
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

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

$$\underset{(a-1)}{\cancel{a'}} + \underset{(b+1)}{\cancel{b'}} + \underset{c}{\cancel{c'}} = \underset{n}{n'}$$

IL_in/INV1_5/INV

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

$$\underset{a-1}{\cancel{a'}} = 0 \lor \underset{c}{\cancel{c'}} = 0$$

Discharging POs of m1: Invariant Preservation in Refinement

IL_in/inv1_4/INV

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

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

$d \in \mathbb{N}$

$d > 0$

$n \in \mathbb{N}$

$n \leq d$

$a \in \mathbb{N}$

$b \in \mathbb{N}$

$c \in \mathbb{N}$

$a + b + c = n$

$a = 0 \lor c = 0$

$a > 0$

$\vdash$

$\boxed{(a - 1) + (b + 1) + c = n}$

MON

$a+b+c = n$

$\vdash$

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

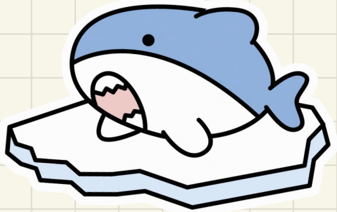
ARI

$\underline{a+b+c = n}$

$\vdash$

$\underline{a+b+c = n}$

HYP



Discharging POs of m1: Invariant Preservation in Refinement

ML_in/inv1_5/INV

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

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

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

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

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

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

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

pre-state satisfaction of inv1_5

post-state satisfaction of inv1_5

MON

$a=0 \vee c=0$
$a>0$
$\vdash$
$(a-1)=0 \vee c=0$

OR_L

$a=0$
$a>0$
$\vdash$
$(a-1)=0 \vee c=0$

EQ_LR, MON

$0>0$
$\vdash$
$(0-1)=0 \vee c=0$

ARI

$\bot$
$\vdash$
$-1=0 \vee c=0$ ✓

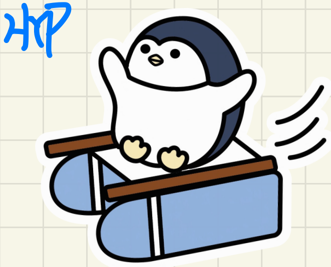
FALSE_L

$c=0$
$a>0$
$\vdash$
$(a-1)=0 \vee c=0$

OR_R2

$c=0$
$a>0$
$\vdash$
$c=0$

HYP ✓



Lecture 2

Part K

*Case Study on Reactive Systems -
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 : $c \in \mathbb{Z}$

"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

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

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

while(true);

incomplete ∵ lacking (1) connection to abs. state (2) safety constraints.

relative to

→ Shockingly, this model can be proved correct w.r.t. inv. preservation.

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

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

indefinitely, preventing other "old" events.

new events diverge

① not deadlock

② livelock ∵ nothing useful ever done

Use of a Variant to Measure New Events Converging

old events

new events

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

```
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
```

preventing divergence

Exercise

Cont. trace with occ. of ML_out s.t. new occ. of IL_in & IL_out becomes possible.

global exp evaluated after each evt. occurrence

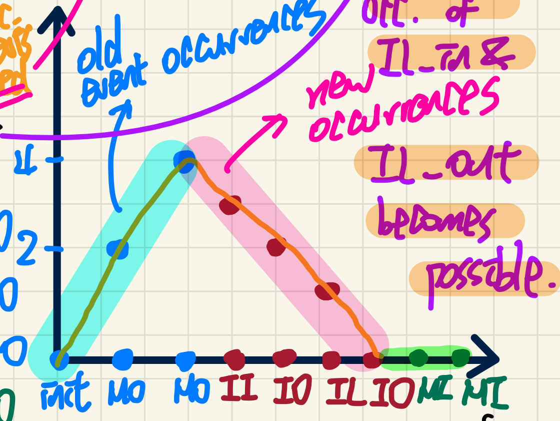
Variants for New Events: $2 \cdot a + b$

no further occ. of new evts allowed

<init, ML_out, ML_out, IL_in, IL_out, IL_in, IL_out, ML_in, ML_in

a = 0	a = 1	a = 2	a = 1	a = 1	a = 0	a = 0	a = 0	a = 0
b = 0	b = 0	b = 0	b = 1	b = 0	b = 1	b = 0	b = 0	b = 0
c = 0	c = 0	c = 0	c = 0	c = 1	c = 1	c = 2	c = 1	c = 0
v = 0	v = 2	v = 4	v = 3	v = 2	v = 1	v = 0	v = 0	v = 0

variant: $2 \cdot a + b$



old event occurrences

new occurrences

occurrences of **concrete** events

PO of Convergence/Non-Divergence/Livelock Freedom

↳ applicable to new events

Variants for New Events: $2 \cdot a + b$

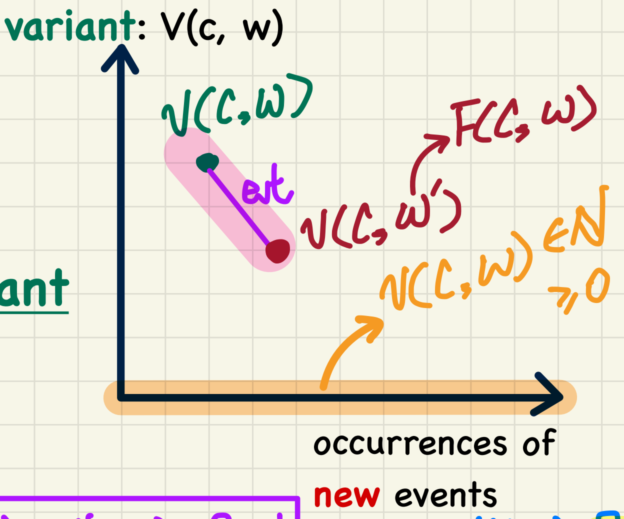
Variant Stays Non-Negative

$$\frac{\begin{array}{l} A(c) \\ I(c,v) \\ J(c,v,w) \\ H(c,w) \\ \vdash \\ V(c,w) \in \mathbb{N} \end{array}}{}\ \text{NAT}$$

IL_in/NAT

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

$\vdash 2 \cdot a + b \in \mathbb{N}$



$V(c,w) = 2 \cdot a + b$

A New Event Occurrence Decreases Variant

evt

$$\frac{\begin{array}{l} A(c) \\ I(c,v) \\ J(c,v,w) \\ H(c,w) \\ \vdash \\ V(c, \boxed{F(c,w)}) < V(c,w) \end{array}}{}\ \text{VAR}$$

effect of con. evt

pre-state

post-state

IL_in/VAR

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

$* \vdash \boxed{2 \cdot (a-1) + (b+1) < 2 \cdot a + b}$

$V(c,w') = 2 \cdot a' + b' = 2 \cdot (a-1) + (b+1) <$

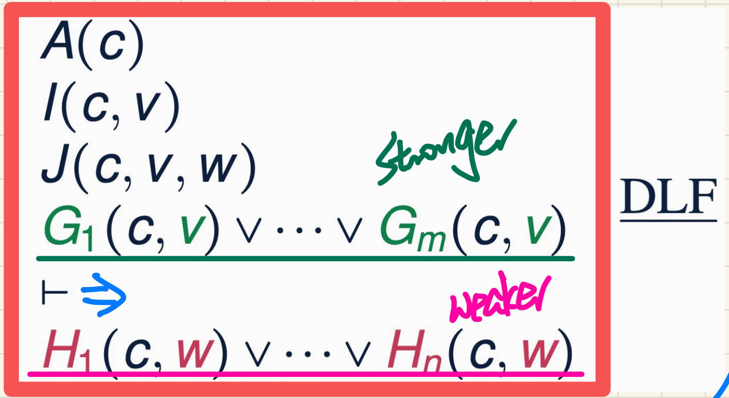
Lecture 2

Part L

Case Study on Reactive Systems - Bridge Controller First Refinement: Relative Deadlock Freedom

Idea of Relative Deadlock Freedom

$\{x \mid P(x)\}$

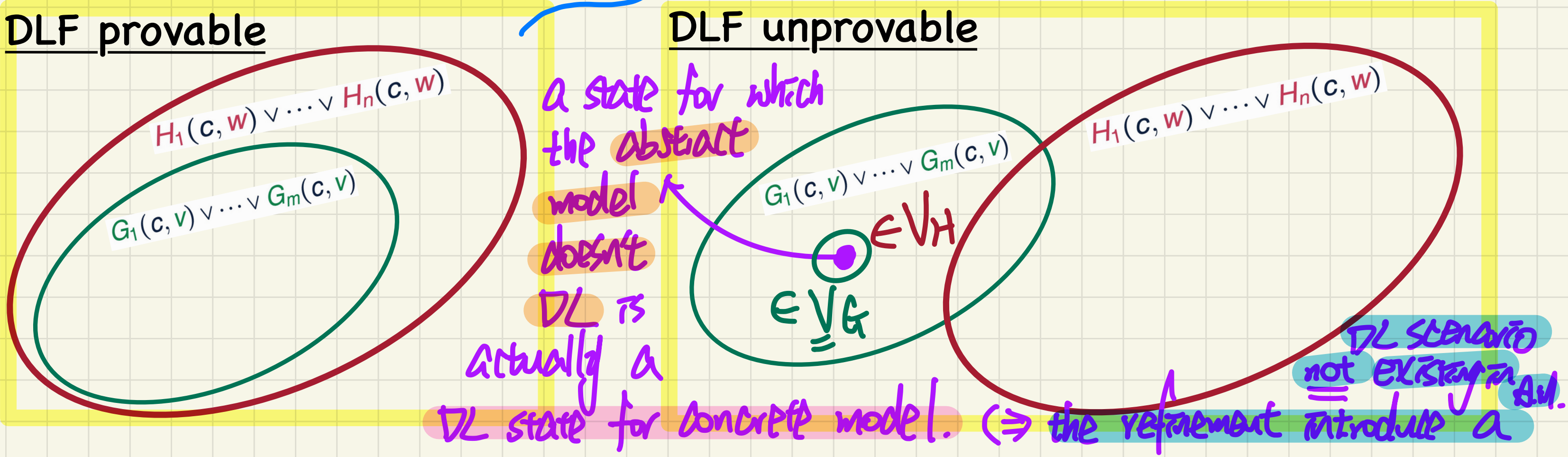


$$\begin{array}{l} A(c) \\ I(c,v) \\ J(c,v,w) \\ G_1(c,v) \vee \cdots \vee G_m(c,v) \\ \vdash \Rightarrow \\ H_1(c,w) \vee \cdots \vee H_n(c,w) \end{array} \quad \text{DLF}$$

stronger

weaker

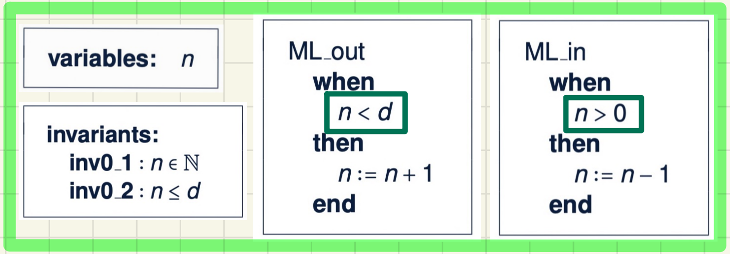
If an abstract state doesn't deadlock, then the corresponding concrete state doesn't DL..


DLF provable
$H_1(c,w) \vee \cdots \vee H_n(c,w)$
$G_1(c,v) \vee \cdots \vee G_m(c,v)$
DLF unprovable
$G_1(c,v) \vee \cdots \vee G_m(c,v)$
$H_1(c,w) \vee \cdots \vee H_n(c,w)$
$\in V_H$
$\in V_G$

a state for which the abstract model doesn't DL is actually a DL state for concrete model. (⇒ the refinement introduce a DL scenario not existing in abs. M.

