EECS3342 System Specification and Refinement

Lecture Notes

Fall 2024

Jackie Wang

Lecture 1 - Sep. 5

Syllabus & Introduction

Formal Methods:
Theorem Proving vs. Model Checking

Course Learning Outcomes (CLOs)

CLO1 Document requirements organizing them into appropriate categories such as environmental constraints versus functional properties (safety and progress).

CLO2 Construct high level, abstract mathematical models of a system (consisting of both the system and its environment) amenable to formal reasoning.

CLO3 Apply set theory and predicate logic to express functional and safety properties from the requirements as events, guards, system variants and invariants of a state-event model.

CLO4 Use models to reason about and predict their safety and progress properties.

CLO5 Plan and construct a sequence of refinements from abstract high-level specifications to implemented code.

CLO6 Prove that a concrete system refines an abstract model.

CLO7 Apply the method to a variety of systems such as sequential, concurrent and embedded systems.

CLO8 Use practical tools for constructing and reasoning about the models.

CLO9 Use Hoare Logic and Dijkstra weakest precondition calculus to derive correct designs.

mathematical

Formal Methods

discrete math

(∵ computer system is discrete (0,1))

$x \in \{1..10\}$

$y \in \{1..5\}$

- 3342
  - theorem proving
    - manual (interactions with prover)
    - deductions, constants, variables, axioms, invariants ⇒ properties?
- 4315 Mission-Critical system
  - model checking
    - model + properties? → model checker
      - yes → proved
      - no → not proved (counter example)
    - state explosion
    - (tool does not halt)
    - system is too large for the tool to conclude

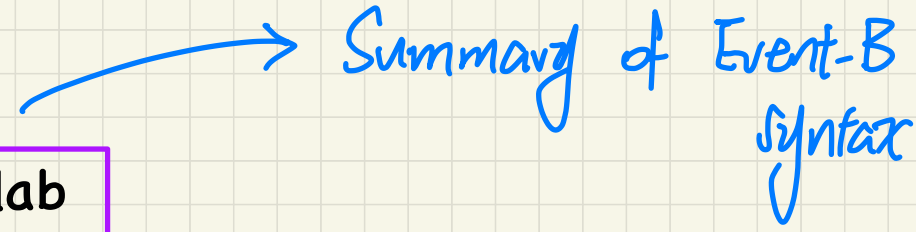
Lecture 2 - Sep. 10

Introduction

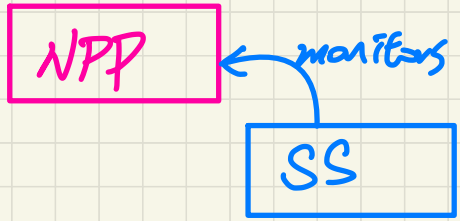
Safety-Critical vs. Missional-Critical
Professional Engineers: Code of Ethics
Safety Property/Invariant
Verification vs. Validation

Announcements/Reminders

- Priority: Lab1
- Wednesday's lab



Safety-Critical System



1. nuclear power plant
   + nuclear shutdown system

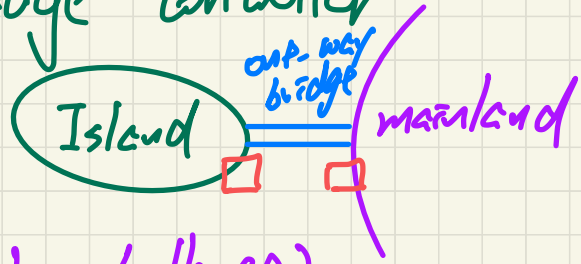
2. radiation

3. "glove"

4. pacemaker (pacemaker challenge)

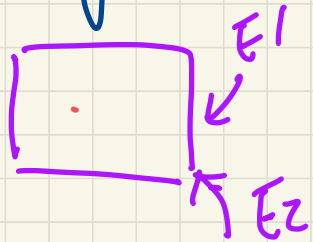
5. auto-pilot & auto-driving.

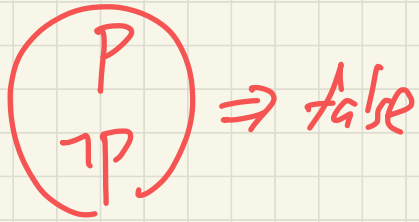
6. bridge controller



Acceptance Criteria

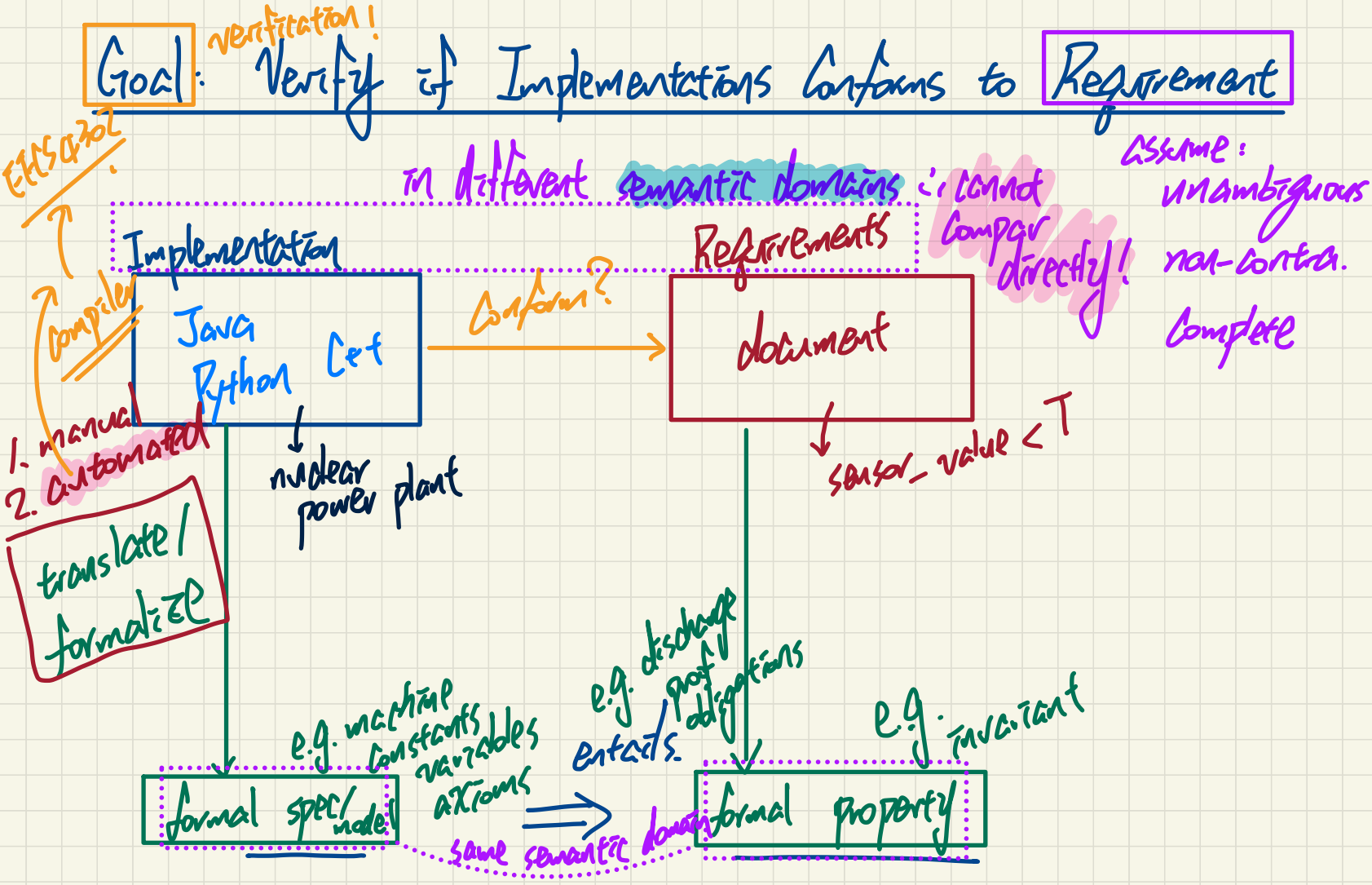
1. Req. precise
   - ↳ no ambiguities, no contradiction





   - ↳ complete
     - ↳ no missing scenarios

Labs


Goal: Verify if Implementation conforms to Requirement
verification!
Eclipse
compiler
in different semantic domains
cannot compare directly!
assume:
unambiguous
non-contra.
complete
Implementation
Java
Python
Cet
nuclear power plant
conform?
Requirements
document
sensor_value < T
1. manual
2. automated
translate / formalize
e.g. machine constants variables axioms
e.g. discharge proof obligations
entails
e.g. invariant
formal spec/model
formal property
same semantic domain

Mission-Critical vs. Safety-Critical

Safety critical

When defining safety critical it is beneficial to look at the definition of each word independently. Safety typically refers to being free from danger, injury, or loss. In the commercial and military industries this applies most directly to human life. Critical refers to a task that must be successfully completed to ensure that a larger, more complex operation succeeds. Failure to complete this task compromises the integrity of the entire operation. Therefore a safety-critical application for an RTOS implies that execution failure or faulty execution by the operating system could result in injury or loss of human life.

Safety-critical systems demand software that has been developed using a well-defined, mature software development process focused on producing quality software. For this very reason the DO-178B specification was created. DO-178B defines the guidelines for development of aviation software in the USA. Developed by the Radio Technical Commission for Aeronautics (RTCA), the DO-178B standard is a set of guidelines for the production of software for airborne systems. There are multiple criticality levels for this software (A, B, C, D, and E).

These levels correspond to the consequences of a software failure:

- Level A is catastrophic
- Level B is hazardous/severe
- Level C is major
- Level D is minor
- Level E is no effect

SCS

MCS

Safety-critical software is typically DO-178B level A or B. At these higher levels of software criticality the software objectives defined by DO-178B must be reviewed by an independent party and undergo more rigorous testing. Typical safety-critical applications include both military and commercial flight, and engine controls.

Mission critical

A mission refers to an operation or task that is assigned by a higher authority. Therefore a mission-critical application for an RTOS implies that a failure by the operating system will prevent a task or operation from being performed, possibly preventing successful completion of the operation as a whole.

Mission-critical systems must also be developed using well-defined, mature software development processes. Therefore they also are subjected to the rigors of DO-178B. However, unlike safety-critical applications, mission-critical software is typically DO-178B level C or D. Mission-critical systems only need to meet the lower criticality levels set forth by the DO-178B specification.

Generally mission-critical applications include navigation systems, avionics display systems, and mission command and control.

Source: http://pdf.cloud.opensystemsmedia.com/advancedtca-systems.com/SBS.Jan04.pdf

SCS vs. MCS

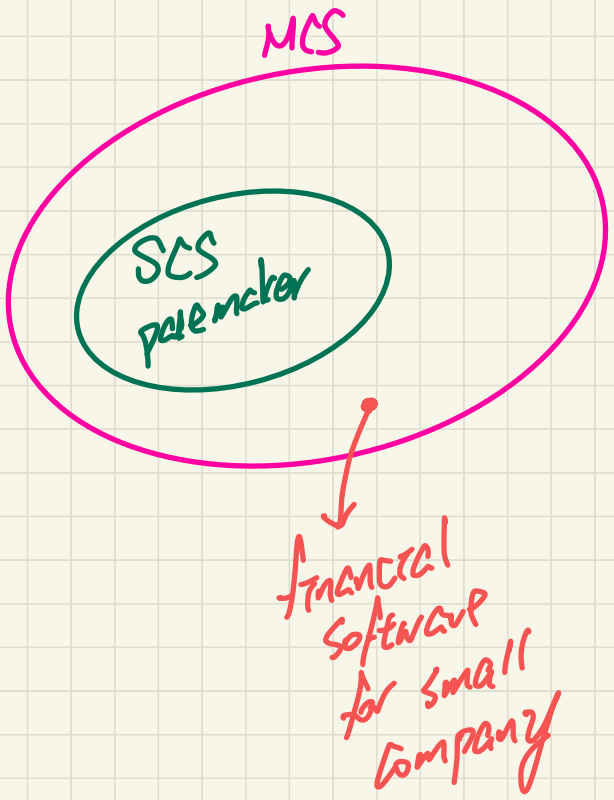
(1) ✗ SCS $\Leftrightarrow$ MCS

(2) ✓ SCS $\Rightarrow$ MCS

(3) ✗ MCS $\Rightarrow$ SCS

SCS $\subseteq$ MCS

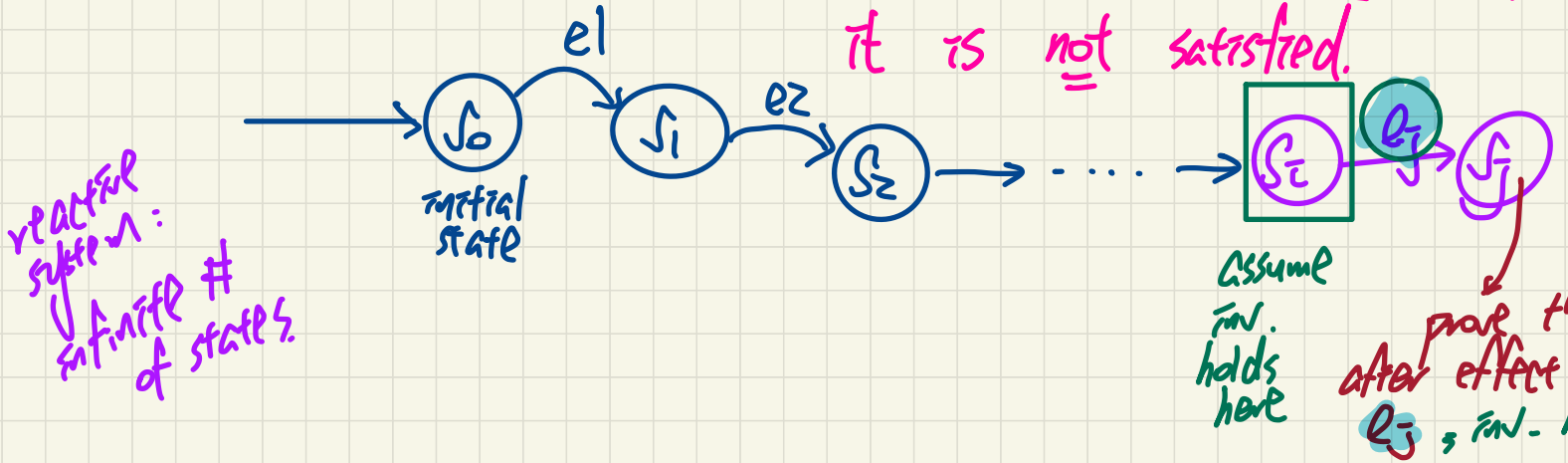
$\Rightarrow$



Safety Property / Invariant

↳ Every possible state of the system should satisfy it.

↳ If there's at least one state where the inv. does not hold, it is not satisfied.



assumption: req. already given

Verification: Are we building the product right?

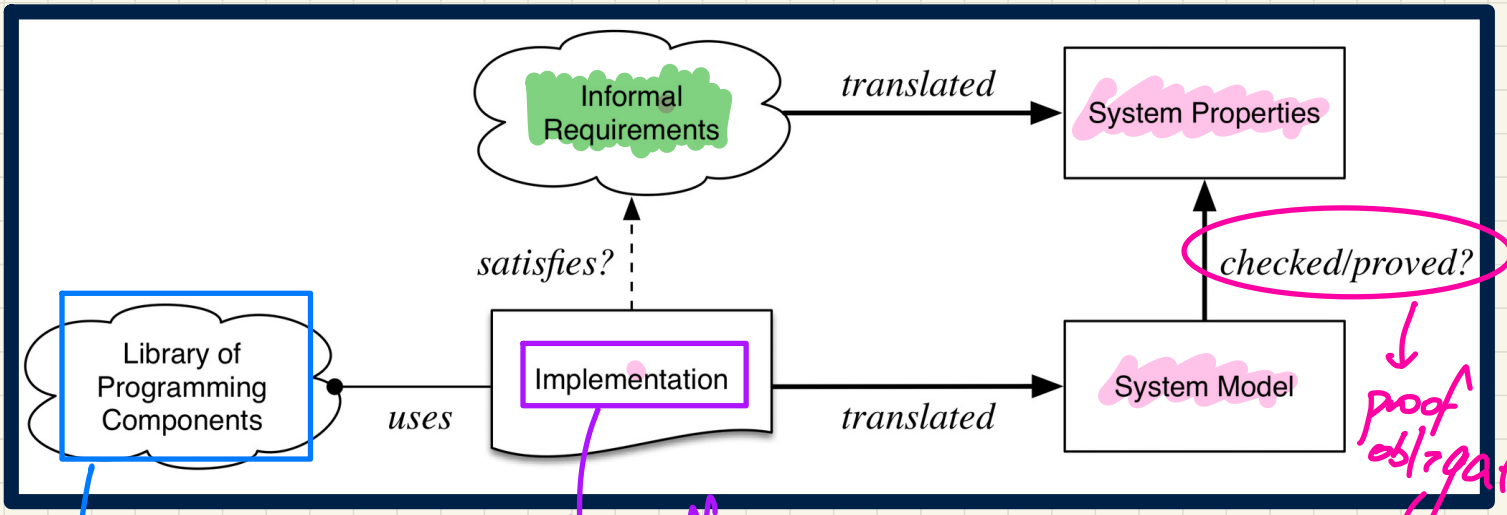
process of construction

4312.