PO of Relative Deadlock Freedom

✓ Abstract m0



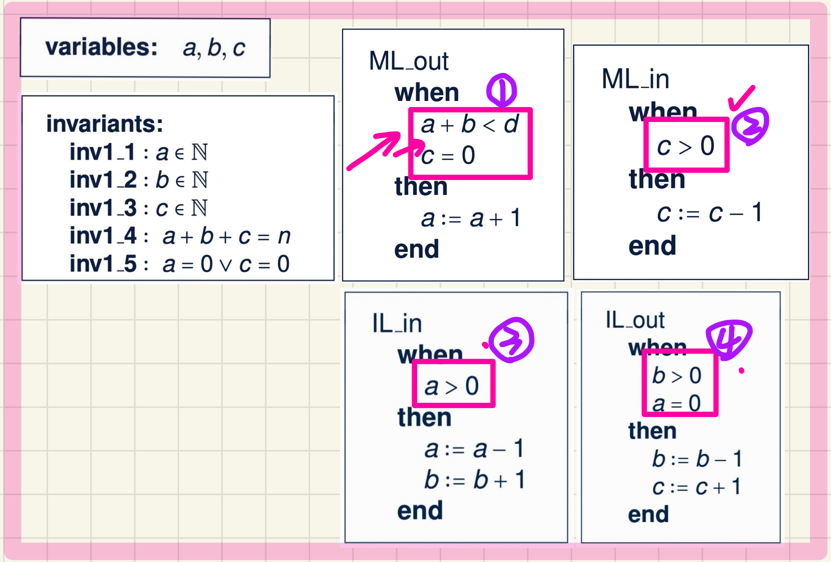
variables: n

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

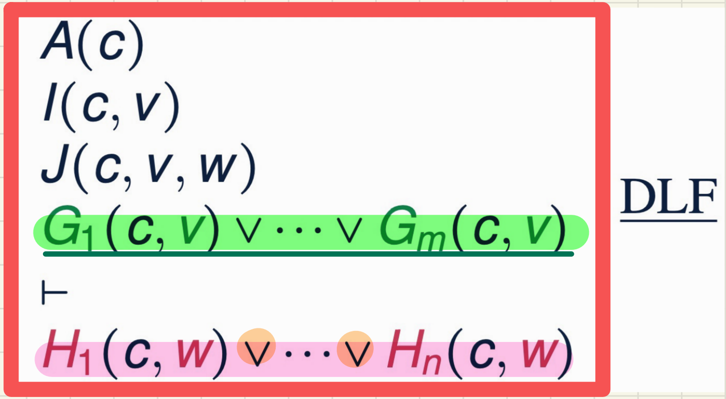
invariants:
inv1_1 : $a \in \mathbb{N}$
inv1_2 : $b \in \mathbb{N}$
inv1_3 : $c \in \mathbb{N}$
inv1_4 : $a + b + c = n$
inv1_5 : $a = 0 \lor c = 0$

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

```
ML_in
  when  ✓ ②
    $c > 0$
  then
    $c := c - 1$
  end
```

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

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



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

DLF

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

$$\vdash \begin{array}{ll} & ①\ (a+b) < d \land c = 0 \\ \lor & c > 0\ ② \\ \lor & a > 0\ ③ \\ \lor & (b > 0) \land (a = 0) \\ & ④ \end{array}$$

Example Inference Rules

$\nearrow \wedge$

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

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

Look Up: OR_L

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

$H \Rightarrow P \vee Q$

$\equiv \{ \text{def. of} \Rightarrow : x \Rightarrow y \equiv \neg x \vee y \}$

$\neg H \vee (P \vee Q)$

$\equiv \{ \text{commutativity} : x \vee (y \vee z) \equiv (x \vee y) \vee z \}$

$\underline{(\neg H \vee P)} \vee Q$

$\equiv \{ \text{double negation} : p \equiv \neg\neg p \}$

$\neg\neg(\underline{\neg H} \vee P) \vee Q$

$\equiv \{ \text{de morgan} : \neg(x \vee y) \equiv \neg x \wedge \neg y \}$, d.n. ✓

$\neg(H \wedge \neg P) \vee Q$

$\equiv \{ \text{def. of} \Rightarrow \}$

$H \wedge \neg P \Rightarrow Q$

Discharging POs of m1: Relative Deadlock Freedom

Part 1

Exercise

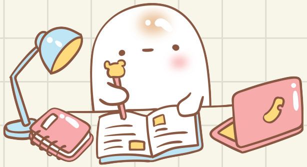
$$\frac{H1 \vdash G}{H1, H2 \vdash G}\ \textbf{MON}$$

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

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

$$\begin{array}{l}
d \in \mathbb{N} \\
d > 0 \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \lor c = 0 \\
n < d \lor n > 0 \\
\vdash \\
\quad\quad a + b < d \land c = 0 \\
\lor \quad c > 0 \\
\lor \quad a > 0 \\
\lor \quad b > 0 \land a = 0
\end{array}$$

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



Discharging POs of m1: Relative Deadlock Freedom

Part 2

Exercise

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

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

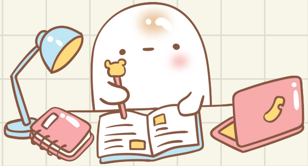
$$\frac{}{P \vdash E = E}\ \textbf{EQ}$$

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

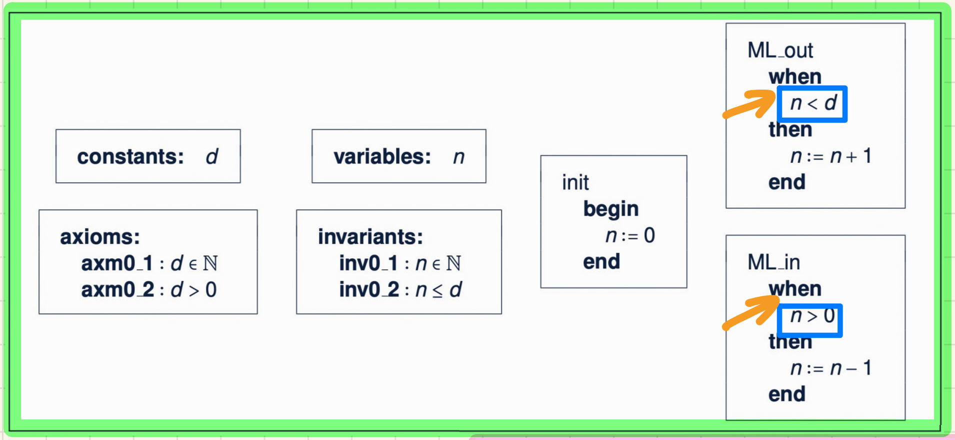
$$\frac{H \vdash Q}{H \vdash P \lor Q}\ \textbf{OR\_R2}$$

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

$$\begin{array}{ll}
d > 0 & \\
b = 0 \lor b > 0 & \\
\vdash & \\
 & b < d \land 0 = 0 \\
\lor & b > 0 \land 0 = 0
\end{array}$$



Initial Model and 1st Refinement: Provably Correct



Abstract m0

constants: d

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

variables: n

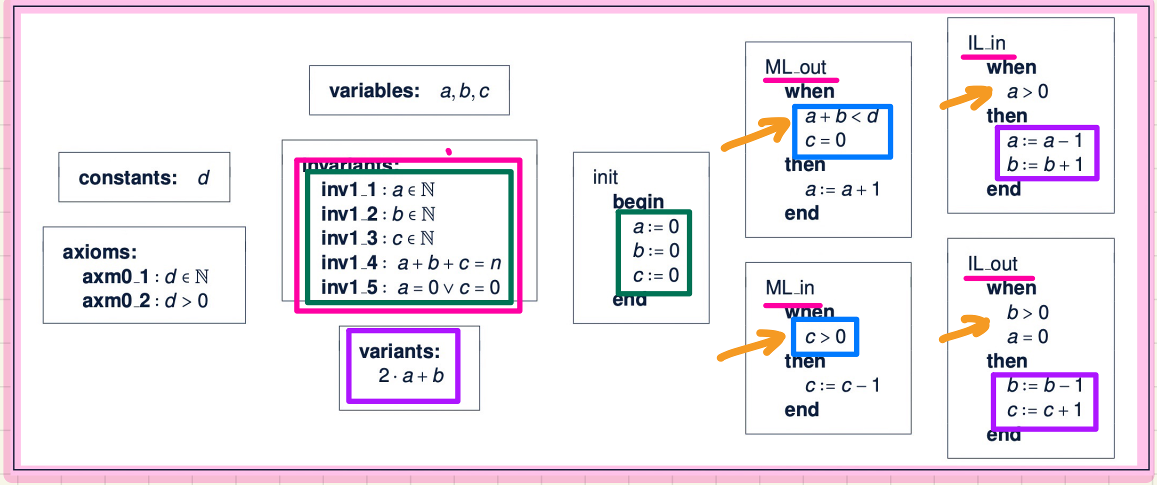
invariants:
inv0_1 : $n \in \mathbb{N}$
inv0_2 : $n \leq d$

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 : $a = 0 \vee c = 0$

variants:
$2 \cdot a + b$

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:

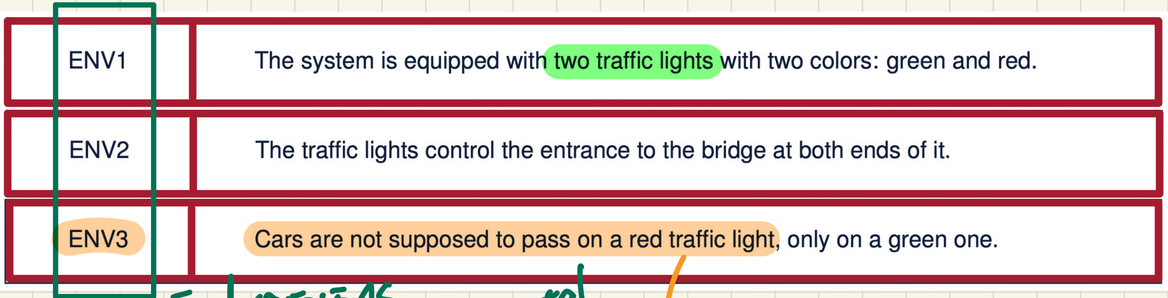
- + Guard Strengthening
- + Invariant Establishment
- + Invariant Preservation
- + Convergence
- + Relative Deadlock Freedom

Lecture 2

Part M

Case Study on Reactive Systems - 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.

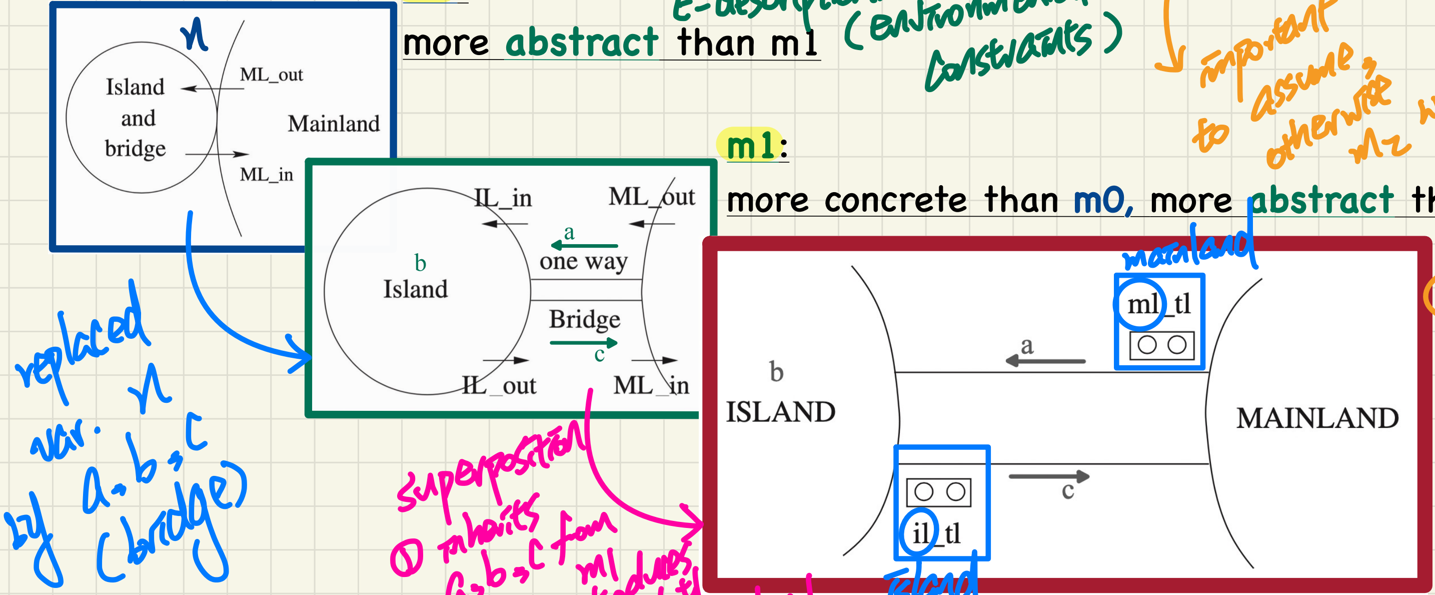
E-descriptions (environmental constraints)

important to assume, otherwise m2 would be much more complicated

m0: more abstract than m1

m1: more concrete than m0, more abstract than m2

m2: more concrete than m1



replaced n var. by a, b, c (bridge)

superposition
① inherits a, b, c from m1
② introduces ml_tl, il_tl

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.

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

** $ml\_tl = green \Rightarrow \boxed{a + b \leq d} \land c = 0$

fixed

$a + b < d$

Dynamic Part of Model

variables:
a, b, c
ml_tl
il_tl

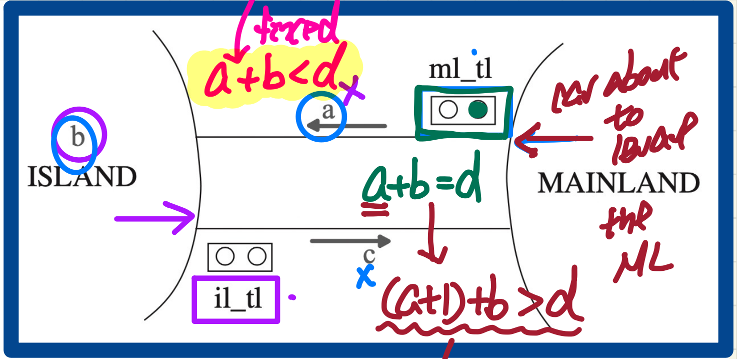
invariants:
inv2_1 : $ml\_tl \in COLOUR$
inv2_2 : $il\_tl \in COLOUR$
inv2_3 : ?? **
inv2_4 : ?? *

Static Part of Model

sets: *COLOR*

constants: *red, green*

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



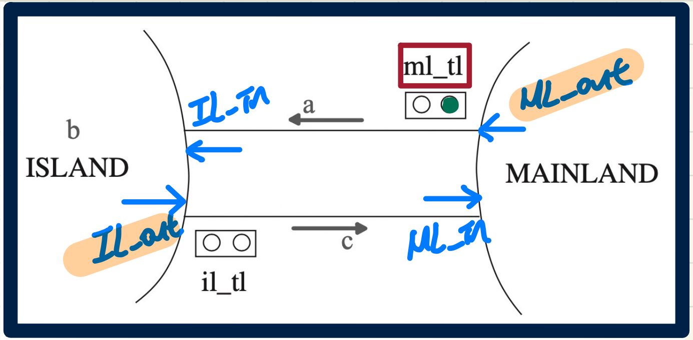
violation of capacity req.

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**).

from driver's perspective

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

$ml\_tl = green$

abstract guards from m1 : $c = 0 \land (a + b < d)$

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

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

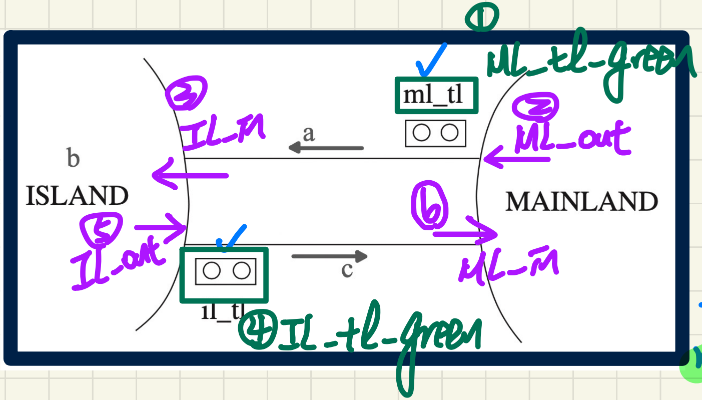
$il\_tl = green$

abstract guards from m1 : $a = 0 \land b > 0$

all these values should not be a driver's concern

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

⟨init, ···, ML_tl_green, ML_out, ···, ··⟩



ML_tl_green:

turn the traffic light **ml_tl** to green

turns ml_tl to green, before a car can exit the ML (ML_out)

ML_tl_green
when
??
then
ml_tl := *green*
end

ml_tl = red

$c=0$

$a+b<d$

$\equiv$ abstract guards of ML_out in m_1

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

IL_tl_green:

turn the traffic light **il_tl** to green

turns IL_tl to green, before a car can exit the IL (IL_out)

IL_tl_green
when
??
then
il_tl := *green*
end

IL_tl = red

$a=0$

$b>0$

$\equiv$ abstract guards of IL_out in m_1

Lecture 2

Part N

Case Study on Reactive Systems - Bridge Controller 2nd Refinement: Invariant Preservation

PO/VC Rule of Invariant Preservation: Sequents

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

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

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

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

post-state version of inv.

Concrete m2

$*\ \underbrace{il\_tl'}_{il\_tl} = green \Rightarrow \underbrace{b'}_{b} > 0 \land \underbrace{a'}_{a+1} = 0$

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** : $\boxed{il\_tl = green \Rightarrow b > 0 \land a = 0}$

```
ML_out
  when
    $ml\_tl = green$
  then
    $a := a + 1$
  end
```

BAP: $a' = a+1 \land b' = b \land c' = c \land ml\_tl' = ml\_tl \land il\_tl' = il\_tl$

```
IL_out
  when
    $il\_tl = green$
  then
    $b := b - 1$
    $c := c + 1$
  end
```

ML_out/inv2_4/INV

axm0_1	$d \in \mathbb{N}$
axm0_2	$d > 0$
axm2_1	$COLOUR = \{green, red\}$
axm2_2	$green \neq red$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
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	$a = 0 \lor c = 0$
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	$il\_tl = green \Rightarrow b > 0 \land a = 0$
Concrete guards of *ML_out*	$ml\_tl = green$
	$\vdash$
Concrete invariant **inv2_4** with *ML_out*'s effect in the post-state	$*\ il\_tl = green \Rightarrow b > 0 \land (a + 1) = 0$

abs. inv.

con. inv.

con. guard of ML_out

Exercise: Specify IL_out/inv2_3/INV

Example Inference Rules

$$\frac{H, P, Q \vdash R}{H, P, \boxed{P \Rightarrow Q} \vdash R}\ \textbf{IMP\_L}$$

→ implicative hypothesis

$$\frac{H \overset{\wedge}{,} P \vdash Q}{H \overset{\Rightarrow}{\vdash} \boxed{P \Rightarrow Q}}\ \textbf{IMP\_R}$$

→ implicative goal

$$\frac{H, \neg Q \vdash P}{H, \neg P \overset{\Rightarrow}{\vdash} Q}\ \textbf{NOT\_L}$$

$\neg P \Rightarrow Q$

$\equiv \neg Q \Rightarrow P$

Modus ponens

$(p \Rightarrow q) \wedge p \equiv q$

Shunting

$p \wedge q \Rightarrow r \equiv p \Rightarrow (q \Rightarrow r)$

Contrapositive:

$p \Rightarrow q \equiv \neg q \Rightarrow \neg p$

Discharging POs of m2: Invariant Preservation

First Attempt

$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \vee c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
$il\_tl = green \Rightarrow b > 0 \wedge a = 0$
$ml\_tl = green$
$\vdash$
$il\_tl = green \Rightarrow b > 0 \wedge (a + 1) = 0$

ML_out/inv2_4/INV

Outstanding Sequent

$green \neq red$
$ml\_tl = green$
$il\_tl = green$
$\vdash$
$1 = 0$

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

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

$$\frac{H, P, Q \vdash R}{H, P, P \Rightarrow Q \vdash R}\ \text{IMP\_L}$$

$$\frac{H, P \vdash Q}{H \vdash P \Rightarrow Q}\ \text{IMP\_R}$$

MON

$green \neq red$
$il\_tl = green \Rightarrow b > 0 \wedge a = 0$
$ml\_tl = green$
$\vdash$
$il\_tl = green \Rightarrow b > 0 \wedge (a + 1) = 0$

IMP_R

$green \neq red$
$il\_tl = green \Rightarrow b > 0 \wedge a = 0$
$ml\_tl = green$
$il\_tl = green$
$\vdash$
$b > 0 \wedge (a + 1) = 0$

IMP_L

$green \neq red$
$b > 0 \wedge a = 0$
$ml\_tl = green$
$il\_tl = green$
$\vdash$
$b > 0 \wedge (a + 1) = 0$

AND_L

$green \neq red$
$b > 0$
$a = 0$
$ml\_tl = green$
$il\_tl = green$
$\vdash$
$b > 0 \wedge (a + 1) = 0$

AND_R

$green \neq red$
$b > 0$
$a = 0$
$ml\_tl = green$
$il\_tl = green$
$\vdash$
$b > 0$

HYP

$green \neq red$
$b > 0$
$a = 0$
$ml\_tl = green$
$il\_tl = green$
$\vdash$
$(a + 1) = 0$

EQ_LR, MON

$green \neq red$
$ml\_tl = green$
$il\_tl = green$
$\vdash$
$(0 + 1) = 0$

ARI

$green \neq red$
$ml\_tl = green$
$il\_tl = green$
$\vdash$
$1 = 0$

??

Discharging POs of m2: Invariant Preservation

First Attempt

IL_out/inv2_3/INV

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

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

$$\frac{H, P, Q \vdash R}{H, P, P \Rightarrow Q \vdash R}\ \textbf{IMP\_L}$$

$$\frac{H, P \vdash Q}{H \vdash P \Rightarrow Q}\ \textbf{IMP\_R}$$

$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$il\_tl = green$
$\vdash$
$ml\_tl = green \Rightarrow a + (b-1) < d \land (c+1) = 0$

MON

$green \neq red$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green$
$\vdash$
$ml\_tl = green \Rightarrow a + (b-1) < d \land (c+1) = 0$

IMP_R

$green \neq red$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green$
$ml\_tl = green$
$\vdash$
$a + (b-1) < d \land (c+1) = 0$

IMP_L

$green \neq red$
$a + b < d \land c = 0$
$il\_tl = green$
$ml\_tl = green$
$\vdash$
$a + (b-1) < d \land (c+1) = 0$

AND_L

$green \neq red$
$a + b < d$
$c = 0$
$il\_tl = green$
$ml\_tl = green$
$\vdash$
$a + (b-1) < d \land (c+1) = 0$

AND_R

$green \neq red$
$a + b < d$
$c = 0$
$il\_tl = green$
$ml\_tl = green$
$\vdash$
$a + (b-1) < d$

MON

$a + b < d$
$\vdash$
$a + (b-1) < d$

ARI

$green \neq red$
$a + b < d$
$c = 0$
$il\_tl = green$
$ml\_tl = green$
$\vdash$
$(c+1) = 0$

EQ_LR, MON

$green \neq red$
$il\_tl = green$
$ml\_tl = green$
$\vdash$
$(0 + 1) = 0$

ARI

$green \neq red$
$il\_tl = green$
$ml\_tl = green$
$\vdash$
$1 = 0$

??

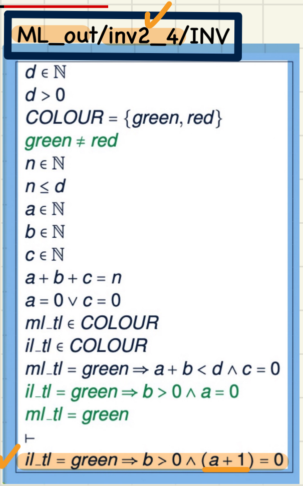
Understanding the Failed Proof on INV

variables:
a, b, c
ml_tl
il_tl

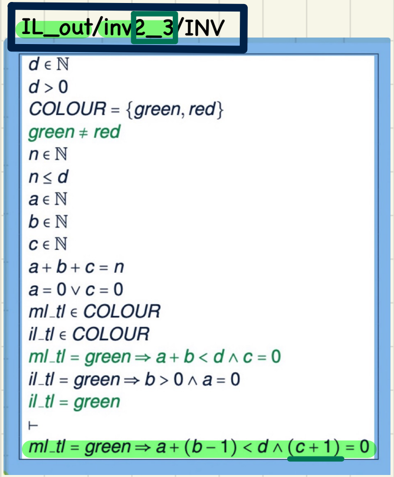
invariants:
inv2_1 : $ml\_tl \in COLOUR$
inv2_2 : $il\_tl \in COLOUR$
inv2_3 : $ml\_tl = green \Rightarrow a + b < d \land \underline{c = 0}$
inv2_4 : $\underline{il\_tl = green} \Rightarrow b > 0 \land \underline{a = 0}$

```
ML_out
   when
      ml_tl = green
   then
      a := a + 1
   end
```

```
IL_out
   when
      il_tl = green
   then
      b := b − 1
      c := c + 1
   end
```


ML_out/inv2_4/INV ✓

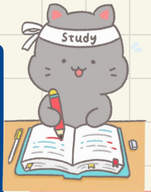
$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$ml\_tl = green$
$\vdash$
✓ $il\_tl = green \Rightarrow b > 0 \land (\underline{a + 1}) = 0$

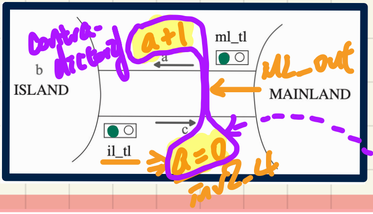

IL_out/inv2_3/INV

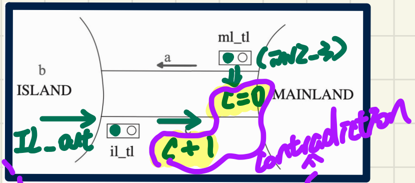
$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$il\_tl = green$
$\vdash$
$ml\_tl = green \Rightarrow a + (b - 1) < d \land (\underline{c + 1}) = 0$

Unprovable Sequent:

$$
\begin{array}{ll}
 & green \neq red \\
\land & il\_tl = green \\
\land & ml\_tl = green \\
\vdash & \\
 & 1 = 0
\end{array}
$$







⟨ $\underbrace{init}$,	$\underbrace{ML\_tl\_green}$,	$\underbrace{ML\_out}$,	$\underbrace{IL\_in}$,	$\underbrace{IL\_tl\_green}$,	$\underbrace{IL\_out}$,	$\underbrace{ML\_out}$ ⟩
$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$
$a' = 0$	$a' = 0$	$\mathbf{a' = 1}$	$\mathbf{a' = 0}$	$a' = 0$	$a' = 0$	$\mathbf{a' = 1}$
$b' = 0$	$b' = 0$	$b' = 0$	$\mathbf{b' = 1}$	$b' = 1$	$\mathbf{b' = 0}$	$b' = 0$
$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$\mathbf{c' = 1}$	$c' = 1$
$ml\_tl' = red$	$\mathbf{ml\_tl' = green}$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = green$
$il\_tl' = red$	$il\_tl' = red$	$il\_tl' = red$	$il\_tl' = red$	$\mathbf{il\_tl' = green}$	$il\_tl' = green$	$il\_tl' = green$