Validation: Are we building the right product?

Are the req. given truly intended by customers?

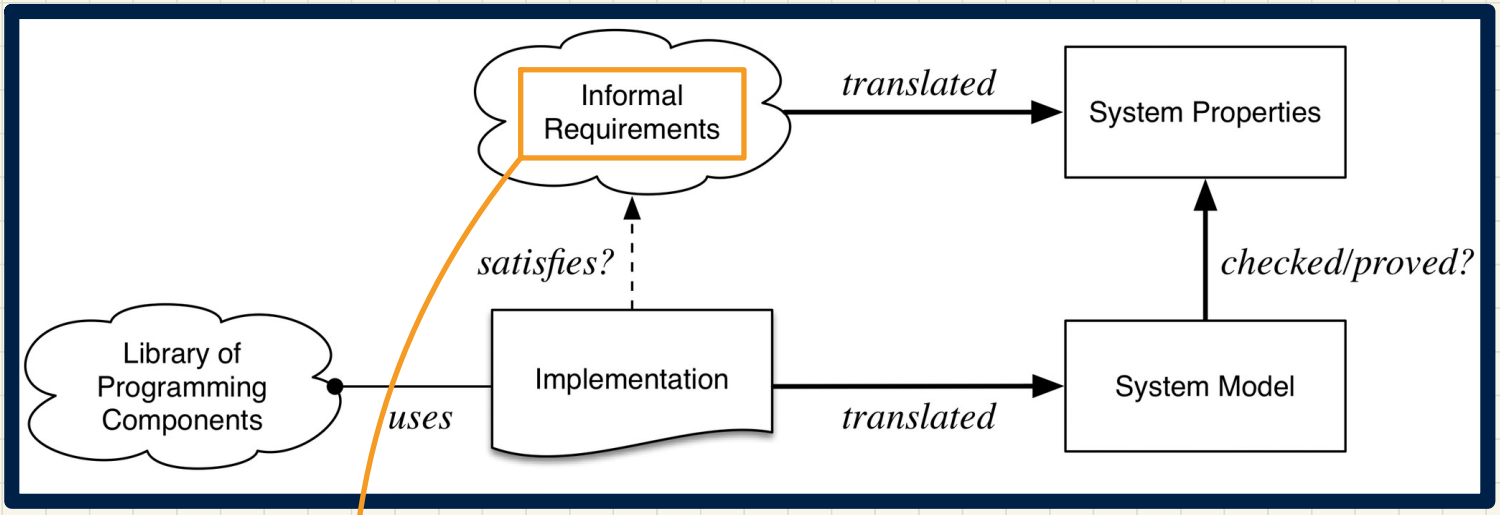
Building the product right?



Are the library classes reliable?
(additional verification needed e.g. testing)

1. In a diff. semantic domain
2. make use of library modules.

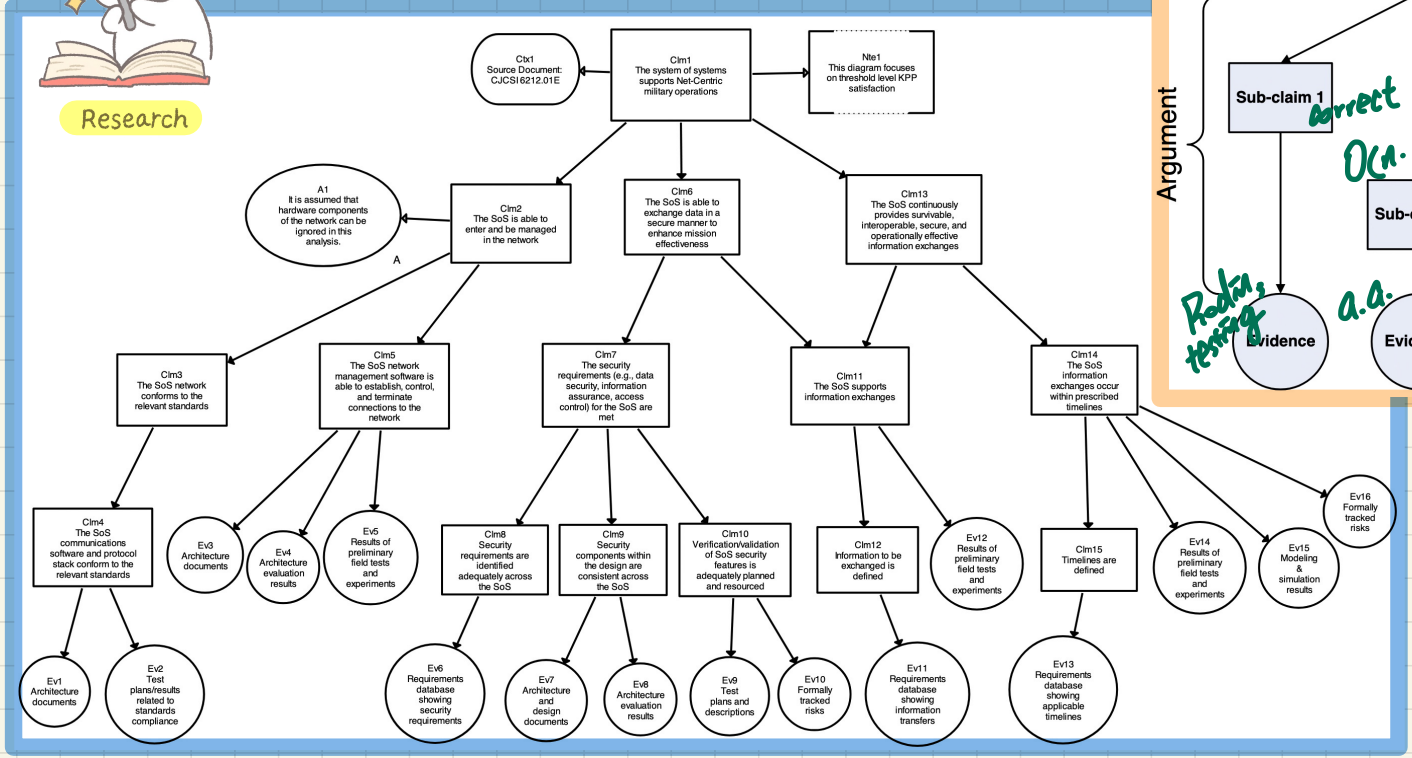
Building the right product?

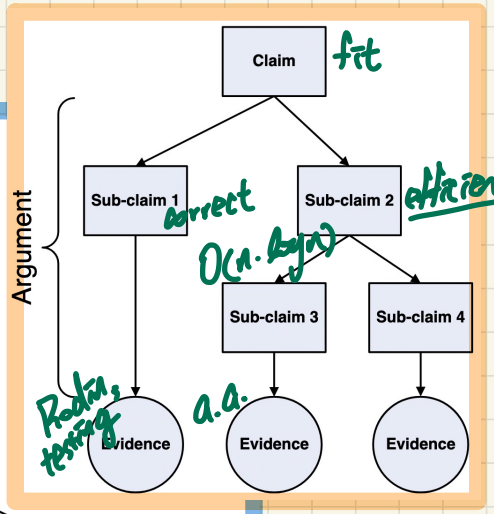


for validation, critical on what's given.

Certifying Systems: Assurance Cases

Research on "Assurance Cases" if interested!





Source: https://resources.sei.cmu.edu/asset_files/whitepaper/2009_019_001_29066.pdf

Lecture 3 - Sep. 12

Introduction, Math Review

Model-Based Development
Propositional Logic
⇒ : Analogy, Truth Table, Alternative Terms
Universal vs. Existential Quantification

Announcements/Reminders

- Priority: **Lab 1**
- Study along with the **Math Review** lecture notes.

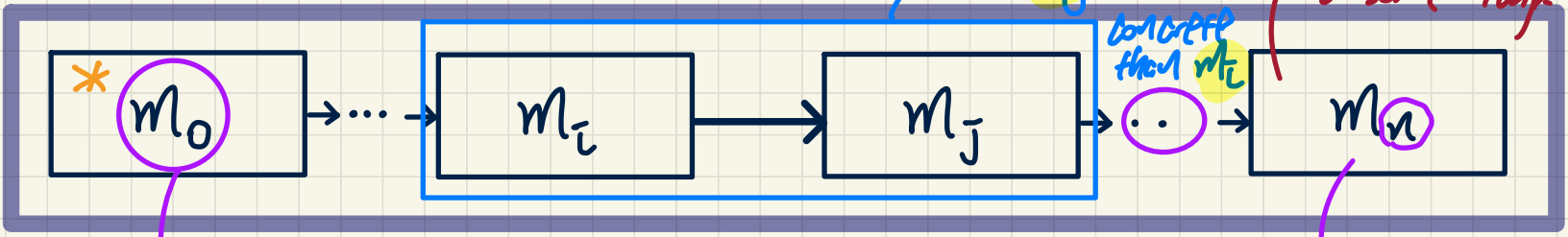
< m_0 , m_1 , m_2 >

Correct by Construction

n+1 models

1. m_i is more abstract than m_j
2. m_j is more concrete than m_i

straightforward to adapt to some imp.



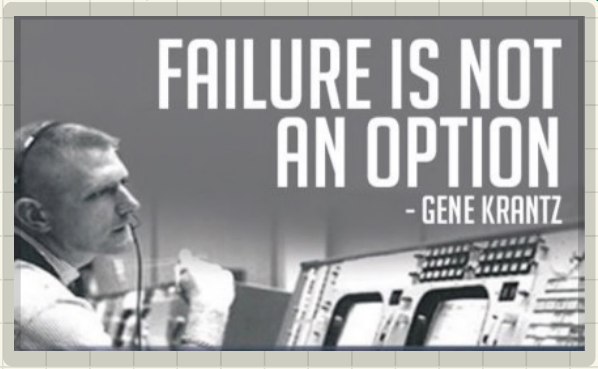
initial model (most abstract ⇒ simplest ⇒ faltered on the most details about imp.) least concrete

* Having a single "superman" model and verify it is not feasible

final model least abstract most concrete

⇒ most sophisticated

⇒ contains the most imp. details



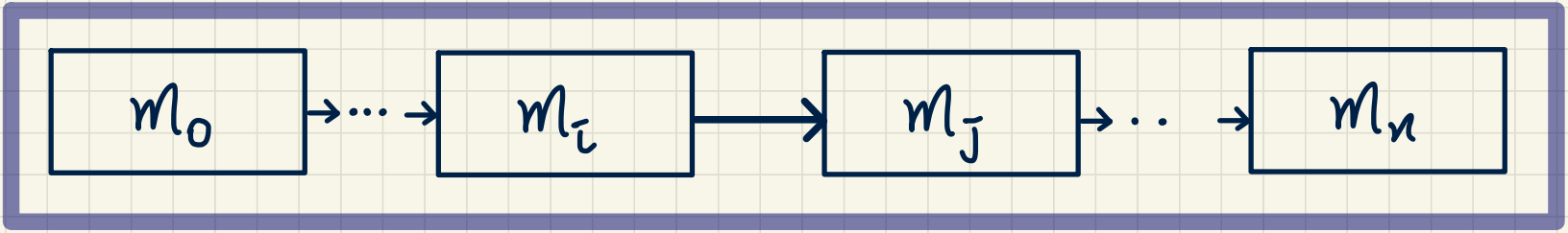
3. m_i is refined by m_j
4. m_j refines m_i

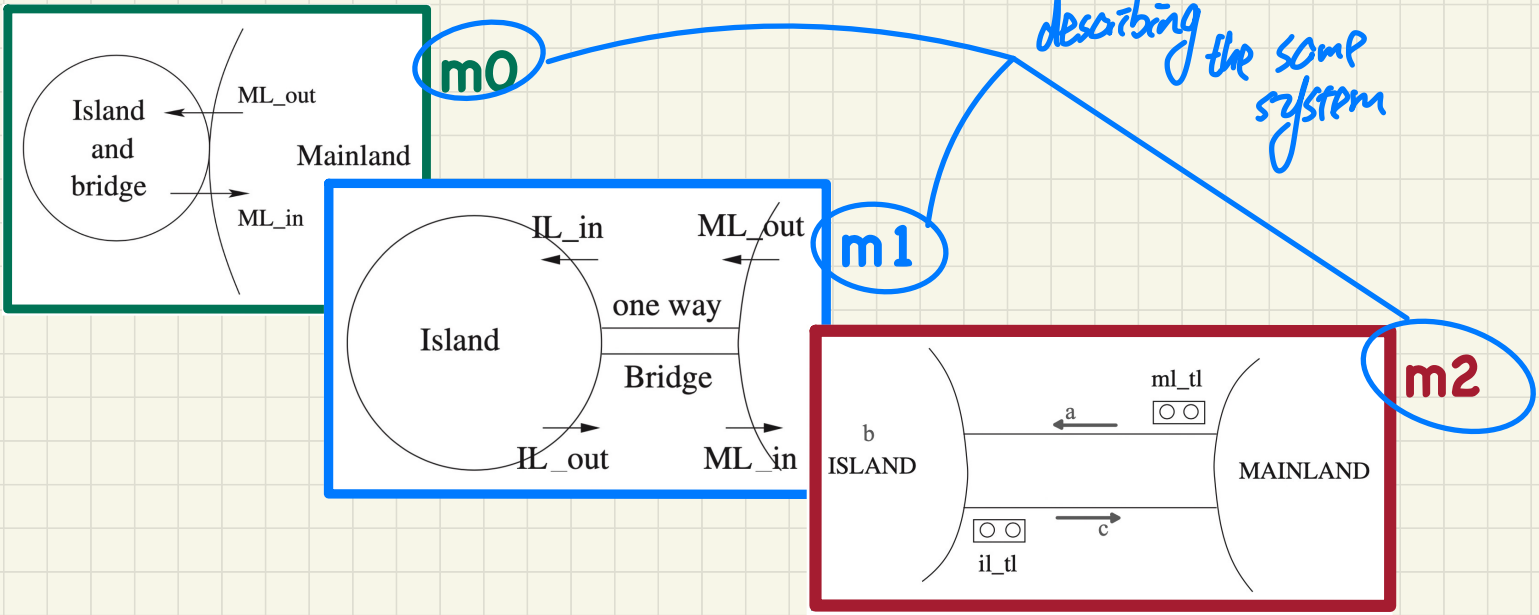
To establish this formally

1. State a set of proof obligations (POs)
2. prove/disprove POs

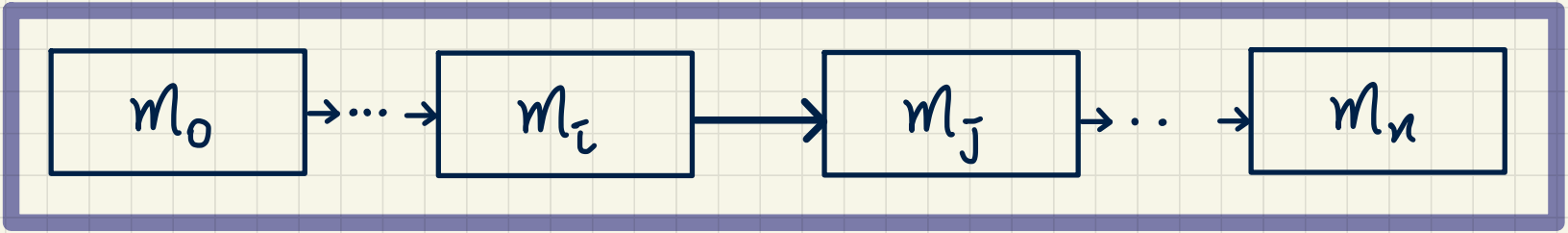
Source: https://audiobookstore.com/audiobooks/failure-is-not-an-option-1.aspx