Lecture 2

Part 0

Case Study on Reactive Systems - Bridge Controller
2nd Refinement: Fixing the Model
Adding an Invariant

Fixing m2: Adding an Invariant

REQ3	The bridge is one-way or the other, not both at the same time.

inv2_5 : $ml\_tl = red \lor il\_tl = red$

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

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

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

Concrete m2

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
  when
    $ml\_tl = green$
  then
    $a := a + 1$
  end
```

```
IL_out
  when
    $il\_tl = green$
  then
    $b := b - 1$
    $c := c + 1$
  end
```

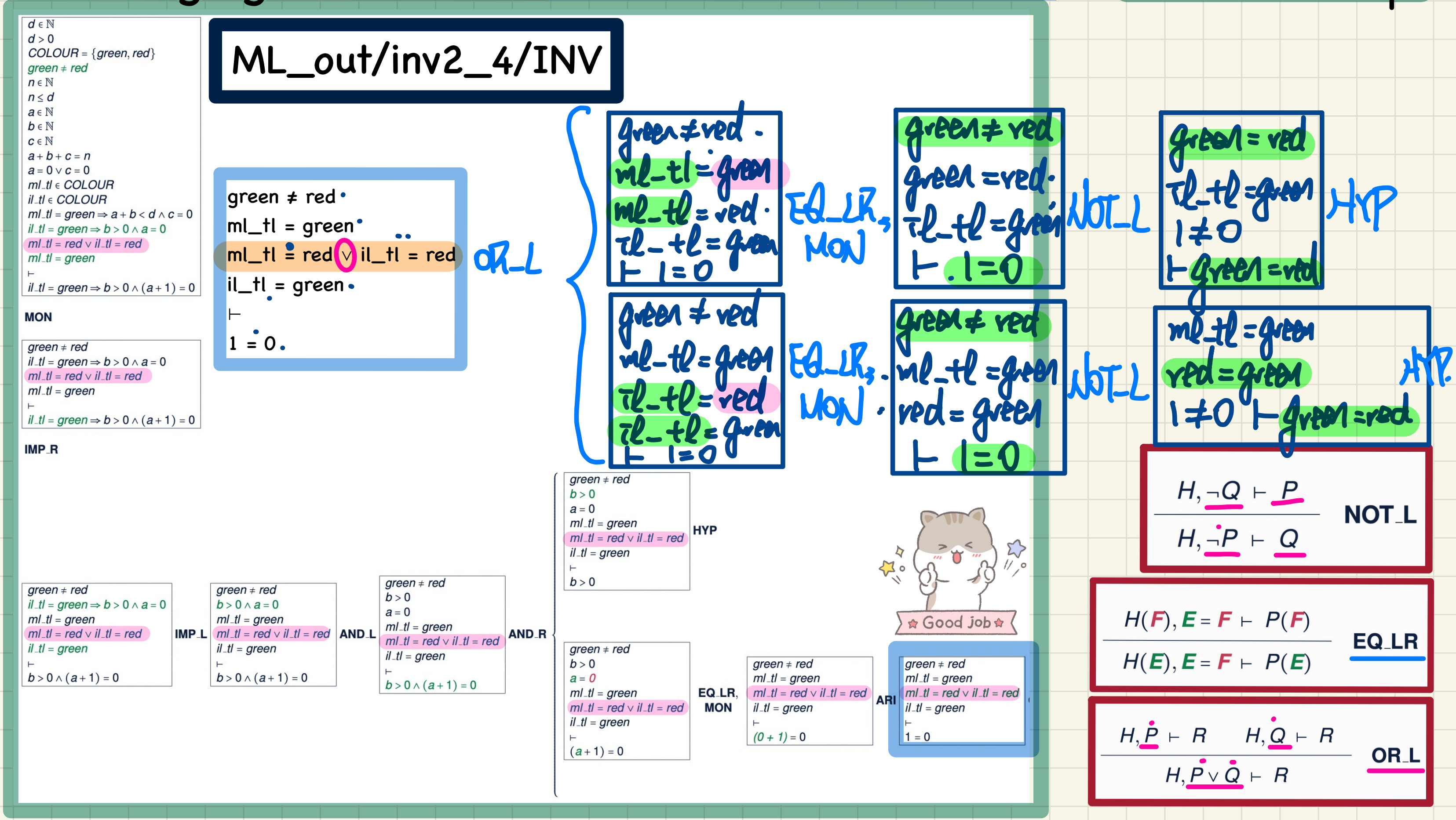
ML_out/inv2_4/INV

axm0_1	$d \in \mathbb{N}$
axm0_2	$d > 0$
axm2_1	$COLOUR = \{green, red\}$
axm2_2	$green \neq red$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
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	$a = 0 \lor c = 0$
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	$il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5	$ml\_tl = red \lor il\_tl = red$
Concrete guards of *ML_out*	$ml\_tl = green$
	$\vdash$
Concrete invariant **inv2_4** with *ML_out*'s effect in the post-state	$il\_tl = green \Rightarrow b > 0 \land (a + 1) = 0$

Exercise: Specify IL_out/inv2_3/INV

Discharging POs of m2: Invariant Preservation

Second Attempt



ML_out/inv2_4/INV

$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$il\_tl = green \Rightarrow b > 0 \land (a + 1) = 0$

MON

$green \neq red$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$il\_tl = green \Rightarrow b > 0 \land (a + 1) = 0$

IMP_R

$green \neq red$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$ml\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$b > 0 \land (a + 1) = 0$

IMP_L

$green \neq red$
$b > 0 \land a = 0$
$ml\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$b > 0 \land (a + 1) = 0$

AND_L

$green \neq red$
$b > 0$
$a = 0$
$ml\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$b > 0 \land (a + 1) = 0$

AND_R

$green \neq red$
$b > 0$
$a = 0$
$ml\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$b > 0$

HYP

$green \neq red$
$b > 0$
$a = 0$
$ml\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$(a + 1) = 0$

EQ_LR, MON

$green \neq red$
$ml\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$(0 + 1) = 0$

ARI

$green \neq red$
$ml\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$1 = 0$

Good job

$green \neq red$
$ml\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$1 = 0$

OR_L

$green \neq red$
$ml\_tl = green$
$ml\_tl = red$
$il\_tl = green$
$\vdash 1 = 0$

EQ_LR, MON

$green \neq red$
$green = red$
$il\_tl = green$
$\vdash 1 = 0$

NOT_L

$green = red$
$il\_tl = green$
$1 \neq 0$
$\vdash green = red$

HYP

$green \neq red$
$ml\_tl = green$
$il\_tl = red$
$il\_tl = green$
$\vdash 1 = 0$

EQ_LR, MON

$green \neq red$
$ml\_tl = green$
$red = green$
$\vdash 1 = 0$

NOT_L

$ml\_tl = green$
$red = green$
$1 \neq 0 \vdash green = red$

HYP

$$\frac{H, \neg Q \vdash P}{H, \neg P \vdash Q} \quad \text{NOT\_L}$$

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

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

Discharging POs of m2: Invariant Preservation

Second Attempt

IL_out/inv2_3/INV

$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$ml\_tl = green \Rightarrow a + (b - 1) < d \land (c + 1) = 0$

MON

$green \neq red$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$ml\_tl = red \lor il\_tl = red$
$il\_tl = green$
$\vdash$
$ml\_tl = green \Rightarrow a + (b - 1) < d \land (c + 1) = 0$

IMP_R

$green \neq red$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$a + (b - 1) < d \land (c + 1) = 0$

IMP_L

$green \neq red$
$a + b < d \land c = 0$
$il\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$a + (b - 1) < d \land (c + 1) = 0$

AND_L

$green \neq red$
$a + b < d$
$c = 0$
$il\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$a + (b - 1) < d \land (c + 1) = 0$

AND_R

$green \neq red$
$a + b < d$
$c = 0$
$il\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$a + (b - 1) < d$

MON

$a + b < d$
$\vdash$
$a + (b - 1) < d$

ARI

$green \neq red$
$a + b < d$
$c = 0$
$il\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$(c + 1) = 0$

EQ_LR, MON

$green \neq red$
$il\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$(0 + 1) = 0$

ARI

$green \neq red$
$il\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$1 = 0$

$green \neq red$
$il\_tl = green$
$ml\_tl = red \lor il\_tl = red$
$ml\_tl = green$
$\vdash$
$1 = 0$

Assignment

$$\frac{H, \neg Q \vdash P}{H, \neg P \vdash Q} \quad \textbf{NOT\_L}$$

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

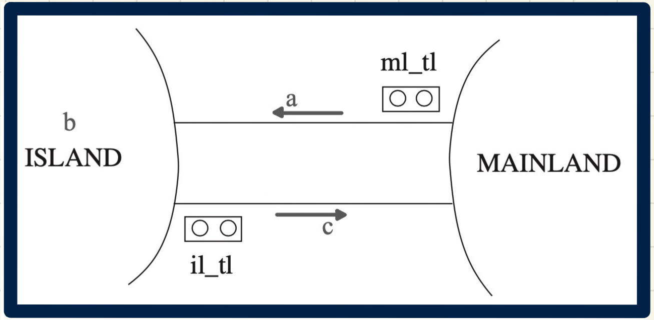
$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, P \lor Q \vdash R} \quad \textbf{OR\_L}$$

Lecture 2

Part P

Case Study on Reactive Systems - Bridge Controller
2nd Refinement: Fixing the Model
Adding Actions

Fixing m2: Adding Actions



```
ML_tl_green
  when
    ml_tl = red
    a + b < d
    c = 0
  then
    ml_tl := green
    il_tl := red
  end
```

$ml\_tl' = \underset{\#}{g}$ $\land$ $il\_tl' = \underset{=}{r} \land a' = a \land b' = b \land c' = c$

```
IL_tl_green
  when
    il_tl = red
    b > 0
    a = 0
  then
    il_tl := green
    ml_tl := red
  end
```

ML_tl_green/inv2_5/INV

axm0_1	$d \in \mathbb{N}$
axm0_2	$d > 0$
axm2_1	$COLOUR = \{green, red\}$
axm2_2	$green \neq red$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
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	$a = 0 \lor c = 0$
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	$il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5	$ml\_tl = red \lor il\_tl = red$

Concrete guards

$ml\_tl = red$

$a + b < d$

$c = 0$

$\vdash$

* $\boxed{green = red \lor red = red}$

* $\overset{g}{ml\_tl'} = red \lor \overset{r}{il\_tl'} = red$

Exercise: Proof

Exercise: Specify IL_tl_green/inv2_5/INV

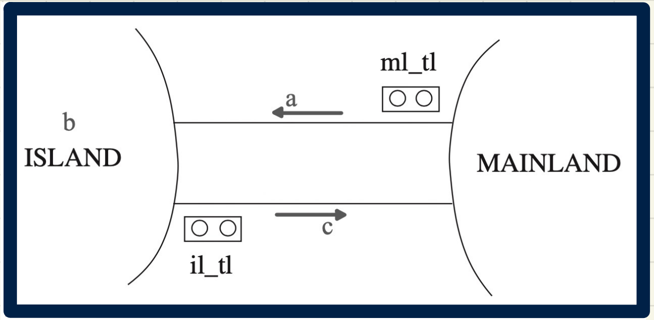
Lecture 2

Part Q

Case Study on Reactive Systems - Bridge Controller 2nd Refinement: Fixing the Model Splitting Events

Invariant Preservation: ML_out/inv2_3/INV

ML_out/inv2_4 discussed earlier



variables:
a, b, c
ml_tl
il_tl

```
ML_out
  when
    $ml\_tl = green$
  then
    $a := a + 1$
  end
```

```
IL_out
  when
    $il\_tl = green$
  then
    $b := b - 1$
    $c := c + 1$
  end
```

invariants:

inv2_1 : $ml\_tl \in COLOUR$

inv2_2 : $il\_tl \in COLOUR$

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

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

ML_out/inv2_3/INV

axm0_1	$\{ d \in \mathbb{N}$
axm0_2	$\{ d > 0$
axm2_1	$\{ COLOUR = \{green, red\}$
axm2_2	$\{ green \neq red$
inv0_1	$\{ n \in \mathbb{N}$
inv0_2	$\{ n \leq d$
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	$\{ a = 0 \vee c = 0$
inv2_1	$\{ ml\_tl \in COLOUR$
inv2_2	$\{ il\_tl \in COLOUR$
inv2_3	$\{ ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
inv2_4	$\{ il\_tl = green \Rightarrow b > 0 \wedge a = 0$
inv2_5	$\{ ml\_tl = red \vee il\_tl = red$
Concrete guards of *ML_out*	$\{ ml\_tl = green$
	$\vdash$
Concrete invariant **inv2_3** with *ML_out*'s effect in the post-state	$\{ ml\_tl = green \Rightarrow (a + 1) + b < d \wedge c = 0$

Exercise: Specify IL_out/inv2_4/INV

IL_out/inv2_3 discussed earlier

Discharging POs of m2: Invariant Preservation

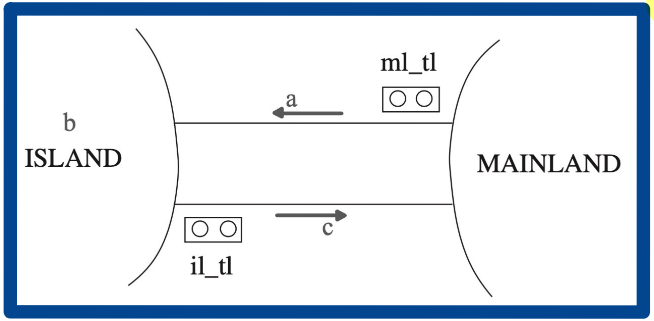
First Attempt

$$
\begin{array}{l}
d \in \mathbb{N} \\
d > 0 \\
COLOUR = \{green, red\} \\
green \neq red \\
n \in \mathbb{N} \\
n \leq d \\
a \in \mathbb{N} \\
b \in \mathbb{N} \\
c \in \mathbb{N} \\
a + b + c = n \\
a = 0 \lor c = 0 \\
ml\_tl \in COLOUR \\
il\_tl \in COLOUR \\
ml\_tl = green \Rightarrow a + b < d \land c = 0 \\
il\_tl = green \Rightarrow b > 0 \land a = 0 \\
ml\_tl = red \lor il\_tl = red \\
ml\_tl = green \\
\vdash \\
ml\_tl = green \Rightarrow (a + 1) + b < d \land c = 0
\end{array}
$$

ML_out/inv2_3/INV → Exercise

IL_out/inv2_4/INV

↳ expected to see: a similar unprovable sequent



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

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

$$\frac{H, P \vdash Q}{H \vdash P \Rightarrow Q}\ \textbf{IMP\_R}$$

MON

$$
\begin{array}{l}
ml\_tl = green \Rightarrow a + b < d \land c = 0 \\
\vdash \\
\boxed{ml\_tl = green} \Rightarrow (a + 1) + b < d \land c = 0
\end{array}
$$

IMP_R

$$
\begin{array}{l}
ml\_tl = green \Rightarrow a + b < d \land c = 0 \\
\boxed{ml\_tl = green}\ ✔ \\
\vdash \\
(a + 1) + b < d \land c = 0
\end{array}
$$

IMP_R

$$
\begin{array}{l}
a + b < d \land c = 0 \\
ml\_tl = green \\
\vdash \\
(a + 1) + b < d \land c = 0
\end{array}
$$

AND_L

$$
\begin{array}{l}
a + b < d \\
c = 0 \\
ml\_tl = green \\
\vdash \\
(a + 1) + b < d \land c = 0
\end{array}
$$

AND_R

$$
\begin{array}{l}
a + b < d \\
c = 0 \\
ml\_tl = green \\
\vdash \\
(a + 1) + b < d
\end{array}
$$

??

$$
\begin{array}{l}
a + b < d \\
c = 0 \\
ml\_tl = green \\
\vdash \\
c = 0
\end{array}
$$

HYP

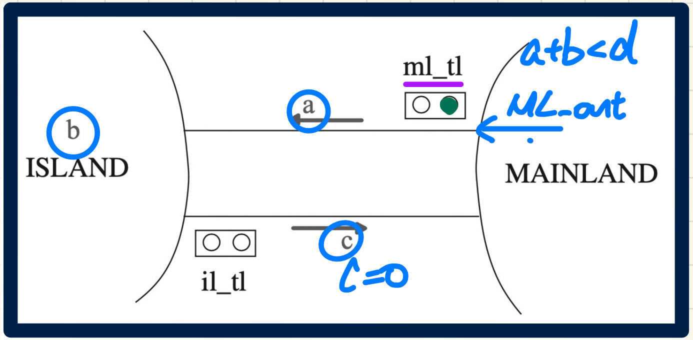
Understanding the Failed Proof on INV

variables:
a, b, c
ml_tl
il_tl

invariants:
inv2_1 : $ml\_tl \in COLOUR$
inv2_2 : $il\_tl \in COLOUR$
inv2_3 : $ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
inv2_4 : $il\_tl = green \Rightarrow b > 0 \wedge a = 0$

```
ML_out
   when
      ml_tl = green
   then
      a := a + 1
   end
```

```
IL_out
   when
      il_tl = green
   then
      b := b − 1
      c := c + 1
   end
```



$$\boxed{\begin{array}{l} x < y \\ \Rightarrow x + 1 < y \end{array}}$$

e.g. $x = 3$
$y = 4$

Unprovable Sequent from ML_out/inv2_3/INV

$$\begin{array}{ll} & a + b < d \\ \wedge & c = 0 \\ \wedge ✓ & ml\_tl = green \\ \vdash & \\ & (a + 1) + b < d \end{array}$$

$d = 3, b = 0, a = 0$	$[\ (a+1) + b < d$ evaluates to ***true*** $]$
$d = 3, b = 1, a = 0$	$[\ (a+1) + b < d$ evaluates to ***true*** $]$
$d = 3, b = 0, a = 1$	$[\ (a+1) + b < d$ evaluates to ***true*** $]$
$d = 3, b = 0, a = 2$	$[\ (a+1) + b < d$ evaluates to ***false*** $]$
$d = 3, b = 1, a = 1$	$[\ (a+1) + b < d$ evaluates to ***false*** $]$
$d = 3, b = 2, a = 0$	$[\ (a+1) + b < d$ evaluates to ***false*** $]$

$(a+1)+b \boxed{\neq d}$ another ML_out allowed

ML_out

$(a+1)+b \boxed{= d}$ no more ML_out allowed $\Rightarrow$ $\boxed{ml\_tl := red}$

inv2_3 is preserved
∵ false ⇒ _ ✓

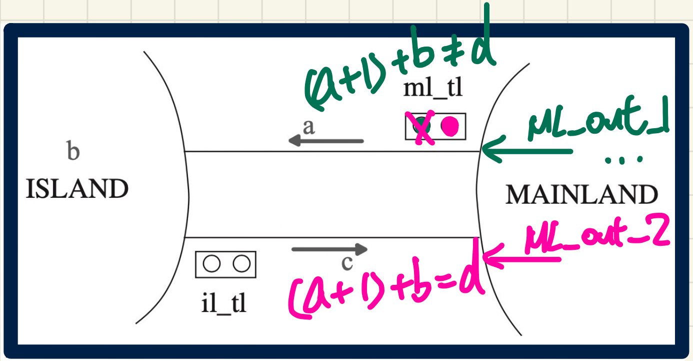
Fixing m2: Splitting Events

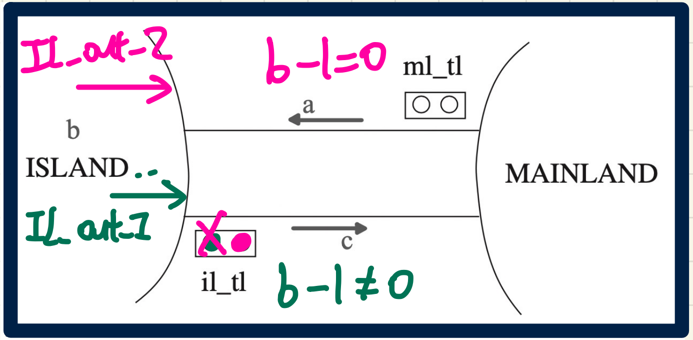
m1: ML_out IL_out

refines

m2: ML_out_1 ML_out_2 IL_out_1 IL_out_2

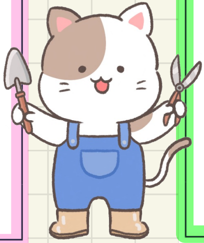
old, concrete events





```
ML_out_1
  when
    $ml\_tl = green$
    $a + b + 1 \neq d$
  then
    $a := a + 1$
  end
```

```
ML_out_2
  when
    $ml\_tl = green$
    $a + b + 1 = d$
  then
    $a := a + 1$
    $ml\_tl := red$
  end
```



```
IL_out_1
  when
    $il\_tl = green$
    $b \neq 1$ ≡ $b-1 \neq 0$
  then
    $b := b - 1$
    $c := c + 1$
  end
```

```
IL_out_2
  when
    $il\_tl = green$
    $b = 1$ ≡ $b-1=0$
  then
    $b := b - 1$
    $c := c + 1$
    $il\_tl := red$
  end
```

6 ↑ 8 : ML_out split
IL_out split

# of sequents for INV:
$8 \times 5 = 40$

Lecture 2

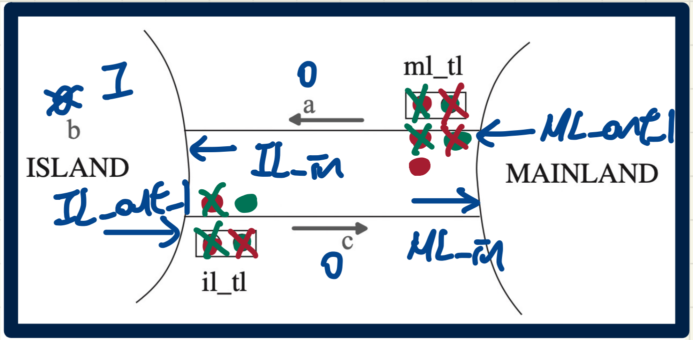
Part R

Case Study on Reactive Systems - Bridge Controller
2nd Refinement: Livelock/Divergence

Current m2 May Livelock

```
ML_tl_green
  when
  ✓ ml_tl = red
  ✓ a + b < d
  ✓ c = 0
  then
    ml_tl := green
    il_tl := red
  end
```

```
IL_tl_green
  when
    il_tl = red
    b > 0
    a = 0
  then
    il_tl := green
    ml_tl := red
  end
```



d=2

Expected trace: no divergence from this trace

<init, ML_tl_green, ML_out_1, IL_in,
a new event
old events
IL_tl_green, IL_out_1, ML_in>

also a valid trace of m2, but leading to livelock

Is ML_tl_g. enabled?

Is IL_tl_g. enabled?

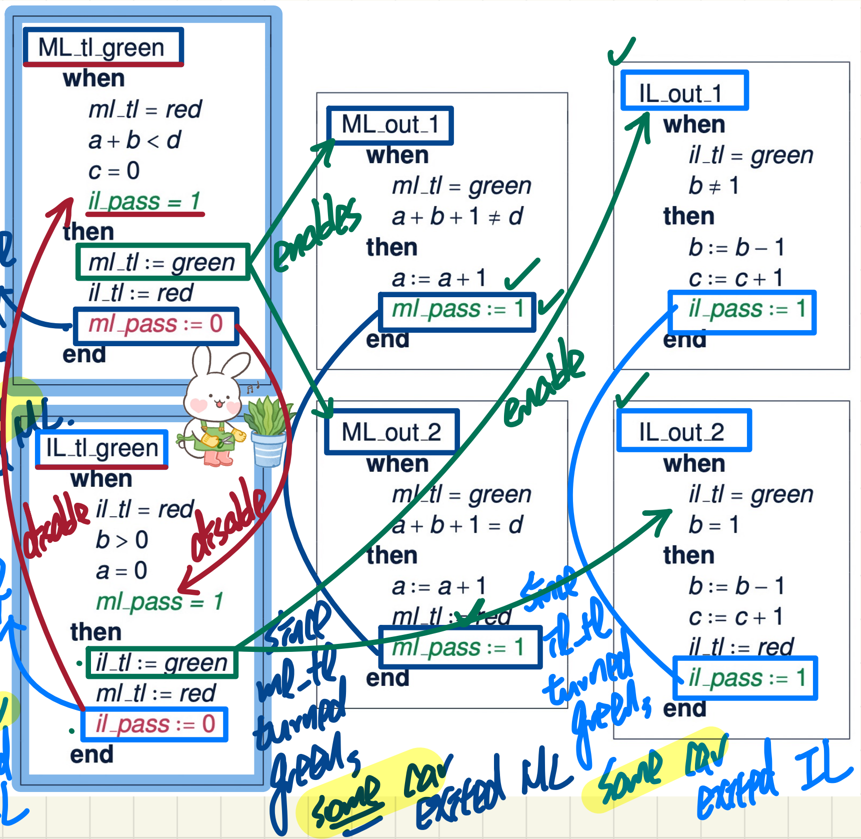
⟨ init ,	ML_tl_green ,	ML_out_1 ,	IL_in ,	IL_tl_green ,	ML_tl_green ,	IL_tl_green , . . . ⟩
$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$
$a' = 0$	$a' = 0$	$a' = 1$	$a' = 0$	$a' = 0$	$a' = 0$	$a' = 0$
$b' = 0$	$b' = 0$	$b' = 0$	$b' = 1$	$b' = 1$	$b' = 1$	$b' = 1$
$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$
$ml\_tl = red$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = red$	$ml\_tl' = green$	$ml\_tl' = red$
$il\_tl = red$	$il\_tl' = red$	$il\_tl' = red$	$il\_tl' = red$	$il\_tl' = green$	$il\_tl' = red$	$il\_tl' = green$

pattern of divergence

Fixing m2: Regulating Traffic Light Changes

To break the divergence pattern, after each new evt occurring, some old events occur.

Divergence Trace: <init, ML_tl_green, ML_out_1, IL_in, IL_tl_green, ML_tl_green, IL_tl_green, ...>