Correct by Construction: Bridge Controller System



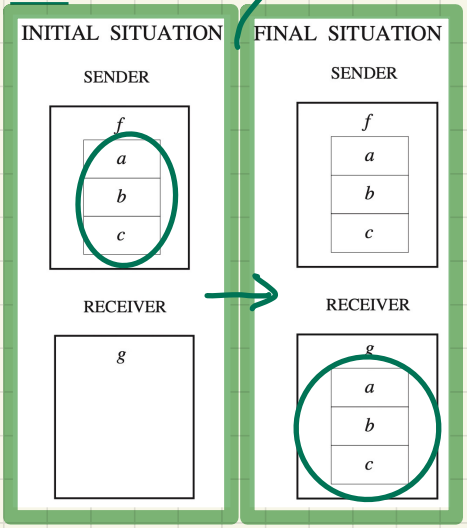


Correct by Construction: File Transfer Protocol

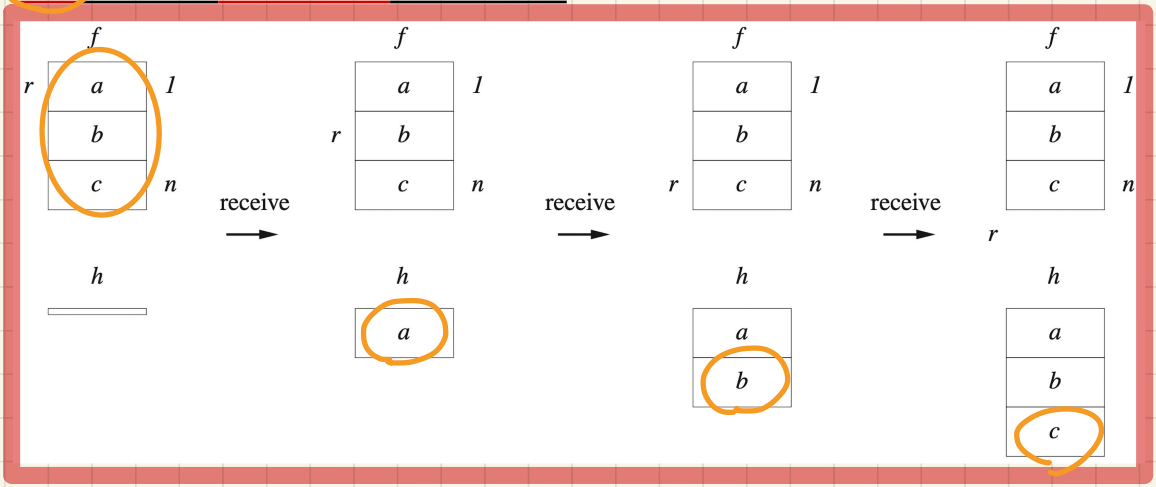


m0

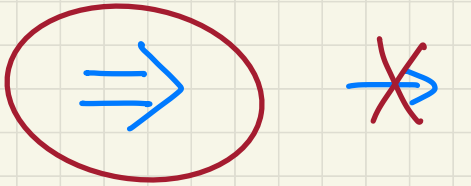
abstraction: instantaneous transmission

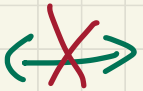


m1: more concrete than m0

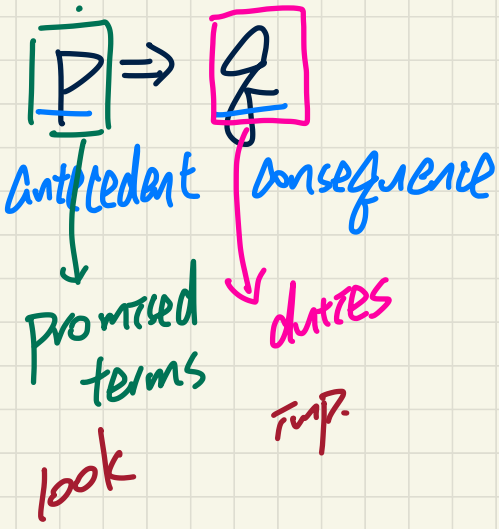


Implication :





Implication ≈ Whether a Contract is Honoured



T	⇒	T	≡ T
T	⇒	F	≡ F
F	⇒	T	≡ T
F	⇒	F	≡ T

Contract is not considered violated

zero of ⇒ : F ⇒ P ≡ T

identity of ⇒ : T ⇒ P ≡ P

Expressing Implications

p: snow storm
q: cancel class

given that p is true, the only way for $p \Rightarrow q$ to be true

q if p, p is sufficient for q

p	q	p ⇒ q
true	true	true
true	false	false
false	true	true
false	false	true

is when q is also true when ¬q and ¬(¬p)

$p \Rightarrow q$ is false

q unless ¬p

cond. 1 cond. 2

p	q	p ⇒ q
true	true	true
true	false	false
false	true	true
false	false	true

p only if q, q is necessary for p

p	q	p ⇒ q
true	true	true
true	false	false
false	true	true
false	false	true

E.g. $x > 0$ if $y \leq 10$

$y \leq 10 \Rightarrow x > 0$

$p \Leftrightarrow q$

$p \Leftrightarrow q$ 1. $p \Leftarrow q$ (p if q)

2. $p \Rightarrow q$ (p only if q)

$p \Rightarrow q$

(1) Inverse: $\neg p \Rightarrow \neg q$

(2) Converse $q \Rightarrow p$

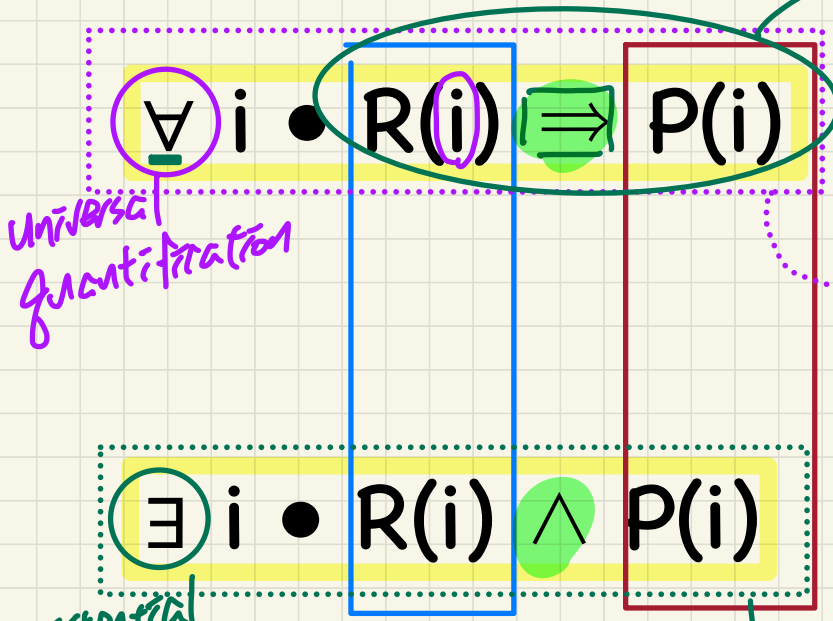
(3) Contrapositive: $\neg q \Rightarrow \neg p$

(Inverse of Converse)

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

Predicate Logic: Quantifiers

- syntax
- base cases in programming



what makes this T?

property to be asserted.

for each i,
if i satisfies R,
then i satisfies P.

what happens if R(i) corresponds to an empty range?

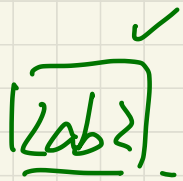
there's at least one i,
s.t. i satisfies R and
i satisfies P.

Lecture 4 - Sep. 17

Math Review

Logical ($\wedge$) vs. Programming (&&)
False Range ¬ R(x) vs. Empty Array
Proof Strategies of Quantifiers

Announcements/Reminders

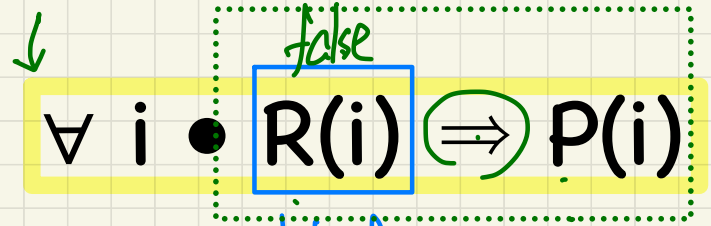


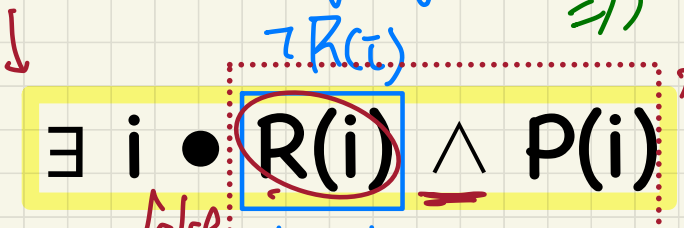
- Lab1 due this Friday at noon.
- Scheduled lab sessions tomorrow.
- Study along with the Math Review lecture notes.

Predicate Logic: Quantifiers

$\forall i \cdot Q(i)$
$\exists i \cdot Q(i)$

- syntax
- base cases in programming





input

can you find a # in a that shows otherwise?

```
boolean allPositive(int[] a){
  if(a.length == 0){return ① true;}
}
```

∵ no witness can be found in empty array to prove otherwise

```
boolean somePositive(int[] a){
  if(a.length == 0){return ② false;}
}
```

∵ no witness can be found in empty array to prove.

Logical Operator vs. Programming Operator

p	q	$p \wedge q$	$p \vee q$
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

short circuit

p && q

↳ evaluate from L to R

↳ if L evaluates to false

↳ evaluation of R is skipped

Q. Are the $\wedge$ and $\vee$ operators equivalent to, respectively, && and || in Java?

$p \wedge q$

① both p and q are well-defined

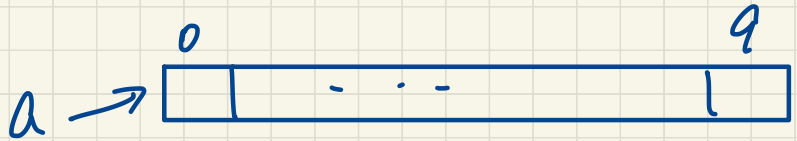
② "evaluate" both sides separately.

commutative

Math: $p \wedge q \equiv q \wedge p$

Java: p && q q && p

Accessing Array



a.length == 10

(1) i < a.length && a[i] > 10 && i >= 0

① ② ③

say i == -2 (fast)

① i < a.length (T)

-2

② a[-2] > 10

crash!

say i == 100

① 100 < a.length (F)

10

a[i] > 10 is skipped and expression just evaluates to false.

Math

i < a.length ∧ i >= 0 ∧ a[i] > 10

i < a.length ∧ i >= 0 ⇒ a[i] > 10

ill-defined expression

✓ (2) i < a.length && i >= 0 && a[i] > 10

guarding constraint.

$\mathbb{Z}$ set of integers

$$-\infty, \dots, -1, 0, 1, \dots, +\infty$$

$\mathbb{N}$ set of natural numbers

$$0, 1, 2, \dots, +\infty$$

$\mathbb{N}_1$

$$1, 2, \dots, +\infty$$

$$\forall \bar{i}, \bar{j} \bullet \bar{i} \in \mathbb{N} \wedge \bar{j} \in \mathbb{Z} \Rightarrow P(\bar{i}, \bar{j})$$

need to consider all combinations of $(\bar{i}, \bar{j})$.

Logical Quantifiers: Examples

$\forall\, i \bullet i \in \mathbb{N} \Rightarrow i \geq 0$ (T)

$i \in \{0, 1, \ldots, +\infty\}$

$\forall\, i \bullet i \in \mathbb{Z} \Rightarrow i \geq 0$ false witness: -1

$-1 \in \mathbb{Z}$ (T) $\Rightarrow$ $-1 \geq 0$ (F)

(F)

$\forall\, i, j \bullet i \in \mathbb{Z} \wedge j \in \mathbb{Z} \Rightarrow i < j \vee i > j$

False witness $i = j$ $i = 1$, $j = 1$

$\exists\, i \bullet i \in \mathbb{N} \wedge i \geq 0$ (T) witness: 0

$\exists\, i \bullet i \in \mathbb{Z} \wedge i \geq 0$ (T) witness: 3

$\exists\, i, j \bullet i \in \mathbb{Z} \wedge j \in \mathbb{Z} \wedge (i < j \vee i > j)$ (T) witness: $i = 2$, $j = 3$

Logical Quantifiers: Examples

Goal: show $R(\bar{i}) \Rightarrow P(\bar{i}) \equiv$ true

How to **prove** $\forall\ i \bullet R(i) \Rightarrow P(i)$?

trivial (1) show $\neg R(\bar{i})$ ∵ zero of $\Rightarrow$: false $\Rightarrow P \equiv$ true

harder (2) show $R(\bar{i}), P(\bar{i})$ (e.g. all elements in a non-empty array are positive)

How to **prove** $\exists\ i \bullet R(i) \wedge P(i)$?

Goal: show $R(\bar{i}) \wedge P(\bar{i}) \equiv$ true. (1) give a witness $\bar{j}$, s.t. $R(\bar{j})$ and $P(\bar{j})$

Goal: show $R(\bar{i}) \Rightarrow P(\bar{i}) \equiv$ false not hard

How to **disprove** $\forall\ i \bullet R(i) \Rightarrow P(i)$?

not hard. (1) show $R(\bar{i}), \neg P(\bar{i})$

give a witness $\bar{j}$, s.t. $R(\bar{j})$ but $\neg P(\bar{j})$

How to **disprove** $\exists\ i \bullet R(i) \wedge P(i)$?

Goal: show $R(\bar{i}) \wedge P(\bar{i})$ is false.

trivial (1) show $\neg R(\bar{i})$: false $\wedge P \equiv$ false

harder (2) $R(\bar{i}), \neg P(\bar{i})$ for those $\bar{i}$ satisfying $R(\bar{i})$, it's the case $\neg P(\bar{i})$.

Prove/Disprove Logical Quantifications

all x in $1..10$ satisfies this.

- Prove or disprove: $\forall x \bullet (x \in \mathbb{Z} \land 1 \le x \le 10) \Rightarrow x > 0.$

$x \in 1..10$ (non-empty range $R(x)$ not false).

- Prove or disprove: $\forall x \bullet (x \in \mathbb{Z} \land 1 \le x \le 10) \Rightarrow x > 1.$

Exercise.

- Prove or disprove: $\exists x \bullet (x \in \mathbb{Z} \land 1 \le x \le 10) \land x > 1.$

By choosing a witness 1, which:

$T \land (1 \in \mathbb{Z} \land 1 \le 1 \le 10 \land (1 > 1))$ F

F

non-empty range.

witness: 2

- Prove or disprove that $\exists x \bullet (x \in \mathbb{Z} \land 1 \le x \le 10) \land x > 10$?

Exercise.

not sufficient by just one witness to disprove $\exists$.

Logical Quantifications: Conversions

R(x): x ∈ 3342_class

P(x): x receives A+

$\forall x \cdot Q(x) \Leftrightarrow \neg \exists x \cdot \neg Q(x)$

$(\forall X \bullet R(X) \Rightarrow P(X)) \Leftrightarrow \neg(\exists X \bullet R \wedge \neg P)$

$\forall x \cdot R(x) \Rightarrow P(x)$

$\Leftrightarrow \{ \forall x \cdot Q(x) \Leftrightarrow \neg \exists x \cdot \neg Q(x) \}$

$\neg \exists x \cdot \neg (R(x) \Rightarrow P(x))$

$\Leftrightarrow \{ p \Rightarrow q \equiv \neg p \vee q \}$

$\neg \exists x \cdot \neg(\neg R(x) \vee P(x))$

$\Leftrightarrow \{ \neg(p \vee q) \equiv \neg p \wedge \neg q,\ \neg(\neg p) \equiv p \}$

$\neg \exists x \cdot R(x) \wedge \neg P(x)$

$(\exists X \bullet R \wedge P) \Leftrightarrow \neg(\forall X \bullet R \Rightarrow \neg P)$

Exercise

De Morgan

Lecture 5 - Sep. 19

Math Review

Set Comprehension
Relating Sets vs. Postconditions
Power Set (Enumeration, Cardinality)

Announcements/Reminders

- Lab1 due tomorrow (Friday) at noon.
- Lab2 to be released soon afterwards.

Sets: Definitions and Membership

Ordering: $\{1, 2, 3\} = \{2, 3, 1\}$

No repetition: $\{1, \underline{2}, 3, \underline{2}\}$ ✗

① $\{x \mid 0 \le x \le 2\}$

$= \{0, 1, 2\}$

② $\{(x, y) \mid x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge 1 \le x \le 2 \wedge 3 \le y \le 4\}$

$= \{(1, 3), (1, 4), (2, 3), (2, 4)\}$

Set Comprehension

$\{$ expression representing the form of a member in the resulting set $\mid$ constraint (values satisfying this con. will be included) $\}$ as members

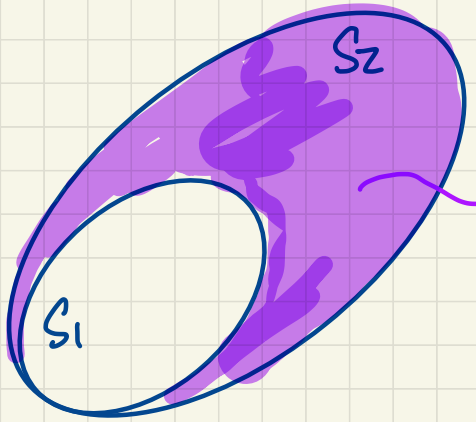
Empty set: $\{\}$ $\emptyset$

$|\emptyset| = 0$

Relating Sets

subset

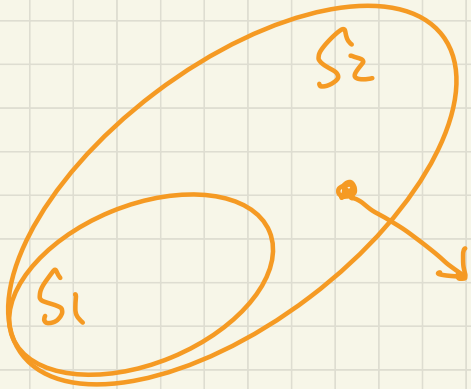
$S_1 \subseteq S_2$



$S_2 \setminus S_1$ may be empty

proper subset

$S_1 \subset S_2$



$|S_2 \setminus S_1| > 0$

Relating Sets: Exercises

$S \subset S$ false

$\emptyset \subset S$ true if $|S| > 0$

false if $S = \emptyset$ $\quad \emptyset \subset \emptyset$ false.