```
ML_tl_green
  when
    ml_tl = red
    a + b < d
    c = 0
    il_pass = 1
  then
    ml_tl := green
    il_tl := red
    ml_pass := 0
  end
```

```
IL_tl_green
  when
    il_tl = red
    b > 0
    a = 0
    ml_pass = 1
  then
    il_tl := green
    ml_tl := red
    il_pass := 0
  end
```

```
ML_out_1
  when
    ml_tl = green
    a + b + 1 ≠ d
  then
    a := a + 1
    ml_pass := 1
  end
```

```
ML_out_2
  when
    ml_tl = green
    a + b + 1 = d
  then
    a := a + 1
    ml_tl := red
    ml_pass := 1
  end
```

```
IL_out_1
  when
    il_tl = green
    b ≠ 1
  then
    b := b − 1
    c := c + 1
    il_pass := 1
  end
```

```
IL_out_2
  when
    il_tl = green
    b = 1
  then
    b := b − 1
    c := c + 1
    il_tl := red
    il_pass := 1
  end
```

since ml-tl turned green, no car exited ML.
since il-tl turned green, no car exited IL
disable
disable
enables
enable
since ml-tl turned green, some car exited ML
since il-tl turned green, some car exited IL
✓
✓
✓
✓

d = 2	ml_pass	il_pass
< init,	1	1
ML_tl_green,	0	1
ML_out_1,	1	1
ML_out_2,	1	1
IL_in,	1	1
IL_in,	1	1
IL_tl_green,	1	0
IL_out_1,	1	1
IL_out_2,	1	1
ML_in,	1	1
ML_in	1	1
>		

∵ ml_tl, il_tl both red

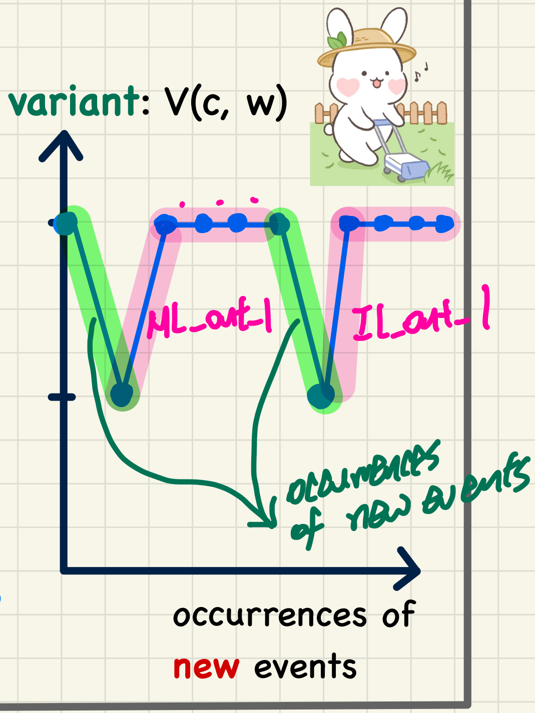
Fixing m2: Measuring Traffic Light Changes

```
ML_tl_green
  when
    ml_tl = red
    a + b < d
    c = 0
    il_pass = 1
  then
    ml_tl := green
    il_tl := red
    ml_pass := 0
  end
```

```
IL_tl_green
  when
    il_tl = red
    b > 0
    a = 0
    ml_pass = 1
  then
    il_tl := green
    ml_tl := red
    il_pass := 0
  end
```

d = 2	ml_pass	il_pass	
< init,	1	1	2
ML_tl_green,	0	1	1 /
ML_out_1,	1	1	2
ML_out_2,	1	1 dd	2
IL_in,	1	1 BBB	2
IL_in,	1	1	2
IL_tl_green,	1	0	1 /
IL_out_1,	1	1	2
IL_out_2,	1	1 dd	2
ML_in,	1	1 BBBS	2
ML_in	1	1	2
>			

variants : *ml_pass* + *il_pass*



PO of Convergence/Non-Divergence/Livelock Freedom

A New Event Occurrence Decreases Variant

$A(c)$

$I(c, v)$

$J(c, v, w)$

$H(c, w)$

$\vdash$

$\boxed{V(c, F(c, w))} < \boxed{V(c, w)}$

Post-state evaluation

Pre-state evaluation

VAR

applicable to new events

* $\cancel{ml\_pass'}^{\,0} + \cancel{il\_pass'}^{\,il\_pass}$

$< ml\_pass + il\_pass$

Variants: ml_pass + il_pass

ML_tl_green/VAR

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

$COLOUR = \{green, red\}$ $green \neq red$

$n \in \mathbb{N}$ $n \leq d$] m0

$a \in \mathbb{N}$ $b \in \mathbb{N}$ $c \in \mathbb{N}$] m1

$a + b + c = n$ $a = 0 \lor c = 0$

$ml\_tl \in COLOUR$ $il\_tl \in COLOUR$] m2

$ml\_tl = green \Rightarrow a + b < d \land c = 0$ $il\_tl = green \Rightarrow b > 0 \land a = 0$

$ml\_tl = red \lor il\_tl = red$

$ml\_pass \in \{0, 1\}$ $il\_pass \in \{0, 1\}$

$ml\_tl = red \Rightarrow ml\_pass = 1$ $il\_tl = red \Rightarrow il\_pass = 1$

$ml\_tl = red$ $a + b < d$ $c = 0$]

$il\_pass = 1$

Concrete guards of ML_tl_green

$\vdash$

* $\boxed{0 + il\_pass < ml\_pass + il\_pass}$

```
ML_tl_green
  when
    ml_tl = red
    a + b < d
    c = 0
    il_pass = 1
  then
    ml_tl := green
    il_tl := red
    ml_pass := 0
  end
```

BAP:

$ml\_pass' = 0$

$il\_pass' = il\_pass$

Lecture 2

Part S

Case Study on Reactive Systems - Bridge Controller 2nd Refinement: Relative Deadlock Freedom

PO of Relative Deadlock Freedom

axm0_1	$d \in \mathbb{N}$
axm0_2	$d > 0$
axm2_1	$COLOUR = \{green, red\}$
axm2_2	$green \neq red$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
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	$a = 0 \lor c = 0$
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	$il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5	$ml\_tl = red \lor il\_tl = red$
inv2_6	$ml\_pass \in \{0, 1\}$
inv2_7	$il\_pass \in \{0, 1\}$
inv2_8	$ml\_tl = red \Rightarrow ml\_pass = 1$
inv2_9	$il\_tl = red \Rightarrow il\_pass = 1$

Disjunction of *abstract* guards

	$a + b < d \land c = 0$	**guards of** *ML_out* **in** m_1
$\lor$	$c > 0$	**guards of** *ML_in* **in** m_1
$\lor$	$a > 0$	**guards of** *IL_in* **in** m_1
$\lor$	$b > 0 \land a = 0$	**guards of** *IL_out* **in** m_1

$\vdash$

Disjunction of *concrete* guards

	$ml\_tl = red \land a + b < d \land c = 0 \land il\_pass = 1$	**guards of** *ML_tl_green* **in** m_2
$\lor$	$il\_tl = red \land b > 0 \land a = 0 \land ml\_pass = 1$	**guards of** *IL_tl_green* **in** m_2
$\lor$	$ml\_tl = green \land a + b + 1 \neq d$	**guards of** *ML_out_1* **in** m_2
$\lor$	$ml\_tl = green \land a + b + 1 = d$	**guards of** *ML_out_2* **in** m_2
$\lor$	$il\_tl = green \land b \neq 1$	**guards of** *IL_out_1* **in** m_2
$\lor$	$il\_tl = green \land b = 1$	**guards of** *IL_out_2* **in** m_2
$\lor$	$a > 0$	**guards of** *ML_in* **in** m_2
$\lor$	$c > 0$	**guards of** *IL_in* **in** m_2

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

```
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
```

Concrete m2

```
ML_tl_green
  when
    $ml\_tl = red$
    $a + b < d$
    $c = 0$
    $il\_pass = 1$
  then
    $ml\_tl := green$
    $il\_tl := red$
    $ml\_pass := 0$
  end
```

```
IL_tl_green
  when
    $il\_tl = red$
    $b > 0$
    $a = 0$
    $ml\_pass = 1$
  then
    $il\_tl := green$
    $ml\_tl := red$
    $il\_pass := 0$
  end
```

```
ML_out_1
  when
    $ml\_tl = green$
    $a + b + 1 \neq d$
  then
    $a := a + 1$
    $ml\_pass := 1$
  end
```

```
IL_out_1
  when
    $il\_tl = green$
    $b \neq 1$
  then
    $b := b - 1$
    $c := c + 1$
    $il\_pass := 1$
  end
```

```
ML_out_2
  when
    $ml\_tl = green$
    $a + b + 1 = d$
  then
    $a := a + 1$
    $ml\_tl := red$
    $ml\_pass := 1$
  end
```

```
IL_out_2
  when
    $il\_tl = green$
    $b = 1$
  then
    $b := b - 1$
    $c := c + 1$
    $il\_tl := red$
    $il\_pass := 1$
  end
```

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

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

Discharging POs of m2: Relative Deadlock Freedom

$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow b > 0 \land a = 0$
$ml\_tl = red \lor il\_tl = red$
$ml\_pass \in \{0, 1\}$
$il\_pass \in \{0, 1\}$
$ml\_tl = red \Rightarrow ml\_pass = 1$
$il\_tl = red \Rightarrow il\_pass = 1$
$\quad a + b < d \land c = 0$
$\lor \; c > 0$
$\lor \; a > 0$
$\lor \; b > 0 \land a = 0$
$\vdash$
$\quad ml\_tl = red \land a + b < d \land c = 0 \land il\_pass = 1$
$\lor \; il\_tl = red \land b > 0 \land a = 0 \land ml\_pass = 1$
$\lor \; ml\_tl = green$
$\lor \; il\_tl = green$
$\lor \; a > 0$
$\lor \; c > 0$

Ex.1

⋮

$d \in \mathbb{N}$
$d > 0$
$b \in \mathbb{N}$
$ml\_tl = red$
$il\_tl = red$
$ml\_tl = red \Rightarrow ml\_pass = 1$
$il\_tl = red \Rightarrow il\_pass = 1$
$\vdash$
$\quad b < d \land ml\_pass = 1 \land il\_pass = 1$
$\lor \; b > 0 \land ml\_pass = 1 \land il\_pass = 1$

Ex.2

…

$d \in \mathbb{N}$
$d > 0$
$b \in \mathbb{N}$
$ml\_tl = red$
$il\_tl = red$
$ml\_pass = 1$
$il\_pass = 1$
$\vdash$
$\quad b < d \land ml\_pass = 1 \land il\_pass = 1$
$\lor \; b > 0 \land ml\_pass = 1 \land il\_pass = 1$

Ex3

…

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

ARI

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

OR_L

$d > 0$
$b > 0$
$\vdash$
$b < d \lor b > 0$

OR_R2

$d > 0$
$b > 0$
$\vdash$
$b > 0$

HYP

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

EQ_LR, MON

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

OR_R1

$d > 0$
$\vdash$
$0 < d$

HYP

1st Refinement and 2nd Refinement: Provably Correct

variables: a, b, c

constants: d

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

variants:
$2 \cdot a + b$

```
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
```

Abstract m1

Art

Correctness Criteria:

+ Guard Strengthening
+ Invariant Establishment
+ Invariant Preservation
+ Convergence
+ Relative Deadlock Freedom

superposition

variables:
a
b
c
ml_tl
il_tl
ml_pass
il_pass

constants: d

sets: $COLOR$

axioms:
- **axm0_1** : $d \in \mathbb{N}$
- **axm0_2** : $d > 0$
- **axm2_1** : $COLOR = \{green, red\}$
- **axm2_2** : $green \neq red$

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** : $il\_tl = green \Rightarrow b > 0 \land a = 0$
- **inv2_5** : $ml\_tl = red \lor il\_tl = red$
- **inv2_6** : $ml\_pass \in \{0, 1\}$
- **inv2_7** : $il\_pass \in \{0, 1\}$
- **inv2_8** : $ml\_tl = red \Rightarrow ml\_pass = 1$
- **inv2_9** : $il\_tl = red \Rightarrow il\_pass = 1$

variants:
$ml\_pass + il\_pass$

divergence freedom

```
ML_tl_green
  when
    $ml\_tl = red$
    $a + b < d$
    $c = 0$
    $il\_pass = 1$
  then
    $ml\_tl := green$
    $il\_tl := red$
    $ml\_pass := 0$
  end
```

```
IL_tl_green
  when
    $il\_tl = red$
    $b > 0$
    $a = 0$
    $ml\_pass = 1$
  then
    $il\_tl := green$
    $ml\_tl := red$
    $il\_pass := 0$
  end
```