$S_1 = S_2 \iff \underbrace{S_1 \subseteq S_2}_{|S_1| \leq |S_2|} \wedge \underbrace{S_2 \subseteq S_1}_{|S_2| \leq |S_1|}$

Sets: Exercises

$\neg(e \in S)$

Set membership: Rewrite $e \notin S$ in terms of $\in$ and $\neg$

Find a common pattern for defining:

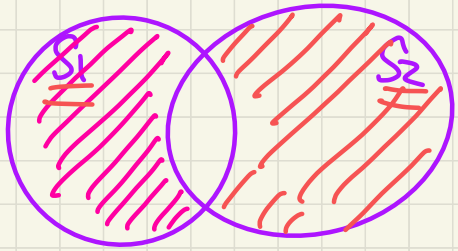
1. $=$ (numerical equality) via $\leq$ and $\geq$
2. $=$ (set equality) via $\subseteq$ and $\supseteq$

$x = y \Leftrightarrow x \leq y \wedge y \leq x$

$S_1 = S_2 \Leftrightarrow S_1 \subseteq S_2 \wedge S_2 \subseteq S_1$

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

	S	T	U
S	$\subseteq$ T $\subset$ F	$\subseteq$ T $\subset$ F	$\subseteq$ F $\subset$ F
T	$\subseteq$ T $\subset$ F	$\subseteq$ T $\subset$ F	$\subseteq$ F $\subset$ F
U	$\subseteq$ T $\subset$ T	$\subseteq$ T $\subset$ T	$\subseteq$ T $\subset$ F



Is set difference ($\setminus$) commutative?

$S_1 \setminus S_2 = S_2 \setminus S_1$

Bidirectional Subset Relations: Programming

```
/* Return the set of positive elements from input. */
HashSet<Integer> allPositive(HashSet<Integer> input)
```

Formulate the `allPositive` method using a **set comprehension**.

input: {2, 3, -1, 4, -2}
allPositive(input) → {2, 3, 4}

input: {2, 3, -1, 4, -2}
output: {6, 8, 9}

Postcondition

$$allPositive(input) = \{ x \mid x \in \mathbb{Z} \wedge x > 0, \}$$

$x \in input$.

If this is inappropriate, give a witness pair of input & output, s.t. output is wrong but postcond. passes.

appropriate?

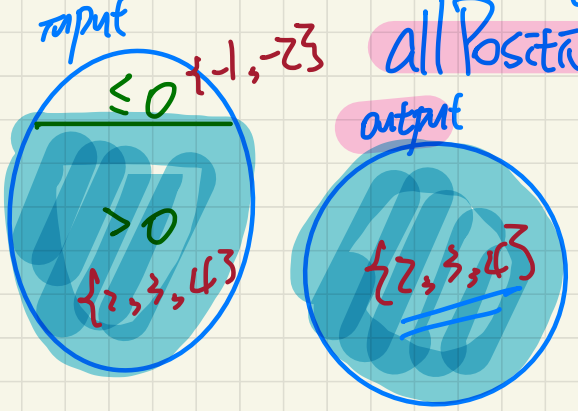
Bidirectional Subset Relations: Programming

```
/* Return the set of positive elements from input. */
HashSet<Integer> allPositive(HashSet<Integer> input)
```

Say:

- S denotes the subset all positive elements from `input`.
- Set `output` denotes the return value from `allPositive`.

Relate the two sets S and output with set operators.



allPositive(input)

Ex.1 Is postcond. appropriate just by (1)

(1) $\{x \mid x \in input \wedge x > 0\} \subseteq output$

(2) $output \subseteq \{x \mid x \in input \wedge x > 0\}$

Ex.2 Is postcond appropriate just by (2)

Bidirectional Subset Relations: Programming

```
/* Return the set of positive elements from input. */
HashSet<Integer> allPositive(HashSet<Integer> input)
```

Say:

- S denotes the subset all positive elements from `input`.
- Set `output` denotes the return value from `allPositive`.

Relate the two sets S and output without set operators.

Ex 3. Express postconditions using $\forall$, $\exists$, $\Rightarrow$, ...

Power Set

$\mathcal{P}(S) = \{ x \mid x \subseteq S \}$ Each member in $\mathcal{P}(S)$ is a subset of S

Calculate the power set of {1, 2, 3}.

1. What's the member in $\mathcal{P}(S)$ that has min card.?
   $\emptyset \in \mathcal{P}(S)$
2. What's the member in $\mathcal{P}(S)$ that has max card.?
   $S \in \mathcal{P}(S)$

$\mathcal{P}(\{1,2,3\})$

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

$= \{ \emptyset,$ card. 0 $\{1\}, \{2\}, \{3\},$ card. 1 $\{1,2\}, \{2,3\}, \{1,3\},$ card. 2 $\{1,2,3\}$ card. 3 $\}$

subsets of card. 1 $= \binom{3}{1}$

subsets of card. 2 $= \binom{3}{2}$

Given a set S, formulate the cardinality of its power set.

$\mathcal{P}(S)$

$$\binom{n}{\bar{c}} = \frac{n!}{(n-\bar{c})!\,\bar{c}!}$$

e.g. $|S|$

$\bar{c}$ → card. of the subset

e.g. $\binom{100}{2}$?

$$\binom{10}{2} = \frac{10!}{(10-2)!\,2!} = \frac{10 * 9}{2!}$$

2 terms counting from 10.

e.g. given $\{1,2,3,4\}$

$$\binom{|\{1,2,3,4\}|}{2}$$

# of subsets of $\{1,2,3,4\}$ that are of card. 2.

Cardinality of Power Set: Interpreting Formula

$\binom{n}{0} = 1$

$\binom{n}{n} = 1$

- Calculate by considering subsets of various cardinalities.
- Calculate by considering whether a member should be included.

Want to know: $|\mathcal{P}(S)|$

$$|\mathcal{P}(S)| = \binom{|S|}{0} + \binom{|S|}{1} + \binom{|S|}{2} + \cdots + \binom{|S|}{|S|-1} + \binom{|S|}{|S|} = 2^{|S|}$$

$\binom{|S|}{0}$: $\emptyset$

$S = \{a, b, c\}$

$|\mathcal{P}(S)| =$

$|2| * (2) * (2)$

include a or not; include b or not; include c or not

a	b	c	
0	0	0	exclude the member from S
0	0	1	$\{c\}$
0	1	0	
1	1	1	include member from S $\{a, b, c\}$

2^3 $2^{|S|}$

Lecture 6 - Sep. 24

Math Review

Incompleteness of Postcondition
Interpreting the choose operator
Cross Product
Relation

Announcements/Reminders

- Lab1 solution released
- Lab2 released

Bidirectional Subset Relations: Programming

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

```
/* Return the set of positive elements from input. */
HashSet<Integer> allPositive(HashSet<Integer> input)
```

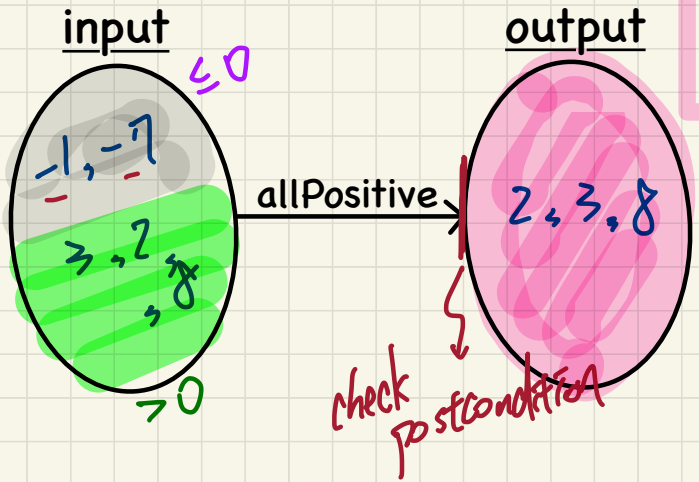
Say:

- S denotes the subset all positive elements from `input`.
- Set `output` denotes the return value from `allPositive`.

Relate the two sets S and output with set operators.

Imp. (faulty)

postcond. (R1) incomplete if it evaluates to true despite the imp. being faulty



Postcondition checks to see if the output is **correct** w.r.t. the input.

(R1) $\{x \mid x \in input \wedge x > 0\} \subseteq output$

(R2) $output \subseteq \{x \mid x \in input \wedge x > 0\}$

entire postcond.

Q1: Why is (R1) alone **incomplete**?

Q2: Why is (R2) alone **incomplete**?

witness: input: $\{-1, -7, 3, 2, 8\}$ output: $\{3, 2, 8\}$ 10?

$\forall \exists \Rightarrow$

(R1) $\{x \mid x \in \text{input} \wedge x > 0\} \subseteq \text{output}$

(R2) $\text{output} \subseteq \{x \mid x \in \text{input} \wedge x > 0\}$

faulty imp.

input: $\{-1, -7, 3, 2, 8\}$

output: $\{2, 3, 8, 10\}$

(R1) T

(R2) F

witness to show that R1 alone as the postcondition is incomplete.

incomplete

What if (R2) alone is the postcond.?

input: $\{-1, -7, 3, 2, 8\}$

~~output: $\{2, 1, 3\}$~~ output: $\{2, 3\}$

satisfies R2 but not R1

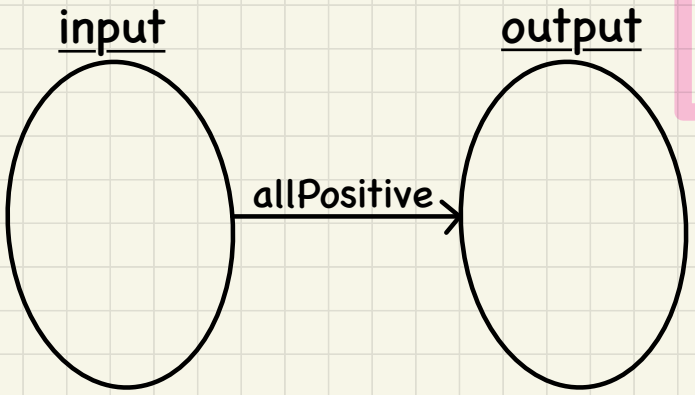
Bidirectional Subset Relations: Programming

```
/* Return the set of positive elements from input. */
HashSet<Integer> allPositive(HashSet<Integer> input)
```

Say:

- S denotes the subset all positive elements from `input`.
- Set `output` denotes the return value from `allPositive`.

Relate the two sets S and output ~~with~~ without set operators.



Postcondition checks to see if the output is **correct** w.r.t. the input.

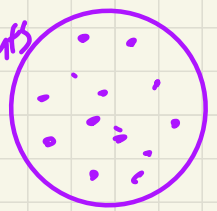
(R1) $\forall\ x \bullet x \in \text{input} \land x > 0 \Rightarrow x \in \text{output}$

(R2) $\forall\ x \bullet x \in \text{output} \Rightarrow x \in \text{input} \land x > 0$

Q1: Why is (R1) alone **incomplete**?

Q2: Why is (R2) alone **incomplete**?

$\binom{n}{r}$ out of n given elements, how many ways are there to make a set of card. r



make a seq of size r

$n \quad (n-1) \quad (n-2) \quad \cdots \quad r$

$= \quad = \quad = \quad = \quad =$

eliminate duplicates

choose operator

$= \dfrac{n!}{(n-r)!\,r!}$

$\binom{4}{2}$ $\quad$ 4 x 3 $\quad$ eliminate duplicates $2!$

a b c d

a b a c b c

$\binom{10}{2} = \dfrac{10!}{(10-2)!\,2!}$

$= \dfrac{n \cdot (n-1) \cdot (n-2) \cdots \_\_\_}{r!}$ $\quad$ r terms $\quad$ $n-r+1$

eliminate duplicates $r!$

$= \dfrac{10!}{8!\,2!}$ $\quad$ 2 terms.

$= \dfrac{10 * 9}{2!}$

Set of Tuples

$$|S_1 \times S_2 \times \cdots \times S_n| = |S_1| \times |S_2| \times \cdots \times |S_n|$$

Given n sets $S_1, S_2, \ldots, S_n$, a ***cross/Cartesian product*** of theses sets is a set of n-tuples.

Each ***n-tuple*** $(e_1, e_2, \ldots, e_n)$ contains n elements, each of which a member of the corresponding set.

each member in the resulting set is a n-tuple.

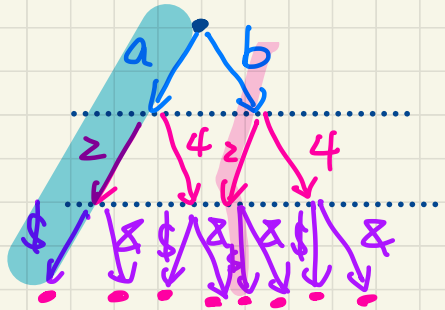
$$S_1 \times S_2 \times \cdots \times S_n = \{(e_1, e_2, \ldots, e_n) \mid e_i \in S_i \wedge 1 \le i \le n\}$$

Example: Calculate $\{a, b\} \times \{2, 4\} \times \{\$, \&\}$

S_1 S_2 S_3

$$= \{(e_1, e_2, e_3) \mid e_1 \in \{a, b\} \wedge e_2 \in \{2, 4\} \wedge e_3 \in \{\$, \&\}\}$$

$$= \{(a, 2, \$), (b, 2, \$), \ldots\}$$



Relation: set of ordered pairs.

e.g. a relation on $\{1, 2, 3\}$ (S) and $\{a, b\}$ (T)...

- Is $(1, a)$ a relation on S and T?

  No! $\because (1, a)$ is not a set.

- Is $\{(1, 2)\}$ a relation on S and T?

  No! the order is wrong.

- Is $\{(1, a), (3, b)\}$ a relation on S and T?

  YES!

  $R_1 = \{(1, a), (3, b)\}$

  $R_2 = \{(3, b), (1, a)\}$

  $R_1 = R_2$

What's the min relation on S and T? $\emptyset$

What's the max relation on S and T? $S \times T$

Lecture 7 - Sep. 26

Math Review

Constructing All Possible Relations
Domain, Range, Inverse
Domain/Range Restriction/Subtraction
Relational Image

Announcements/Reminders

- Lab2 released
- Guide for ~~Written Test~~ Programming 1 to be released next Tue
- Remote TA Support

$\emptyset \in S \leftrightarrow T$ ✓ $\{\emptyset\} \in S \leftrightarrow T$ ✗

Set of Possible Relations

Each member in $S \leftrightarrow T$ is a relation, meaning that each member is a set of pairs.

$\{(a,2)\} \in S \leftrightarrow T$

- **Set** of possible relations on S and T: $\emptyset \in S \leftrightarrow T$, $S \times T \in S \leftrightarrow T$
- Dedicated symbol for set of possible relations on S and T: $S \leftrightarrow T$
- Declare that set r is a relation on S and T:

Example: Enumerate all relations on {a, b} and {2, 4}.

S T

Hint: How many? $|S \leftrightarrow T|$

$S \leftrightarrow T = \mathbb{P}(S \times T)$

$|S \times T|$

set of sets of pairs = relations

$S \times T \in S \leftrightarrow T$

$S \times T$: set of pairs (a single rel.)

$\{\}$ $\emptyset$ relation of card. 0 $\binom{4}{0} = 1$

$\{(a,2)\}, \{(a,4)\}, \{(b,2)\}, \{(b,4)\}$ relations of card. 1 $\binom{4}{1} = 4$

$\{(a,2),(a,4)\}, \ldots$ relations of card. 2 $\binom{4}{2} = \frac{4 \times 3}{2} = 6$

$\{(a,2),(a,4),(b,2),(b,4)\}$ relation of card. 4 $\binom{4}{4} = 1$

$S \times T$

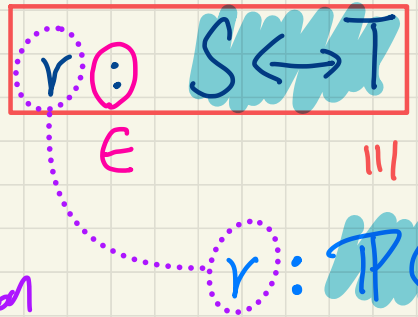
relations of card. 3 $\binom{4}{3} = 4$

Given S, T

$$S \leftrightarrow T = \mathbb{P}(S \times T)$$

$r \subseteq S \leftrightarrow T$

r is a set of relations.



r is a single relation

Exercise

Departure = { toronto, montreal, vancouver }

Destination = { beijing, seoul, pencyg }

airline $\in$ Departure $\leftrightarrow$ Destination

$= \mathbb{P}$(Departure $\times$ Destination)

$|\_| = 9$

How many?

1) enumerate.

$2^{|S \times T|} = 2^9 = 512$

Relational Operations: Domain, Range, Inverse

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

$r \in \text{Alphabet} \leftrightarrow \mathbb{Z}$

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

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)}

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

Exercise: Relate the domains and ranges of r and its inverse.

$r^{\sim}$

$dom(r) = ran(r^{\sim})$

$ran(r) = dom(r^{\sim})$

] algebraic properties

Relational Operations: Image

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

$r[\{a,b\}] = \{ r' \mid (d, r') \in r \wedge d \in \{a,b\} \} = \{1,2,4,5\}$

$\subseteq ran(r)$

$\subseteq$

$r \in S \times T$

$r[S]$ assumption: $S \subseteq S$

Exercises

- Image of {a, b} on r? ✓
- ✓ Image of {1, 2} on r? undefined!
- Image of {1, 2} on the inverse of r? $r^{\sim}[\{1,2\}] = \{a,b,d,e\}$
- Calculate r's range via an image. $ran(r) = r[dom(r)]$
- Calculate r's domain via an image. $dom(r) = r^{\sim}[ran(r)]$

$dom(r^{\sim})$

$$r \in S \leftrightarrow T$$
$$s \subseteq S \quad t \subseteq T$$

	domain	range
Restriction	$s \lhd r$	$r \rhd t$
Subtraction	$s \lhd\!\!\!- r$	$r \rhd\!\!\!- t$

Each of these operators returns a new relation.

$$ds \lhd r = \{(d, r') \mid (d, r') \in r \wedge d \in ds\}$$

a new relation (separate from r)

domain res.

Relational Operations: Restrictions vs. Subtractions

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

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

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

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

Relational Operations: Overriding

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

Example: Calculate r overridden with {(a, 3), (c, 4)}

Hint: Decompose results to those in t's domain and those not in t's domain.

$$r \lhd t = \{(d, r') \mid (d, r') \in t \vee ((d, r') \in r \wedge d \notin dom(t))\}$$

r: relation; t: relation

Lecture 8 - Oct. 1

Math Review, Lab1 Solution, Lab2

Algebraic Properties of Relational Ops
Lab2: Celebrity_0
Functional Properties

Announcements/Reminders

- Lab2 due this Friday
- Guide for Programming Test 1 released

Relational Operations: Overriding

$\{(x,y) \mid P_1(x,y) \vee P_2(x,y)\} = \{(x,y) \mid P_1(x,y)\} \cup \{(x,y) \mid P_2(x,y)\}$

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

Example: Calculate r overridden with {(a, 3), (c, 4)}

Hint: Decompose results to those in t's domain and those not in t's domain.

$r \lhd t = \{(d, r') \mid (d, r') \in t \vee ((d, r') \in r \wedge d \notin dom(t))\}$

relation relation

rel r is overridden by rel t $= \{(d, r') \mid (d, r') \in t\} \cup \{(d, r') \mid (d, r') \in r \wedge d \notin dom(t)\}$

$= t \cup (dom(t) \lhd r)$

$\{(d, r') \mid (d, r') \in r \wedge d \notin dom(t)\} = \{(b,2), (b,5), (d,1), (e,2), (f,3)\}$

r = subtraction

$r \lhd \{(a,3), (c,4)\} = \{(a,3), (c,4)\} \cup (\{a,c\} \lhd r) = \square$

t; t; $(dom(t) \lhd r)$

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

(a, 3) , (c, 4)

Example: Calculate r overridden with {(a, 3), (c, 4)}

b ∈ Account ↔ ℤ

basically r , except all pairs with first elements in the domain of t should agree with t.

t: transfer.

changes = {(a1, b(a1) − amt), (a2, b(a2) + amt)}

THINK: transfer event from (a b)

Exercises: Algebraic Properties of Relational Operations

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

r[s] $s = \{a, b\}$

Define the **image** of set s on r in terms of other relational operations.

Hint: What range of value should be included?

$r[s] = ran(s \lhd r)$

$\subseteq ran(r)$

goes a relation

Define r **overridden with** set t in terms of other relational operations.

Hint: To be in t's domain or not to be in t's domain?

$r \lhd t = t \cup (dom(t) \lhd r)$

$r \in S \leftrightarrow T$ $\quad S = \{a, b, c, d\}$ $\quad T = \{1, 2, 3, 4\}$

$s \subseteq S \quad t \subseteq T$

Problem: reconstruct relation r in terms of restriction & subtraction.

$$r \stackrel{?}{=} (s \lhd r) \cup (s \lhd\!\!\!\!- r)$$

$$r \stackrel{?}{=} (r \rhd t) \cup (r \rhd\!\!\!\!- t)$$

Examples

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

$s = \{b, d\}$

$t = \{1, 4\}$

Lab2: Relational Operators

Among persons, there is a special one: the celebrity. We define the *knows* relation which captures how the celebrity and other persons should be treated differently:

1. No person (including the celebrity) knows himself or herself.
2. The celebrity knows no one.
3. Everyone know the celebrity.

CONTEXT Celebrity_c0

CONSTANTS

k knows relation

c celebrity

P Set person

AXIOMS

axm1: $P \subseteq \mathbb{N}$

axm2: $c \in P$

axm3: $k \in (P \setminus \{c\}) \leftrightarrow P$

axm4: $k^{-1}[\{c\}] = P \setminus \{c\}$

axm5: $k \cap id = \varnothing$

END

MACHINE Celebrity_0

SEES Celebrity_c0

VARIABLES

r

INVARIANTS

inv1: $r \in P$

EVENTS

Initialisation

begin

act1: $r :\in P$

end

Event celebrity ⟨ordinary⟩ $\widehat{=}$

begin

act1: $r := c$

end

END

Axm 4: $K^{-1}[\{c\}] = P \setminus \{c\}$

$K^{-1}[\{c\}]$: celebrity is known by

$P \setminus \{c\}$: every person except the celebrity themselves

tom

$\tilde{K} = \{(Mark, Alan), (Tom, Alan), (Tom, Mark)\}$

$K^{\sim}[\{Tom\}] = \{Alan, Mark\}$

$K^{\sim}[\{Tom\}]$: celebrity Tom is known by

$\{Alan, Mark\}$: everyone except Tom

Axm 3 $k \in (P \setminus \{c\}) \leftrightarrow P$

$\Vert\!\vert$

$k \in P \leftrightarrow P$

$c \notin dom(k)$

Functional Property

$r \in S \leftrightarrow T$

isFunctional(r) $\Leftrightarrow$

$\forall s, t1, t2 \bullet$

$(s \in S \land t1 \in T \land t2 \in T)$

dom val ran val.

$\Rightarrow$

two pairs sharing the same 1st value

$((s, t1) \in r \land (s, t2) \in r \Rightarrow t1 = t2)$

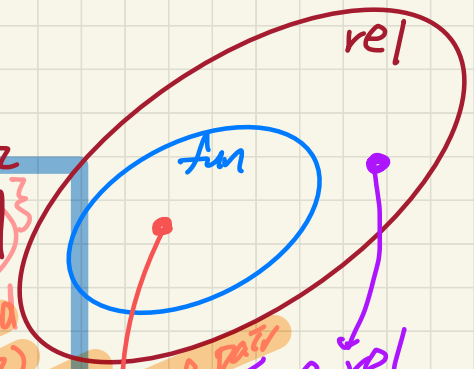
the same dom value cannot map to distinct values in the range.

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

s t1 s t2

(s,t1) and (s,t2) refer to the same pair

2nd values are the same



some rel not qualified to be a fun.

a rel and a function at the same time.

Q: Smallest relation satisfying the functional property. $\emptyset$

Q: How to prove or disprove that a relation r is a function.

Q: Rewrite the functional property using contrapositive.

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

$r = \{ (a, 1), (b, 2), (c, 1) \}$

1. Exercise: check this against the func. property
   → satisfied
2. injective function

Lecture 9 - Oct. 3

Math Review

Partial Function vs. Total Function
Relational Img vs. Function App
Modelling Decision: Rel, Pfun, Tfun

Announcements/Reminders

- Lab2 due tomorrow at **noon**
- Guide for **Programming Test 1** released

Functional Property

* Each domain value maps to at most one range value

** Two distinct values in the range cannot be mapped by the same domain value.

$\{(a,1), (a,+)\}$

isFunctional(r) $\Leftrightarrow$

$$\forall\, s, t1, t2 \bullet$$

$$(\, s \in S \wedge t1 \in T \wedge t2 \in T \,)$$

$$\Rightarrow$$

** $t1 \neq t2 \Rightarrow \neg((s,t_1) \in r \wedge (s,t_2) \in r)$

*

$$(\, (s, t1) \in r \wedge (s, t2) \in r \Rightarrow t1 = t2 \,)$$

- Q: Smallest relation satisfying the functional property. $\emptyset$
- Q: How to **prove** or **disprove** that a relation r is a function.
- Q: Rewrite the functional property using **contrapositive**.

$P \Rightarrow Q \equiv \neg Q \Rightarrow \neg P$

Prove
① Trivial
Show that there's no such $(s,t_1) \wedge (s,t_2)$ in r ($r = \emptyset$)

Disprove find a witness which shows that $r1 \neq r2$ $(d,r_1) \in r$ and $(d,r_2) \in r$

② $r \neq \emptyset$
check to make sure no dom value maps to distinct ran. values.

Partial Functions vs. Total Functions

↛ partial
→ total

set of all possible partial functions

$r \in S \nrightarrow T \Leftrightarrow (\text{isFunctional}(r) \wedge \text{dom}(r) \subseteq S)$ → $\text{dom}(r) \subset S \vee \text{dom}(r) = S$ (total function)

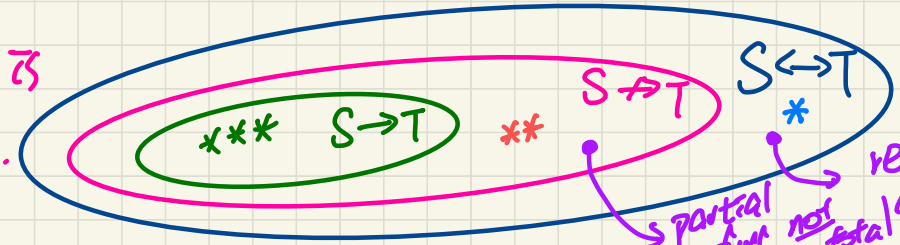
$r \in S \rightarrow T \Leftrightarrow (\text{isFunctional}(r) \wedge \text{dom}(r) = S)$

Exercise: Visualize $S \nrightarrow T$ vs. $S \rightarrow T$

① $r \in S \rightarrow T \Rightarrow r \in S \nrightarrow T \wedge r \in S \leftrightarrow T$

② $r \in S \nrightarrow T \Rightarrow r \in S \leftrightarrow T$

Every function is a partial func.



e.g., { {(2, a), (1, b)}, {(2, a), (3, a), (1, b)} } ⊆ {1, 2, 3} ⇸ {a, b}

only need to check func. prop.

e.g., *** {(2, a), (3, a), (1, b)} ∈ {1, 2, 3} → {a, b}

check ① fun. prop. ② dom(r) = S

e.g., ** {(2, a), (1, b)} ∉ {1, 2, 3} → {a, b}

partial func not total

e.g., * {(2, a), (1, b), (3, a), (1, a)} ∉ {1, 2, 3} → {a, b}

violate func. prop.

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

$T = \{a, b\}$

n^2.

Correct $O(n^3)$ but not tightest

Correct & tightest $O(n^2)$.

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

① It's correct to say r is total. (and most accurate)

② It's correct to say r is partial (but not accurate)

$S = \{1, 2\}$

$T = \{a, b, c\}$

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

↳ sat. func. prop. $\Rightarrow$ partial function $\in S \nrightarrow T$

↳ $dom(r) = \{1, 2\} = S$ $\Rightarrow$ total function $\in S \rightarrow T$

Relational Image vs. Functional Application

A function is a relation.

partial function

$f \in \{1, 2, 3\} \pfun \{a, b\}$

$f = \{ (3, a), (1, b) \}$

mapping for dom value 2 is undefined.

$isFunctional(f) \wedge dom(f) \subseteq \{1,2,3\}$

$dom(f) \subseteq \{1,2,3\}$ — mappings may not exist for certain domain value(s).

$dom(f) = \{1,2,3\}$

Exercises:

$f[\{3\}] = \{a\}$ ($\in dom(f)$)

$f[\{1\}] = \{b\}$

$f[\{2\}] = \{\}$ ($\notin dom(f)$)

(relational images)

treated as a relation

(functional applications)

$f(3) = a$

$f(1) = b$

$f(2) = \bot$ (undefined)

$r[s] = \{ r' \mid (d, r') \in r \wedge d \in s \}$

In Rodin: c is a constant

① $f : \mathbb{Z} \leftrightarrow \mathbb{Z}$

$f[\{c\}]$ → set

always well defined

② $f : \mathbb{Z} \rightarrow \mathbb{Z}$

$f(c)$

well-defined

Rodin generates a proof obligation to ensure that this functional app. is well-defined: $c \in dom(f)$

[ab]

withdraw

guard $a \in dom(b)$

action: $b := \ldots b(a) \ldots$

well-defined

$$r \in S \leftrightarrow T$$

r is a function

each domain value is mapped to at most one value

$$\Rightarrow |r[\{s\}]| \leq |$$

$s \in S$

$$\begin{cases} = 1 & (s \in dom(r)) \\ < 1 \ (=0) & (s \notin dom(r)) \end{cases}$$

Modelling Decision: Relations vs. Functions

An organization has a system for keeping **track** of its employees as to where they are on the premises (e.g., ``Zone A, Floor 23''). To achieve this, each employee is issued with an active badge which, when scanned, synchronizes their current positions to a central database.

Assume the following two sets:

- *Employee* denotes the **set** of all employees working for the organization.
- *Location* denotes the **set** of all valid locations in the organization.

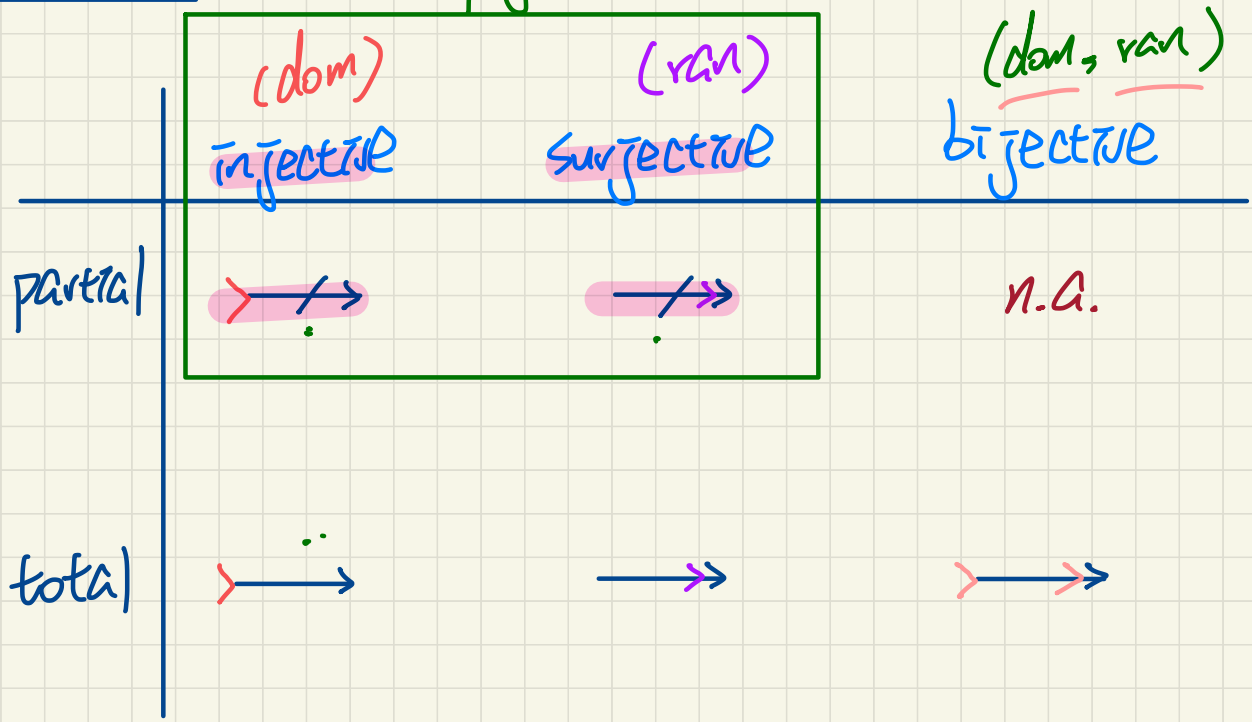
Is $where\_is \in Employee \leftrightarrow Location$ appropriate?