```
ML_out_1
  when
    $ml\_tl = green$
    $a + b + 1 \neq d$
  then
    $a := a + 1$
    $ml\_pass := 1$
  end
```

```
ML_out_2
  when
    $ml\_tl = green$
    $a + b + 1 = d$
  then
    $a := a + 1$
    $ml\_tl := red$
    $ml\_pass := 1$
  end
```

```
IL_out_1
  when
    $il\_tl = green$
    $b \neq 1$
  then
    $b := b - 1$
    $c := c + 1$
    $il\_pass := 1$
  end
```

```
IL_out_2
  when
    $il\_tl = green$
    $b = 1$
  then
    $b := b - 1$
    $c := c + 1$
    $il\_tl := red$
    $il\_pass := 1$
  end
```

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

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

Concrete m2

Lecture 3

Part A

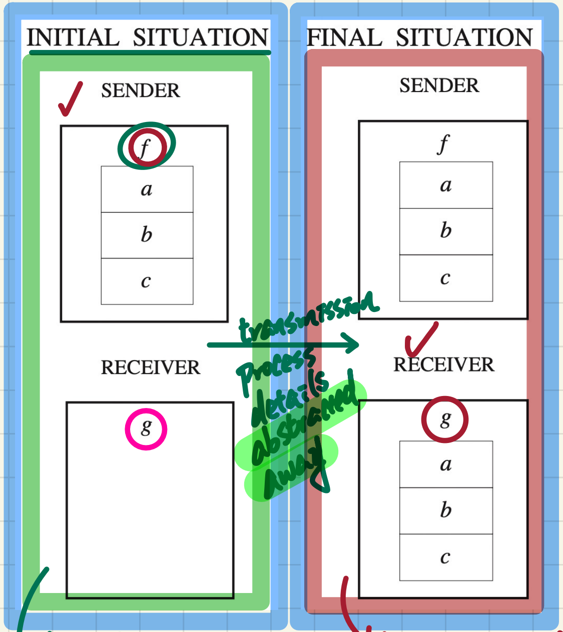
Case Study on Distributed Programs - File Transfer Protocol Initial Model: State and Events

FTP: Abstraction and State Space in the Initial Model

REQ1	The protocol ensures the copy of a file from the sender to the receiver.

e.g. $n=3$ $f \in 1..\cancel{n}_{3} \rightarrow \underline{\underline{D}}$ $d1, d2, d3, \ldots$ $f = \{(1, \underline{d2}), (2, \underline{d1}), (3, \underline{d3})\}$

Synchronous Transmission



$b = \text{FALSE} \Rightarrow g = \emptyset$ $b = \text{TRUE} \Rightarrow g = f$

Static Part of Model

carrier sets: membership abstracted away

sets: D $BOOLEAN$

data item

constants: n f

→ file on sender

↓ max size of file

axioms:

axm0_1 : $n > 0$

axm0_2 : $f \in 1 \,..\, n \rightarrow D$ total function

axm0_3 : $BOOLEAN = \{TRUE, FALSE\}$

Dynamic Part of Model

e.g. $n=3$, $g \in 1..\cancel{n}_{3} \nrightarrow \underline{\underline{D}}$ $d1, d2, d3$ $g = \{(1, \underline{d2}), (3, \underline{d3})\}$

variables: g, b ✓

whether or not the transmission has been completed

invariants:

inv0_1a : $g \in g \in 1 \,..\, n \nrightarrow D$ partial function

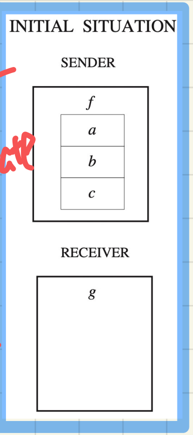
inv0_1b : $b \in BOOLEAN$

inv0_2 : * ??

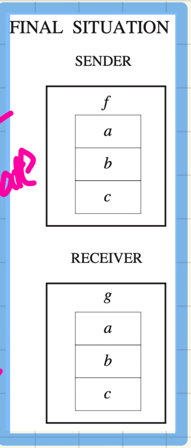
inv0_3 : ** ??

Conditional invariants

FTP: Events of Initial Model



post-state of init event



post-state of final event

sets: $D, BOOLEAN$

constants: n, f

axioms:
- **axm0_1** : $n > 0$
- **axm0_2** : $f \in 1 \,..\, n \rightarrow D$
- **axm0_3** : $BOOLEAN = \{TRUE, FALSE\}$

variables: g, b

invariants:
- **inv0_1a** : $g \in g \in 1 \,..\, n \nrightarrow D$
- **inv0_1b** : $b \in BOOLEAN$
- **inv0_2** : $b = FALSE \Rightarrow g = \varnothing$
- **inv0_3** : $b = TRUE \Rightarrow g = f$

init:

sender's file ready for transmission

```
init
  begin
    ??
  end
```

$g := \varnothing$

$b := FALSE$

enables

final:

sender's file transmitted to receiver

```
final
  when
    ??
  then
    ??
  end
```

$b = FALSE$

$g := f$

$b := TRUE$

before transmission can be completed, it must have not been started

PO of Invariant Establishment

sets: $D, BOOLEAN$

constants: n, f

axioms:
- **axm0_1** : $n > 0$
- **axm0_2** : $f \in 1 \,..\, n \rightarrow D$
- **axm0_3** : $BOOLEAN = \{TRUE, FALSE\}$

variables: g, b

invariants:
- ✓ **inv0_1a** : $g \in 1 \,..\, n \nrightarrow D$ ✓
- **inv0_1b** : $b \in BOOLEAN$
- **inv0_2** : $b = FALSE \Rightarrow g = \varnothing$
- **inv0_3** : $b = TRUE \Rightarrow g = f$

init
  begin
    $g := \varnothing$
    $b := FALSE$
  end

BAP: $g' = \varnothing \wedge b' = FALSE$

Rule of Invariant Establishment

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

$A(c)$

$\vdash$

$I_i(c, \mathbf{K(c)})$

INV

Components

K(c): effect of init's actions

v′ = K(c): BAP of init's actions

Exercise: Generate Sequents from the INV rule.

init/inv0_1a/INV

$n > 0$

$f \in 1..n \rightarrow D$

$BOOLEAN = \{TRUE, FALSE\}$

$\vdash$

$\underset{\emptyset}{g'} \in 1..n \nrightarrow D$

init/inv0_2/INV

$n > 0$

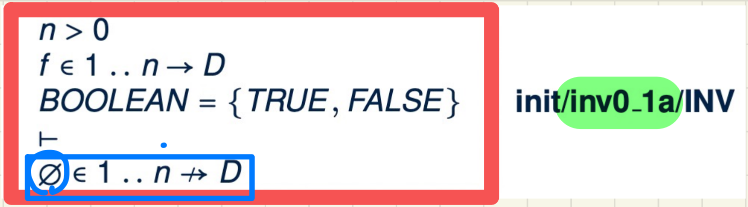
$f \in 1..n \rightarrow D$

$BOOLEAN = \{TRUE, FALSE\}$

$\vdash$

$\underset{FALSE}{b'} = FALSE \Rightarrow \underset{\emptyset}{g'} = \emptyset$

Discharging PO of Invariant Establishment



$n > 0$
$f \in 1 \mathbin{..} n \rightarrow D$
$BOOLEAN = \{TRUE, FALSE\}$
$\vdash$
$\varnothing \in 1 \mathbin{..} n \nrightarrow D$

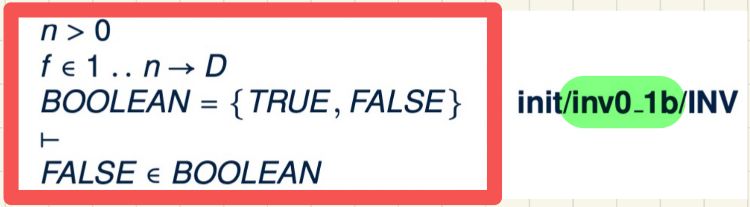
init/inv0_1a/INV

ARI

$$\boxed{\begin{array}{l} n>0 \\ f \in 1 \mathbin{..} n \rightarrow D \\ BOOLEAN = \{TRUE, FALSE\} \\ \vdash \; T \; \cancel{TRUE} \end{array}}$$

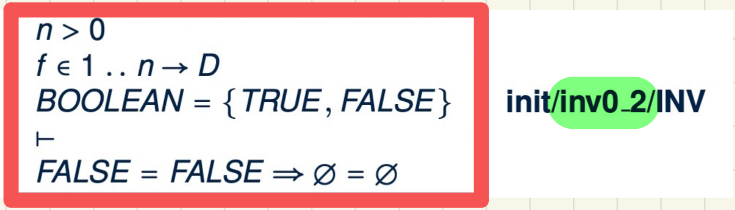
TRUE_R

$\varnothing$ is always a partial function whose domain & range are $\varnothing$



$n > 0$
$f \in 1 \mathbin{..} n \rightarrow D$
$BOOLEAN = \{TRUE, FALSE\}$
$\vdash$
$FALSE \in BOOLEAN$

init/inv0_1b/INV



$n > 0$
$f \in 1 \mathbin{..} n \rightarrow D$
$BOOLEAN = \{TRUE, FALSE\}$
$\vdash$
$FALSE = FALSE \Rightarrow \varnothing = \varnothing$

init/inv0_2/INV

MON

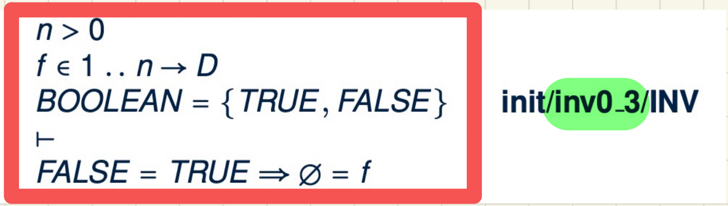
$$\boxed{\begin{array}{l} \vdash \\ FALSE = FALSE \Rightarrow \varnothing = \varnothing \end{array}}$$

ARI

$$\boxed{\begin{array}{l} \vdash \\ \quad T \end{array}}$$

TRUE_R

① $FALSE = FALSE \equiv T$
② $\varnothing = \varnothing \equiv T$
③ $T \Rightarrow T \equiv T$



$n > 0$
$f \in 1 \mathbin{..} n \rightarrow D$
$BOOLEAN = \{TRUE, FALSE\}$
$\vdash$
$FALSE = TRUE \Rightarrow \varnothing = f$

init/inv0_3/INV

PO of Invariant Preservation

sets: $D, BOOLEAN$

constants: n, f

axioms:
- axm0_1 : $n > 0$
- axm0_2 : $f \in 1 \,..\, n \rightarrow D$
- axm0_3 : $BOOLEAN = \{TRUE, FALSE\}$

variables: g, b

invariants:
- ✓ inv0_1a : $g \in 1 \,..\, n \nvrightarrow D$
- ✓ ✓ inv0_1b : $b \in BOOLEAN$
- ✓ inv0_2 : $b = FALSE \Rightarrow g = \varnothing$
- ✓ inv0_3 : $b = TRUE \Rightarrow g = f$