Is $where\_is \in Employee \rightarrow Location$ appropriate?

Is $where\_is \in Employee \nrightarrow Location$ appropriate?

Functions

prog test 1



Review Q & A - Oct. 6

Programming Test 1

PracticeTest1 Solution Walkthrough
Lab1 Solution
Math Review Lecture

Announcements/Reminders

Released:

- Lab2 solutions (PDF & a walkthrough tutorial video)
- PracticeTest1 solution

PracticeTest1 Solution: Context

Test: no Rodin project to import; develop from scratch (and export)

ENV1	A school system is concerned with Students	Finite Carrier Set `STUDENT`
ENV2	A school system is concerned with Courses	Finite Carrier Set `COURSE`
ENV3	A Student has a set of courses they are taking from `COURSE`	inv1: Relation `c`
ENV4	A student can take up to `cap` courses	inv2
ENV5	`cap` is a integer larger than 0	Constant `cap`
ENV6	A school can only accept up to `limit` students	inv3
ENV7	`limit` is a integer larger than 0	Constant `limit`

CONTEXT C0

SETS

STUDENT CARRIER SET

COURSE CARRIER SET

CONSTANTS

cap course cap

limit School Limit

AXIOMS

axm1: $cap \in \mathbb{N}_1$

axm2: $limit \in \mathbb{N}_1$

axm3: $finite(STUDENT)$
needed for welldefinedness of inv3 and inv4

axm4: $finite(COURSE)$
needed for welldefinedness of inv3 and inv4

END

1. `axm1`: Constant `cap` - A Number of courses students can take. `cap` > 0
2. `axm2`: Constant `limit` - A Number of students the school can take.
3. `axm3`: Carrier Set `COURSE` - Finite Set of Abstract Courses
4. `axm4`: Carrier Set `STUDENT` - Finite Set of abstract students, (In and not In school)

$\mathbb{N}$ 0, 1, ...
$\mathbb{N}_1$ 1, 2, ...

$cap > 0$ by itself not a typing constraint

$eecs33447 \in COURSE$ → finite( )

constant or variable → typing constraint

context / machine

Carrier Set: set of certain kind of entities of different kinds

~ no two carrier sets overlap
~ no need to further specify the set's content
~ cannot add/remove to/from carrier set

* alt: ✗ card(STUDENT) ≤ limit → the set of all possible students

PracticeTest1 Solution: Invariants

① typing constraints for variables
② properties (involving constants & variables).

5. **inv1**: Relation **c** - Defined as "Students can take a set of Courses" Maps students in school to a set of courses
6. **inv2**: defines the restriction of ENV4
7. **inv3**: defines the restriction of ENV6 `limit` > 0

c ∈ STUDENT ↔ COURSE ✗
each student is mapped to a set of courses

MACHINE Enrollment0
SEES C0
VARIABLES
c courses student takes
INVARIANTS
inv1: $c \in STUDENT \pfun \mathbb{P}(COURSE)$
Students can take a set of courses
inv3: $\forall a \cdot a \in dom(c) \Rightarrow card(c(a)) \leq cap$
Student Must be taking at most cap courses
inv4: $card(dom(c)) \leq limit$
School limit

relational image?
p. fun.
function application
similar to credit currently admitted
limit in Lab1 students (which may grow for students)
well-defined: $a \in dom(c)$

① ✗ c ∈ STUDENT ↔ COURSE
can this allow any relation that violates reg

② ✓ c ∈ STUDENT ⇸ ℙ(COURSE)
c is a rel, but not accurate for the req.
① e.g. = { (jackie, eecs2020), (jackie, eecs3342) }

② e.g. { (jackie, {eecs2020, eecs3342}) }
a set of courses being a value in ordered pair

PracticeTest1 Solution: Events (Init, Admission, Leave)

Initialisation
begin
act1: $c := \varnothing$
empty set initialization
end

Event admission ⟨ordinary⟩ $\widehat{=}$
Admitting a Student to the School
any
s student being added
where
type_of_s: $s \in STUDENT$
grd1: $s \notin dom(c)$
Make sure student isn't already admitted
grd2: $card(dom(c)) < limit$
Make Sure School Limit isn't reached
then
act1: $c := c \Leftarrow \{s \mapsto \varnothing\}$
Add Student with empty course set
end

Event leave ⟨ordinary⟩ $\widehat{=}$
Remove student once they leave
any
s student leaving
where
type_of_s: $s \in dom(c)$
must be in school
then
act1: $c := \{s\} \lhd c$
remove from relation
end

REQ8	Allow admission of a student. New students have no courses	Event **admission**
REQ9	Allow a student to leave the school	Event **leave**

8. Event **admission**: takes event parameter **s**, the event is enabled when
   8.1 **s** is a student
   8.2 **s** is not already admitted in the school
   8.3 **limit** of the school has not been reached
   If enabled, the event should add a student to the **c** relation with an empty course set.
9. Event **leave**: takes event parameter **s**, the event is enabled when
   9.1 **s** is a student in the school
   If enabled, the event should remove the student from the relation **c**

type

inv4: $dom(c) \leq limit$

make sure that after the extension ≤ should still hold.

var. c is changed ↳ all invariants involving c must be preserved.

e.g. limit = 100 before "admission"

inv. $dom(c) \leq 100$

will be extended by one pair after "admission".

$dom(c) < 100$.

after admission

$dom(c) \leq 100$

c extended with 1 pair

Lab1 Solution: Machine (Variables & Invariants)

MACHINE Bank0

// Initial model of the bank system

SEES C0

VARIABLES

b balance (ENV2)

d cash drawer (REQ7)

owner account owner (ENV9) - Solution to Exercise 4 of Lab1

INVARIANTS

inv1: $b \in ACCOUNT \pfun \mathbb{Z}$

Alt. ✗ $b \in Account \leftrightarrow \mathbb{Z}$

↓

an account may be mapped to more than one value

$b = \{ \ldots, al \mapsto 23, al \mapsto 46, \ldots \}$

inv2: $d \in \mathbb{Z}$

inv3: $\forall a \cdot a \in dom(b) \Rightarrow b(a) \geq -c$

(ENV3)

inv4: $\forall a \cdot a \in dom(b) \Rightarrow b(a) \leq L$

(ENV3) - Solution to Exercise 3 of Lab1

inv5: $owner \in ACCOUNT \pfun PERSON$

(ENV9) - Solution to Exercise 4 of Lab1

inv6: $dom(b) = dom(owner)$

Consistent domains of the balance and owner functions (ENV9) - Solution to Exercise 4 of Lab1 (Note. If we declared this invariant as a theorem, then it must be provable/derivable from other invariants that are declared as axioms, which is not the case. Instead, we also declare this invariant as an axiom (i.e., not as a theorem) so that proof obligations (POs) will be generated regarding it being established (by INITIALIZATION) and preserved (by other events).)

Lab1 Solution: Machine (transfer)

MACHINE Bank0

// Initial model of the bank system

SEES C0

VARIABLES

b balance (ENV2)

d cash drawer (REQ7)

owner account owner (ENV9) - Solution to Exerc

INVARIANTS

inv1: $b \in ACCOUNT \pfun \mathbb{Z}$

inv2: $d \in \mathbb{Z}$

inv3: $\forall a \cdot a \in dom(b) \Rightarrow b(a) \geq -c$
(ENV3)

inv4: $\forall a \cdot a \in dom(b) \Rightarrow b(a) \leq L$
(ENV3) - Solution to Exercise 3 of Lab1

inv5: $owner \in ACCOUNT \pfun PERSON$
(ENV9) - Solution to Exercise 4 of Lab1

inv6: $dom(b) = dom(owner)$

Event transfer ⟨ordinary⟩ $\widehat{=}$

(REQ11) - Solution to Exercise 4 of Lab1

any

a1

a2

v

where

grd1: $a1 \in dom(b)$

grd2: $a2 \in dom(b)$

grd3: $a1 \neq a2$

grd4: $b(a1) - v \geq -c$

grd5: $b(a2) + v \leq L$

grd6: $v \in \mathbb{N}_1$

Necessary to make POs related to inv3/inv4 discharged

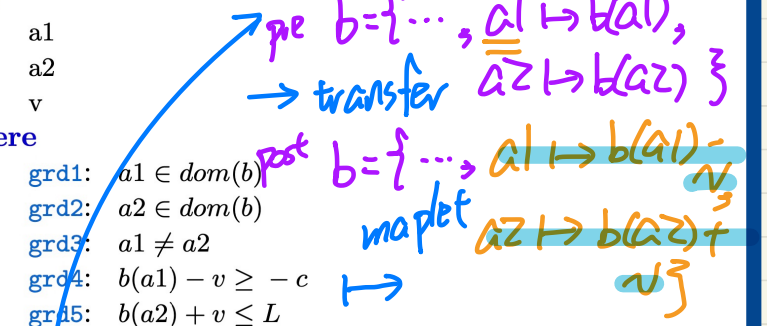
then

act1: $b := b \Leftarrow \{a1 \mapsto b(a1) - v, a2 \mapsto b(a2) + v\}$

Note. It's not allowed to have two actions involving the s
:= ...

end

END



a ↦ b
(a,b).

Rodin
paper.

Lecture 10 - Oct. 8

Math Review, Bridge Controller Intro

Injective vs. Surjective vs. Bijective
Modelling an Array as a Function
?

Announcements/Reminders

- ProgTest1 tomorrow Wednesday
- Lab3 released

total fun ⇒ partial func.

Injective Functions

* it cannot be the case that two distinct domain values map to the same range value.

isInjective(f)

$$\forall s, t_1, t_2 \bullet (s \in S \land t_1 \in T \land t_2 \in T) \Rightarrow ((s, t_1) \in f \land (s, t_2) \in f \Rightarrow t_1 = t_2)$$

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

fun. prop.

inj. prop.

e.g. $(1, b), (3, b)$ witness of violating inj. prop.

true

$(1, b) \in f \land (3, b) \in f$ ⇒ $1 = 3$

false

false

If f is a ***partial injection***, we write: $f \in S \nrightarrow T$

- e.g., $\{ \varnothing, \{(1, \mathbf{a})\}, \{(2, \mathbf{a}), (3, \mathbf{b})\} \} \subseteq \{1, 2, 3\} \nrightarrow \{a, b\}$
- e.g., $\{(1, \mathbf{b}), (2, a), (3, \mathbf{b})\} \notin \{1, 2, 3\} \nrightarrow \{a, b\}$ partial fun. not inj.
- e.g., $\{(1, \mathbf{b}), (3, \mathbf{b})\} \notin \{1, 2, 3\} \nrightarrow \{a, b\}$ partial fun. inj.

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

- e.g., $\{1, 2, 3\} \rightarrowtail \{a, b\} = \varnothing$
- e.g., $\{(2, d), (1, a), (3, c)\} \in \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$
- e.g., $\{(\mathbf{2}, d), (\mathbf{1}, c)\} \notin \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$
- e.g., $\{(2, \mathbf{d}), (1, c), (3, \mathbf{d})\} \notin \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$

** Contrapositive: $s_1 \neq s_2 \Rightarrow \neg((s_1, t) \in f \land (s_2, t) \in f)$ — distinct dom values; s_1 and s_2 cannot map to same ran. value.

the set of all possible total injective functions

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

- e.g., $\{1,2,3\} \rightarrowtail \{a,b\} = \emptyset$ — impossible to construct a total inj. func. S and T
- e.g., $\{(2,d),(1,a),(3,c)\} \in \{1,2,3\} \rightarrowtail \{a,b,c,d\}$
- e.g., $\{(2,d),(1,c)\} \notin \{1,2,3\} \rightarrowtail \{a,b,c,d\}$
- e.g., $\{(2,d),(1,c),(3,d)\} \notin \{1,2,3\} \rightarrowtail \{a,b,c,d\}$

$\{1,2,3\} \rightarrowtail \{a,b\}$

↓ construct injection

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

inj. but not total

to make it total, we can add either (3,a) or (3,b), but it violates inj. prop.

① functional property ⇒ partial function.

② $dom(f) = S$

③ injective property.

	func. prop.	total	inj. prop.
①	✓	✓	✓
②	✓	✗	✓
③	✓	✓	✗

Surjective Functions

total $(f) \iff dom(f) = S$

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

* func. prop.
surjective prop.

** - func. prop
- total
- surjective

If f is a * **partial surjection**, we write: $f \in S \twoheadrightarrow T$

- e.g., $\{\{(1,\mathbf{b}),(2,\mathbf{a})\},\{(1,\mathbf{b}),(2,\mathbf{a}),(3,\mathbf{b})\}\} \subseteq \{1,2,3\} \twoheadrightarrow \{a,b\}$
- e.g., $\{(2,\mathbf{a}),(1,\mathbf{a}),(3,\mathbf{a})\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$ partial fun, not surj.
- e.g., $\{(2,\mathbf{b}),(1,\mathbf{b})\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$ partial fun, not surj.

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

- e.g., $\{\{(2,a),(1,b),(3,a)\},\{(2,b),(1,a),(3,b)\}\} \subseteq \{1,2,3\} \twoheadrightarrow \{a,b\}$
- e.g., $\{(2,a),(3,b)\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$
- e.g., $\{(2,a),(3,a),(1,a)\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$

	fun. prop.	total	surj.
①	✓	✓	✓
②	✓	✓	✓
③	✓	✗	✓
④	✓	✓	✗

Bijective Functions

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

- e.g., $\{1,2,3\} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \{a,b\} = \varnothing$ $\{(1,a),(2,b),(3,?)\}$
- e.g., $\{\{(1,a),(2,b),(3,c)\},\{(2,a),(3,b),(1,c)\}\} \subseteq \{1,2,3\} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., ① $\{(2,b),(3,c),(4,a)\} \notin \{1,2,3,4\} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., ② $\{(1,a),(2,b),(3,c),(4,a)\} \notin \{1,2,3,4\} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., ③ $\{(1,a),(2,c)\} \notin \{1,2\} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \{a,b,c\}$

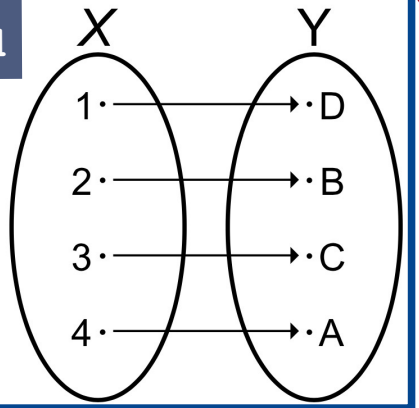
	fun + total	inj	sur.
①	✗	✓	✓
②	✓	✗	✓
③	✓	✓	✗

Exercise

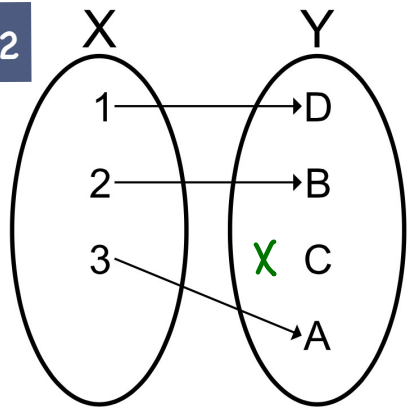
$Y = \{A, B, C, D\}$

$X = \{1, 2, 3, 4\}$

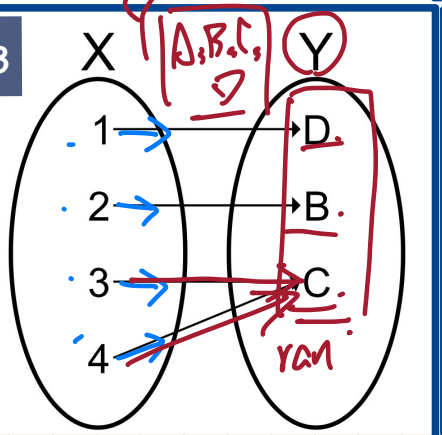
1



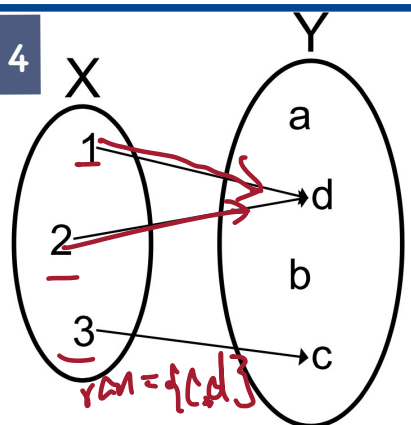
2



3



4



	1	2	3	4
partial	✓	✓	✓	✓
total	✓	✗	✓	✗
injection	✓	✓	✗	✗
surjection	✓	✗	✗	✗
bijection	✓	✗	✗	✗

Lecture 11 - Oct. 10

Bridge Controller Intro

Formalizing Arrays as Functions
Bridge Controller: Intro
Initial Model: Abstract State

Announcements/Reminders

1. Math Review Lecture
2. Lab1 & Lab2
3. No Lab

WrittenTest1 on Wednesday, October 23

↳ guide & practice Q's Monday latest.

Office Hours during **Reading Week** TBA:

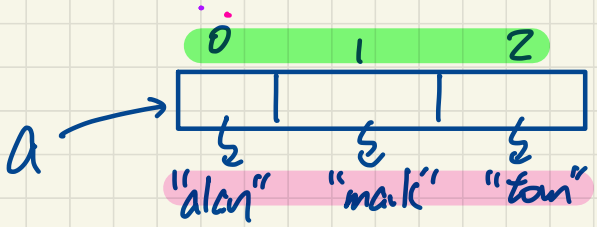
- Help on **Lab3**
- Questions on, e.g., **Lab1**, **Lab2**, **Math Review** Lecture

Bonus Opportunity: Midterm Course Survey (eClass)

Formalizing Arrays as Functions

```
String[] names = {"alan", "mark", "tom"};
```

a



indices

contents

(1) $a \in \mathbb{Z} \nrightarrow String$ (partial)

(2) ✗ $a \in \mathbb{Z} \rightarrow String$ (total)

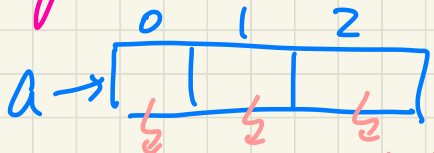
∵ (1) $dom(a) = \mathbb{Z} = \{\ldots, -8, -7, \ldots, 0, \ldots\}$ not appropriate indices

(2b) $a \in \mathbb{N} \rightarrow String$

∵ (2) $\mathbb{N}$ and $\mathbb{Z}$ are infinite.

$a \in \mathbb{Z} \nrightarrow String$

same ran val not mapped by distinct dom. values

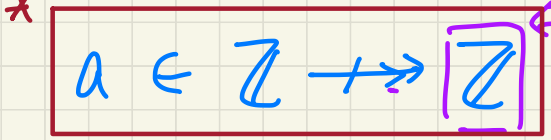


$a = \{(0, \text{"Alan"}), (1, \text{"Mark"}), (2, \text{"Alan"})\}$

$a \in \mathbb{Z} \twoheadrightarrow String$

$ran(a) = String$

all possible strings.



to be revised and discussed after. R.W.

(3, 5)

assume: int type in Java

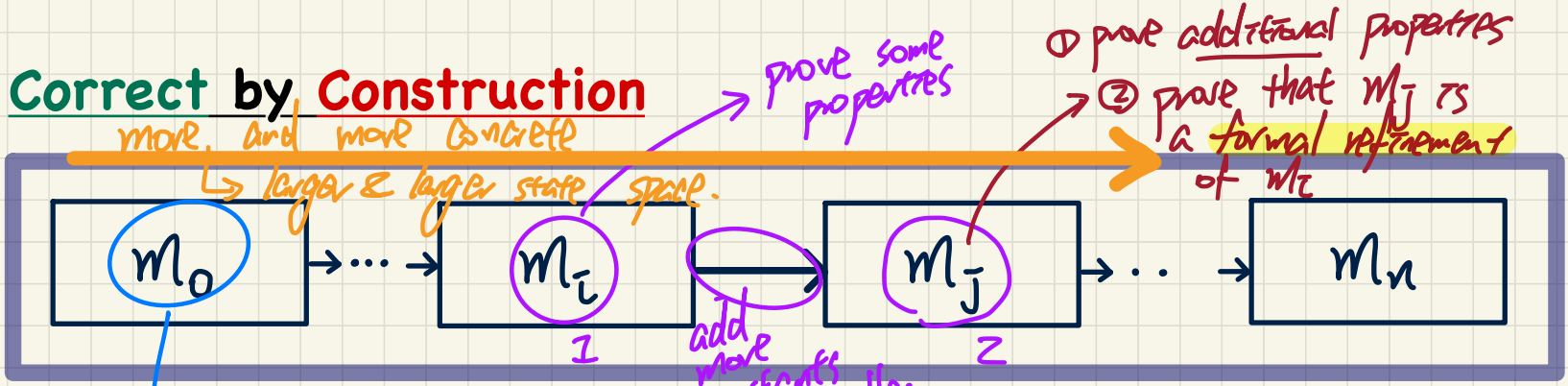
2^{32} possible values.

possible: every stores all int values.

But if no such assumption → ℤ is infinite

→ a ∈ ℤ ⇸ ℤ is not appropriate ∵ ℤ is infinite.

Correct by Construction



initial, most abstract model

M0 → M1 → M2

M1 is refined by M0

1. do not include all variables & constants here
   (only a few at a time in each model)
2. in each model, the # of POs to prove is manageable.

State Space of a Model

typing constraints alone does not restrict the variables/constants to legitimate values.

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

$StateSpace = \{(C, L, accounts) \mid C \in \mathbb{N}_1 \wedge L \in \mathbb{N}_1 \wedge accs \in S \to \mathbb{Z}\}$

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

* $C \in \mathbb{N}_1 \wedge L \in \mathbb{N}_1 \wedge accounts \in String \to \mathbb{Z}$ /* typing constraint */

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

invariant

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

Init: $(10{,}000, 20{,}000, \emptyset)$

C_1 : $(10{,}000, 20{,}000, \{"bill" \mapsto 5000\})$

what if no invariant:

C_2 : $(10{,}000, 20{,}000, \{"bill" \mapsto -50{,}000\})$

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

bottleneck for model checking (4315).

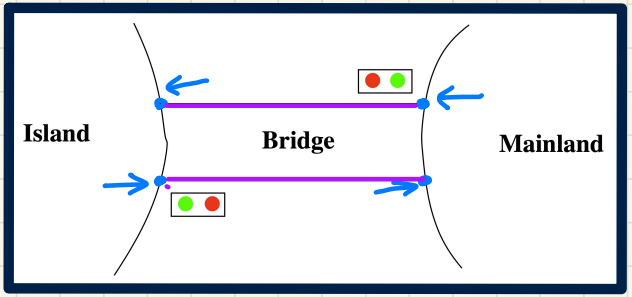
$StateSpace = \{ var/const\ list \mid typing\ constraints \wedge Invariant \}$

without INV, the resulting set is much bigger

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.



* Important assumption to prove safety properties

the only REQ to focus on in the critical section

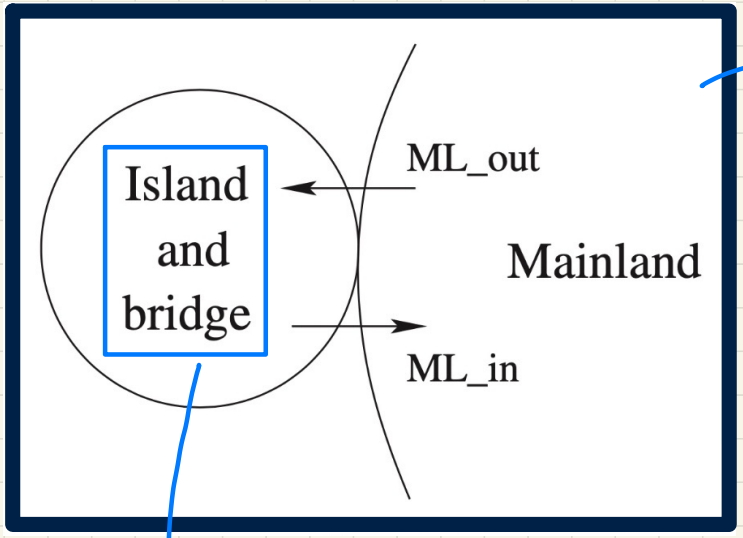
most abstract model

some refinement to introduce to the model

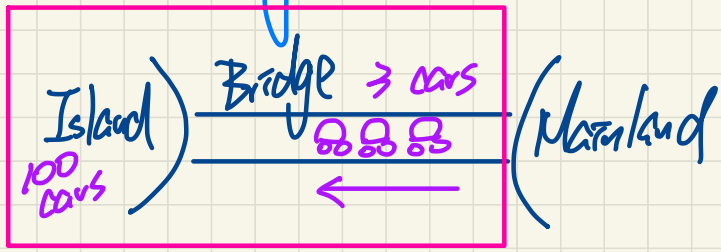
Bridge Controller: Abstraction in the Initial Model

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

in the wlo, consider them not separable!



no bridge!



$n = 100 + 3 = 103$

n: number of cars in Island & the bridge

Island-bridge compound.

Bridge Controller: State Space of the Initial Model

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

Static Part of Model

constants: d

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

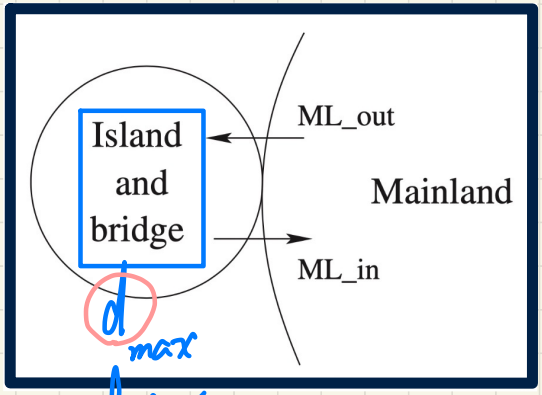
$d \geq 0$

Dynamic Part of Model

variables: n

invariants: ≥ 0
inv0_1 : $n \in \mathbb{N}$
inv0_2 : $n \leq d$

# of cars currently on the island-bridge compound



# of cars in island & bridge

n : current # of cars

$d \leq n$

Bridge Controller: State Transitions of the Initial Model

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

$n = \emptyset \not\in 0$

state space

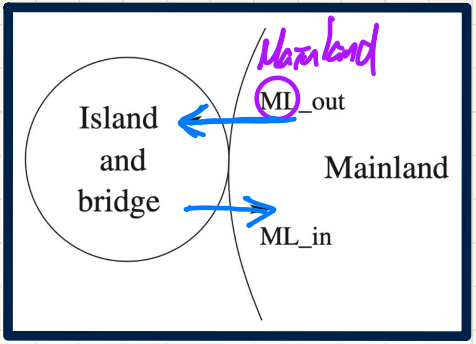
constants: d

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

variables: n

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

must hold true on every state

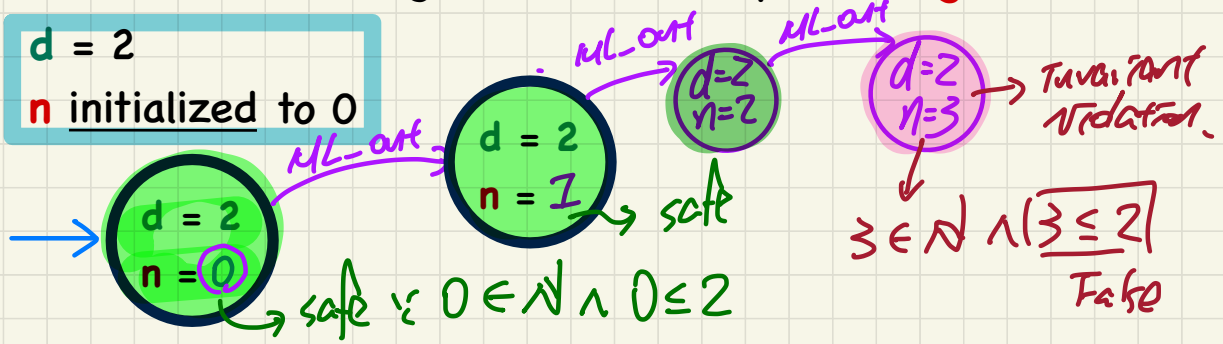



```
ML_out
  begin
    n := n + 1
  end
```

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

State Transition Diagram on an Example Configuration

d = 2
n initialized to 0



Review Q & A - Oct. 20

Written Test 1

Practice Test Questions
Math Review Lecture

Written Test 1: Practice Question 1

Consider the following predicate:

! x, y . x : NAT & y: NAT => x * y > 0

$\forall x, y \cdot x \in \mathbb{N} \wedge y \in \mathbb{N} \Rightarrow x * y > 0$

T or F.

Consider each of the following statements **in isolation**, choose **all** that are correct.

- [x] 1. The predicate is not a theorem and can be disproved by an x-y pair (5, 0).
- [ ] 2. The predicate is not a theorem and can be disproved by an x-y pair (12, 13).
- [ ] 3. The predicate is a theorem and can be proved by an x-y pair (2, 3).
- [ ] 4. The predicate is not a theorem and can be disproved by an x-y pair (12, -2).
- [ ] 5. None of the listed statements is correct.
- [ ] 6. The predicate is a theorem and can be proved by an x-y pair (5, 4).

witness: $5 \in \mathbb{N} \wedge 0 \in \mathbb{N} \Rightarrow 5 * 0 > 0$

$12 \in \mathbb{N} \wedge 13 \in \mathbb{N} \Rightarrow 12 * 13 > 0$

$12 \in \mathbb{N} \wedge -2 \in \mathbb{N} \Rightarrow 12 * -2 > 0$

Written Test 1: Practice Question 2

Consider the following predicate:

#x, y . x : NAT & y: NAT & x * y > 0

$\exists x, y \cdot x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge x * y > 0$

① to prove ∃, need one witness

② to disprove, need to consider all combinations.

Consider each of the following statements **in isolation**, choose **all** that are correct.

- [ ] 1. The predicate is not a theorem and can be disproved by an x-y pair (5, 0).
- [ ] 2. The predicate is not a theorem and can be disproved by an x-y pair (12, -2).
- [ ] 3. None of the listed statements is correct.
- [ ] 4. The predicate is a theorem and can be proved by an x-y pair (-2, -3).
- [x] 5. The predicate is a theorem and can be proved by an x-y pair (5, 4).
- [x] 6. The predicate is a theorem and can be proved by an x-y pair (2, 3).
- [ ] 7. The predicate is not a theorem and can be disproved by an x-y pair (12, 13).

$-2 \in \mathbb{N}$ (F) $\wedge$ $-3 \in \mathbb{N}$ (F) $\wedge$ $-2 * -3 > 0$ (T)

$5 \in \mathbb{N}$ (T) $\wedge$ $4 \in \mathbb{N}$ (T) $\wedge$ $5 * 4 > 0$ (T)

(F) $5 \in \mathbb{N}$ (T) $\wedge$ $0 \in \mathbb{N}$ (T) $\wedge$ $5 * 0 > 0$ (F)

this witness evaluates to false, but not sufficient to disprove ∃.

Written Test 1: Practice Question 3

precedence.

Given two sets S and T, say we write:

- S \/ T for their union
- S /\ T for their intersection
- S \ T for their difference

What is the **cardinality** of the power set of ({a, b, c, d} \ {a, e}) \/ {a, f}? Enter an integer value (with no spaces).

Answer: 32

$\{a,b,c,d\} \setminus \{a,e\} \cup \{a,f\}$

$|\mathbb{P}((\{a,b,c,d\} \setminus \{a,e\}) \cup \{a,f\})|$

$\{b,c,d\} \cup \{a,f\}$

$|\mathbb{P}(\{a,b,c,d,f\})|$

$2^{|\{a,b,c,d,f\}|} = 2^5 = 32$

Written Test 1: Practice Question 4

precedence

Consider the following logical quantification:

!x,y.x:NAT&y:NAT=>x+y>=10&x+y<20

Convert the above predicate to an equivalent one using the other logical quantifier.

Note the following constraints on your answer:

- Only put pairs of parentheses **when necessary**.
- Like the above predicate, there should be **no** white spaces.
- Like the above predicate, numerical constants (i.e., 10, 20) must appear as the right operands of the relational expressions (e.g., x + y >= 10).
- Relational expressions should be simplified whenever possible, e.g., write x >= 20 rather than not(x < 20).

Be cautious about the spellings: this question will be graded **automatically** and no partial marks will be give to spelling mistakes.

Answer:

Written Test 1: Practice Question 5

Consider two sets:

- S = {x, y}
- T = {1, 2, 3}

$S \times T$

$r \in S \leftrightarrow T$ the set of all possible relations between S and T.

Write out the **maximum** relation r such that r : S <-> T.