```
final
  when
    b = FALSE
  then
    g := f
    b := TRUE
  end
```

BAP:

Rule of Invariant Preservation

$$\begin{array}{l} A(c) \\ I(c, v) \\ G(c, v) \\ \vdash \\ I_i(c, E(c, v)) \end{array}$$

Exercise: $g' = f \wedge b' = FALSE$

Generate Sequents from the INV rule.

final/inv0_1a/INV ✓

$$\begin{array}{l} n > 0 \\ f \in 1..n \rightarrow D \\ BOOLEAN = \{TRUE, FALSE\} \\ g \in 1..n \nvrightarrow D \\ b \in BOOLEAN \\ b = FALSE \Rightarrow g = \varnothing \\ b = TRUE \Rightarrow g = f \\ b = FALSE \\ \vdash \; * \end{array}$$

$* \;\; \underset{f}{\cancel{g'}} \in 1..n \nvrightarrow D$

final/inv0_2/INV

$$\begin{array}{l} n > 0 \\ f \in 1..n \rightarrow D \\ BOOLEAN = \{TRUE, FALSE\} \\ g \in 1..n \nvrightarrow D \\ b \in BOOLEAN \\ b = FALSE \Rightarrow g = \varnothing \\ b = TRUE \Rightarrow g = f \\ b \quad TALSE \\ \vdash \; ** \end{array}$$

$\underset{FALSE}{\cancel{b'}} = TRUE \Rightarrow \underset{f}{\cancel{g'}} = f$

Discharging POs of m0: Invariant Preservation

final/inv0_1a/INV

$n > 0$
$f \in 1\,..\,n \to D$ ✓
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1\,..\,n \mathrel{+\!\!\!\!\to} D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$b = FALSE$
$\vdash$
$f \in 1\,..\,n \mathrel{+\!\!\!\!\to} D$

MON

$$\boxed{\begin{array}{l} f \in 1..n \to D \\ \vdash \\ f \in 1..n \mathrel{+\!\!\!\!\to} D \end{array}}$$

ARI

① a total fun. is a special case of partial fun.

② But a partial fun is not necessarily a total fun.

final/inv0_1b/INV

$n > 0$
$f \in 1\,..\,n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1\,..\,n \mathrel{+\!\!\!\!\to} D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$b = FALSE$
$\vdash$
$TRUE \in BOOLEAN$

final/inv0_2/INV

$n > 0$
$f \in 1\,..\,n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1\,..\,n \mathrel{+\!\!\!\!\to} D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$b = FALSE$
$\vdash$
$TRUE = FALSE \Rightarrow f = \varnothing$

MON

$$\boxed{\begin{array}{l} \vdash \\ TRUE = FALSE \Rightarrow f = \varnothing \end{array}}$$

① $TRUE = FALSE \equiv \bot$

② $\bot \Rightarrow P \equiv \top$

ARI

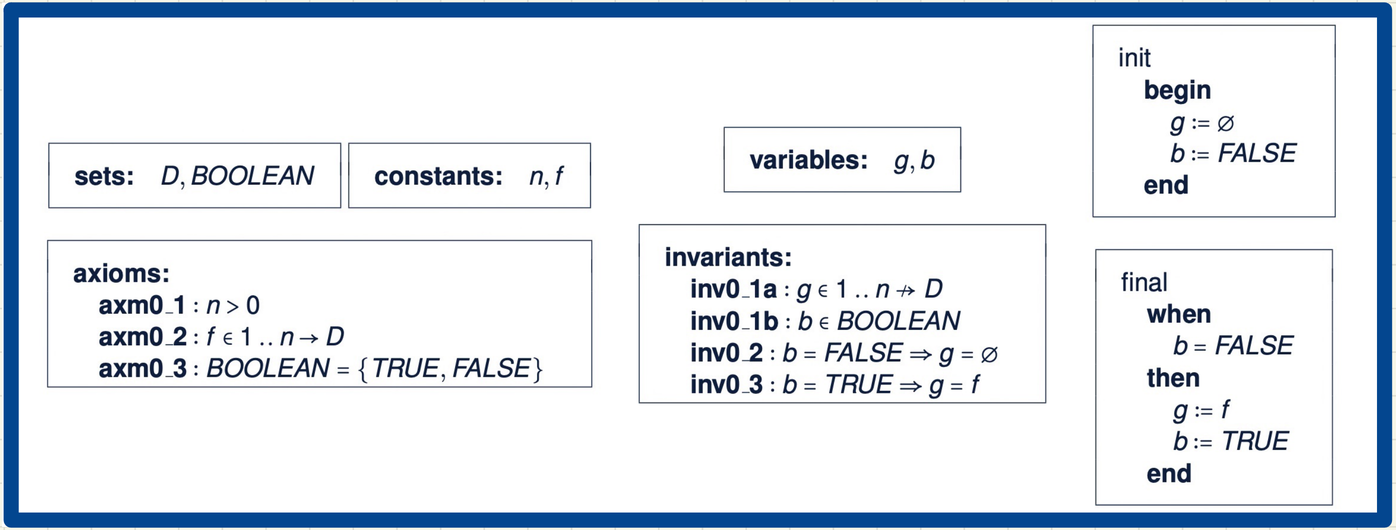
$$\boxed{\vdash \top}$$

TRUE_R

final/inv0_3/INV

$n > 0$
$f \in 1\,..\,n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$g \in 1\,..\,n \mathrel{+\!\!\!\!\to} D$
$b \in BOOLEAN$
$b = FALSE \Rightarrow g = \varnothing$
$b = TRUE \Rightarrow g = f$
$b = FALSE$
$\vdash$
$TRUE = TRUE \Rightarrow f = f$

Summary of the Initial Model: Provably Correct



sets: $D, BOOLEAN$

constants: n, f

axioms:
- **axm0_1** : $n > 0$
- **axm0_2** : $f \in 1 \,..\, n \to D$
- **axm0_3** : $BOOLEAN = \{TRUE, FALSE\}$

variables: g, b

invariants:
- **inv0_1a** : $g \in 1 \,..\, n \pinj D$
- **inv0_1b** : $b \in BOOLEAN$
- **inv0_2** : $b = FALSE \Rightarrow g = \varnothing$
- **inv0_3** : $b = TRUE \Rightarrow g = f$

```
init
  begin
    $g := \varnothing$
    $b := FALSE$
  end
```

```
final
  when
    $b = FALSE$
  then
    $g := f$
    $b := TRUE$
  end
```

Correctness Criteria:

+ Invariant Establishment
+ Invariant Preservation
+ Deadlock Freedom

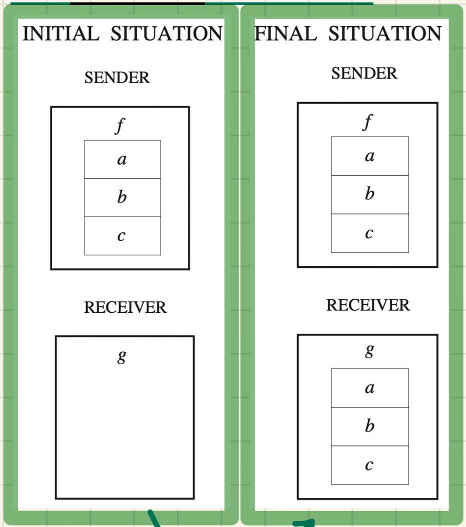
Lecture 3

Part B

*Case Study on Distributed Programs -
File Transfer Protocol
1st Refinement: State, Events, Proofs*

FTP: Abstraction in the 1st Refinement

m0: most abstract

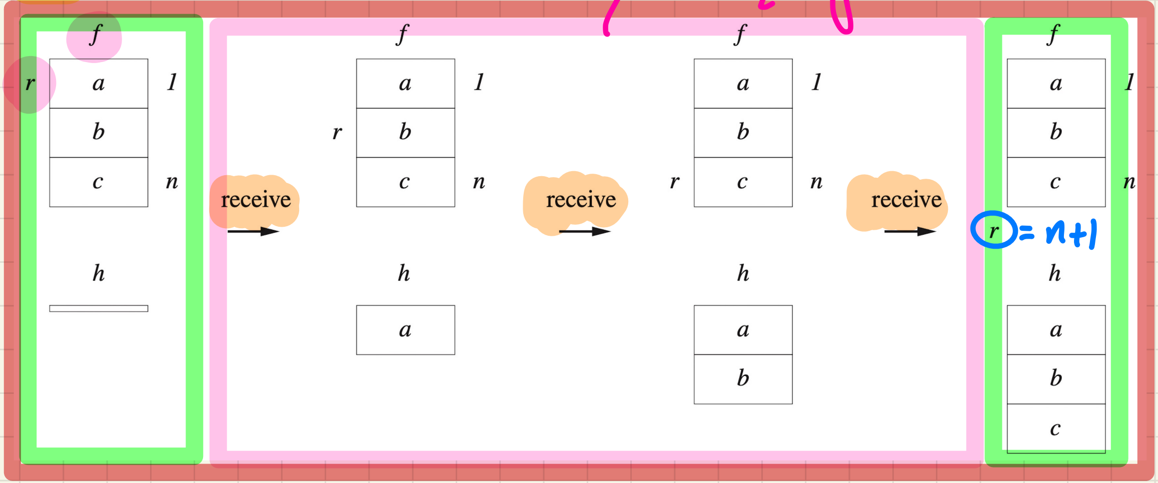


synchronous & instantaneous

REQ2	The file is supposed to be made of a sequence of items.
REQ3	The file is sent piece by piece between the two sites.

refinement:
1. asynchronous
2. gradual

m1: more concrete than m0



FTP: State Space of the 1st Refinement

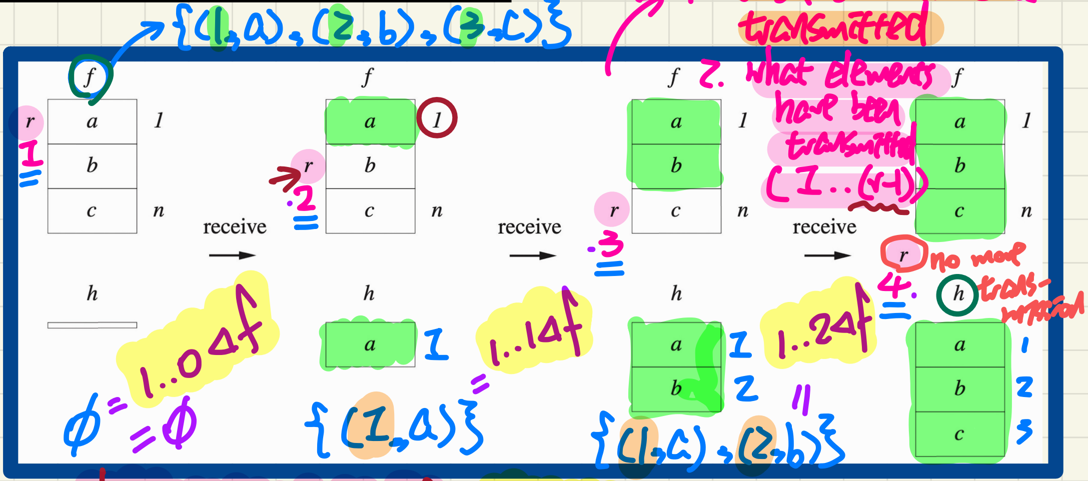
r value indicates:
1. which element to be transmitted
2. what elements have been transmitted (1..(r−1))

Static Part of Model

sets: $D, BOOLEAN$

constants: n, f

axioms:
- axm0_1 : $n > 0$
- axm0_2 : $f \in 1..n \rightarrow D$
- axm0_3 : $BOOLEAN = \{TRUE, FALSE\}$



Dynamic Part of Model

variables: b, h, r

invariants:
- inv1_1 : $r \in 1..n+1$
- inv1_2 : ?? *
- inv1_3 : ?? **
- thm1_1 : ?? ***

to be proved for establishment & preservation

1. need not be proved for establishment & preservation
2. to be proved as derivable from invariants

$* \quad h = \underbrace{(1..(r-1))}_{\{1,2,\dots,r-1\}} \lhd f$

$1..0 = \varnothing$

$** \quad b = TRUE \Rightarrow r = n+1$

$*** \quad b = TRUE \Rightarrow h = f$

$\{(1,a),(2,b),(3,c)\} = \underbrace{1..4}_{dom(f)} \lhd f$

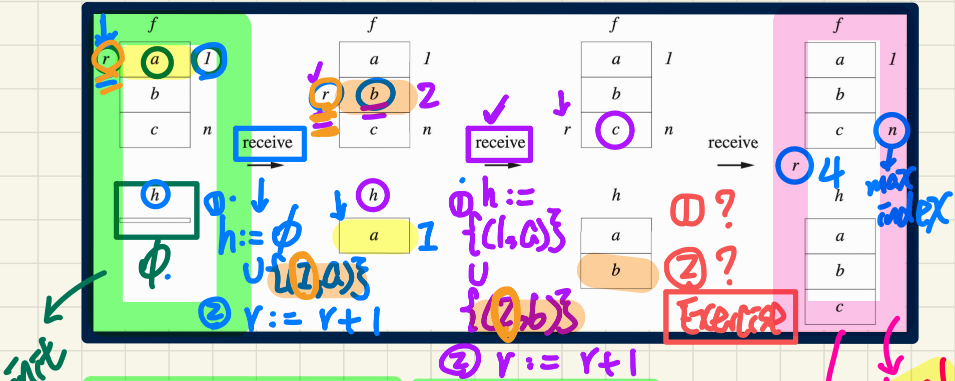
Exercises

inv1_2: elements up to index r − 1 ✓ have been transmitted

inv1_3: transmission completed means no more elements to be transmitted

thm1_1: transmission completed means ✓ receiver has a copy of sender's file

FTP: Concrete Events in 2nd Refinement



sets: $D, BOOLEAN$

constants: n, f

axioms:

- axm0_1 : $n > 0$
- axm0_2 : $f \in 1 \,..\, n \rightarrow D$
- axm0_3 : $BOOLEAN = \{TRUE, FALSE\}$

variables: b, h, r

invariants:

- inv1_1 : $r \in 1 \,..\, n+1$
- inv1_2 : $h = (1 \,..\, r-1) \lhd f$
- inv1_3 : $b = TRUE \Rightarrow r = n+1$
- thm1_1 : $b = TRUE \Rightarrow h = f$

As soon as "receive" becomes disabled, "final" should be ready to occur.

init: getting the transmission ready

```
init
  begin
    ??
  end
```

$b := FALSE$

$h := \emptyset$

$r := 1$

receive: transmitting element by element

```
receive
  when
    ??
  then
    ??
  end
```

$r \leq n$

$h := h \cup \{(r, f(r))\}$

sender's private info should be hidden

occurrence of final is restricted to 1

final: finalizing the transmission

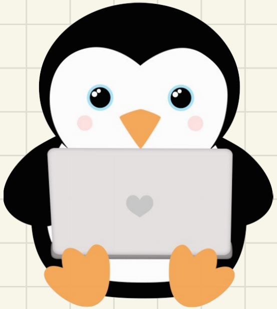
```
final
  when
    ??
  then
    ??
  end
```

$\boxed{b = FALSE}$

$r = n+1$

$b := TRUE$

I hope you enjoyed learning with me ☺



All the best to you !