Requirements. In your answer:

- Pairs must be **sorted** in an **ascending** order by the first elements, or by the second elements if the first elements are identical. For examples: (x, 2) appears before (y, 1), (x, 1) appears before (x, 2), etc.
- No white spaces should be included, e.g., write (x,1) rather than (x, 1).

Be cautious about the spellings: this question will be graded **automatically** and so no partial marks will be given due to spelling mistakes.

Answer:

$\{(x,1),(x,2),(x,3),(y,1),(y,2),(y,3)\}$

Written Test 1: Practice Question 6

Consider two sets:

- S = {x, y}
- T = {1, 2, 3}

Enumerate the following set:

{(a,b) | a : S & b : T & a /= x & b < 3}

Requirements. In your answer:

- Pairs must be **sorted** in an **ascending** order by the first elements, or by the second elements if the first elements are identical. For examples: (x, 2) appears before (y, 1), (x, 1) appears before (x, 2), etc.
- No white spaces should be included, e.g., write (x,1) rather than (x, 1).

Be cautious about the spellings: this question will be graded **automatically** and so no partial marks will be given due to spelling mistakes.

Answer: {(y,1),(y,2)}

property: ① $a \neq x \vee b < 3$ ② $a \neq x \Rightarrow b < 3$ ③ $\neg(a \neq x \vee b < 3)$ (Exercises)

Written Test 1: Practice Question 7

$\binom{n}{m}$ Given n elements, how many ways can we make subsets of size m?

Consider two sets:

- S = {x, y}
- T = {1, 2, 3}

<| dom. res.
<<| dom. sub.

Consider r such that r : S <-> T:

{(x, 1), (x, 3), (y, 1), (y, 2)}

$r = \{(x,1), (x,3), (y,1), (y,2)\}$

What is the result of the following expression:

{x} <<| r |> (T \ {2})

ran. res.

dom res sub.

Requirements. In your answer:

- Pairs must be **sorted** in an **ascending** order by the first elements, or by the second elements if the first elements are identical. For examples: (x, 2) appears before (y, 1), (x, 1) appears before (x, 2), etc.
- No white spaces should be included, e.g., write (x,1) rather than (x, 1).

Be cautious about the spellings: this question will be graded **automatically** and so no partial marks will be given due to spelling mistakes.

ex. $\{x\} \lhd (r \rhd (T \setminus \{2\}))$

Answer:

$\{x\} \lhd (r \rhd (T \setminus \{2\}))$

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

$\{(y,1)\}$

$\{(x,1), (x,3), (y,1)\}$

e.g. $|\{s \mid s \in \mathcal{P}(\{1,2,3,4,5,6\}) \wedge (|s|=3 \vee |s|=5)\}|$

$=$ Compute: $\mathcal{P}(\{1,2,3,4,5,6\})$

$\binom{6}{3}+\binom{6}{5}$

$=20+6$

$=26$.

- $\emptyset$ subset of size 0
- $\binom{6}{3} = \frac{6*5*4}{3!} = 20$ subsets of size 3
- $\{1,2,3,4,5,6\}$ subset of size 6

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

Lecture 12 - Oct. 22

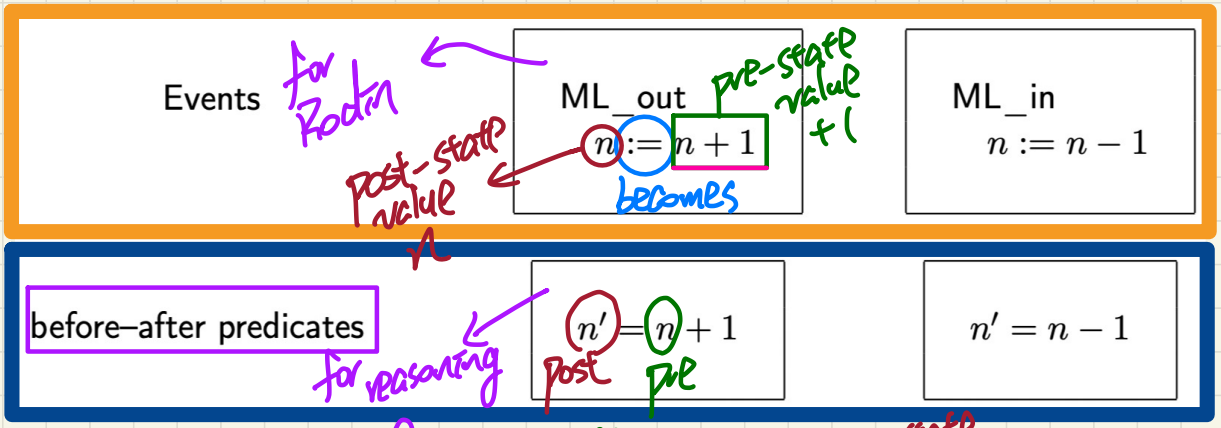
Bridge Controller

Event Action vs. Before-After Predicate
Before- vs. After-States
Sequents: Syntax and Semantics

Announcements/Reminders

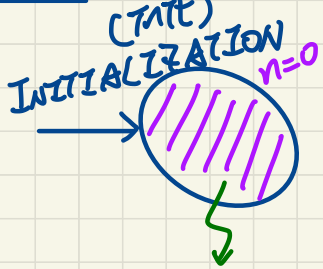
- ProgTest1 results to be released around next Monday.
- WrittenTest1 tomorrow during your enrolled lab session
- Lab4 released (ProgTest2 on November 6)
  + Try to complete Part 1 by Friday.
  + Follow the proof steps in Part 2 & collect questions.
  + Scheduled lab session on October 30.

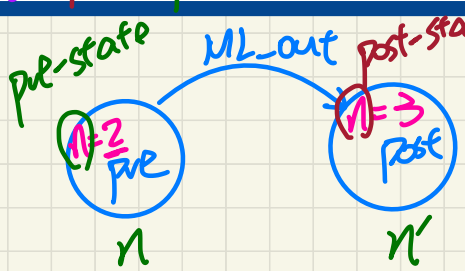
Before-After Predicates of Event Actions



- Pre-State
- Post-State
- Sate Transition

ASM

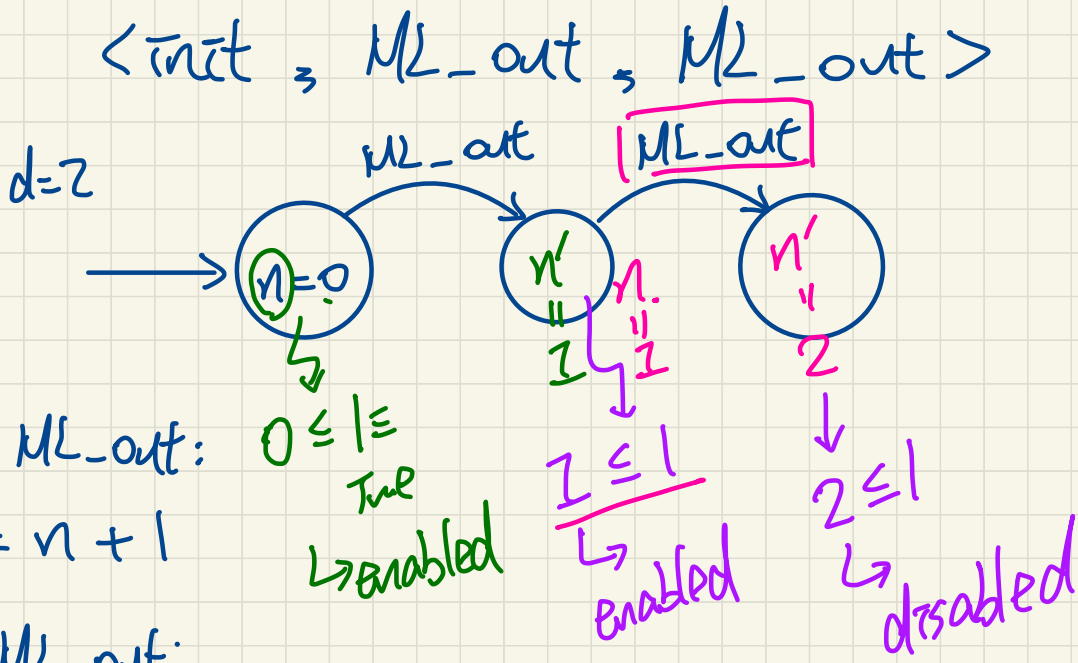




For each variable x:

(1) Write x to denote its pre-state value.

(2) Write x' to denote its post-state value.



action for ML_out:

$n := n + 1$

guard for ML_out:

~~true~~ say: $n \leq 1$

~~(always enabled)~~

Transition of an Event

$b \in Account \pfun \mathbb{N}$

withdraw

$a : Account$

$v : \mathbb{N}$

where

$a \in dom(b)$ 2. I must be true

begin

* effect of event action.

$b := b \domsub \{a \mapsto b(a) - v\}$ *

end

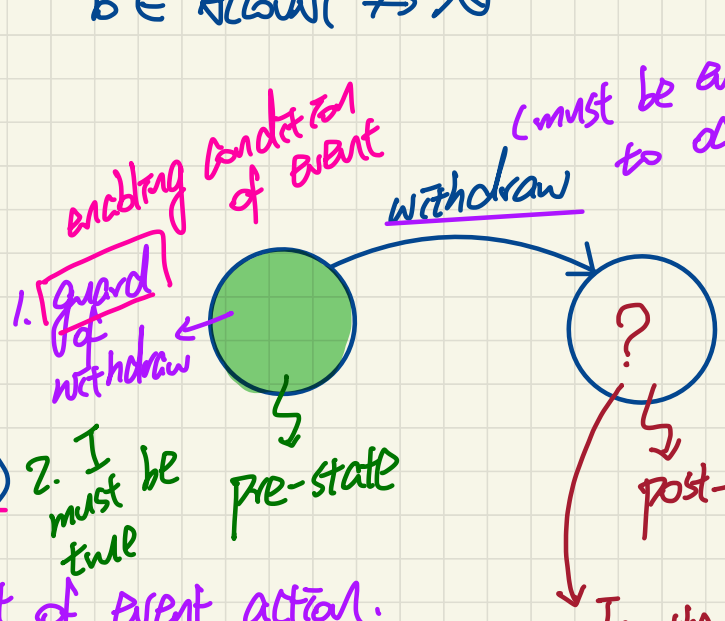
Pre-State:

I

Post-State:

I remains to be true

$\forall a \cdot a \in dom(b) \Rightarrow b(a) \geq -C$



Is the post-state, after the event's action takes effect, still safe?

$I: \forall a \cdot a \in dom(b) \Rightarrow b(a) \geq -C$

Is I maintained?

Invariant I in pre-state:

$$\forall x \cdot x \in dom(b) \Rightarrow b(x) \geqslant -C$$

event action:

$$b := b \lhd \{a \mapsto b(a) - v\}$$

I in post-state:

$$\forall x \cdot x \in dom(b \lhd \{a \mapsto b(a) - v\}) \Rightarrow (b \lhd \{a \mapsto b(a) - v\})(x) \geqslant -C$$

$b \lhd \{a \mapsto b(a) - v\}$: new value of b in post-state of withdraw.

Exercise: Event Actions vs. Before-After Predicates

Q. Are the following event actions suitable for a swap between x and y?

```
swap
  begin
    temp := x
    x := y
    y := temp
  end
```

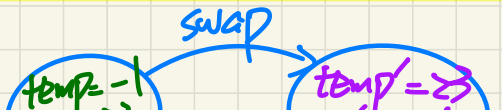
BAP:

$temp' = x$

$x' = y$

$y' = temp$

pre-state value, which has no connection to



Correct:

$x := y$

$y := x$

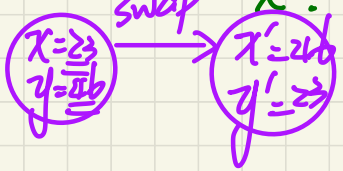
BAP:

$x' = y$

$y' = x$

inappropriate

:= should not be considered as seq. assignment



Design of Events: Invariant Preservation

to be formulated as a proof obligation

variables: n

ML_out
begin
$n := n + 1$
end

ML_in
begin
$n := n - 1$
end

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

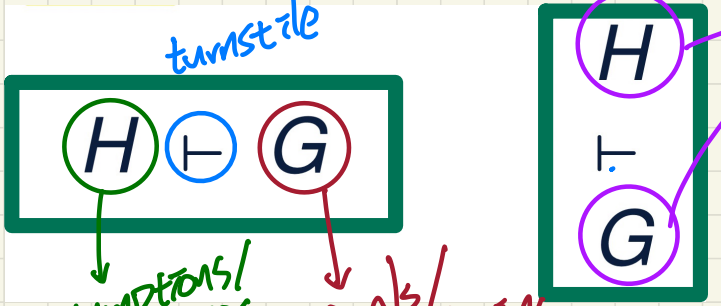
$n \in \mathbb{N}$

Desire: $\forall state \cdot state \in StateSpace \Rightarrow I(state)$

To disprove this: find $state \in StateSpace$ ($n \in d$) but $\neg I(state)$.

Sequents: Syntax and Semantics

Syntax



each is a set of predicates

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

Semantics

$H \vdash G \iff H \Rightarrow G$

predicate

↳ assuming that H is true, G is provable.

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

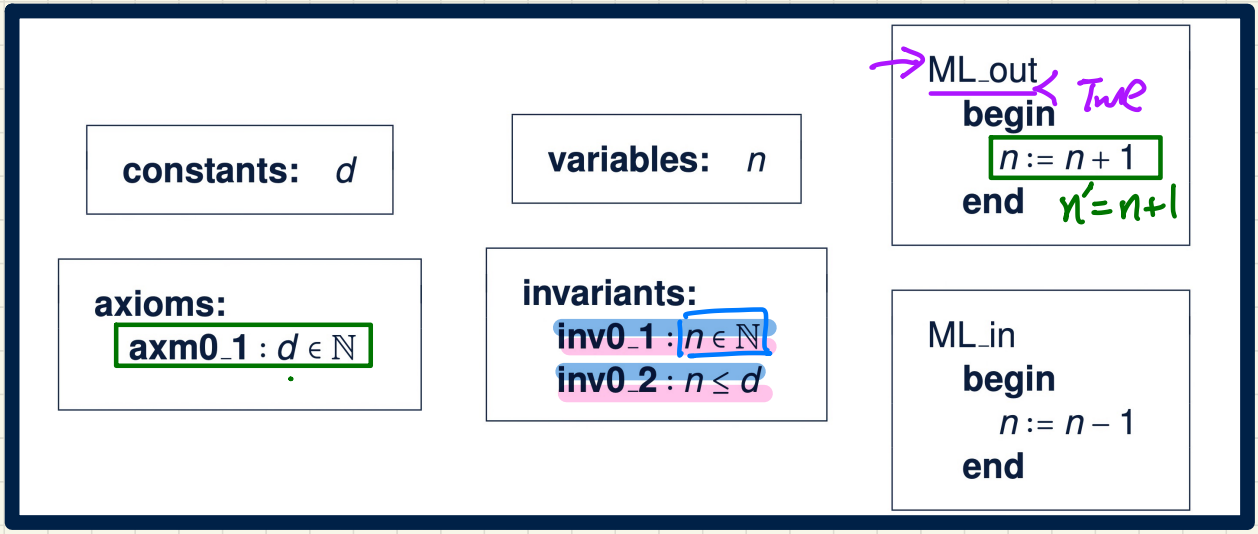
$\vdash G$

① $\boxed{True \vdash G} \equiv True \Rightarrow G \equiv \boxed{G}$

② $\boxed{False \vdash G} \equiv False \Rightarrow G \equiv \boxed{True}$ ↝ nothing to prove!

PO/VC Rule of Invariant Preservation

Exercise ML_in/INV



Axioms

Invariants Satisfied at *Pre-State*

Guards of the Event

⊢

Invariants Satisfied at *Post-State*

INV

ML_out/INV

$d \in \mathbb{N}$

$n \in \mathbb{N}$ (pre-state), $n \le d$, $true$

$\vdash$

$n' \in \mathbb{N} \wedge n' \le d$ (post-state)

ncmp of Proof obligation

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

$\wedge$

$n+1 \le d$

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \le d$

$true$

$\vdash$

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

$n+1 \le d$

Lecture 13 - Oct. 24

Bridge Controller

Proof Obligation of Inv. Preservation
Inference Rule: Syntax and Semantics

Announcements/Reminders

- ProgTest1 & WT1 results to be released next Monday.
- Lab4 released (ProgTest2 on November 6)
  + Try to complete Part 1 ASAP.
  + Follow the proof steps in Part 2 & collect questions.
  + Scheduled lab session on October 30.

↓
- Demo of Lab4 Part2
- Q&A -

PO/VC Rule of Invariant Preservation

verification condition

proof obligation

ML_out

pre I → post I

constants: d

variables: n

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

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

model no

numbers of INV

ML_out when True
begin
$n := n + 1$
end

BAP:

ML_in
begin
$n := n - 1$
end

(A) $n' \in \mathbb{N} \wedge n' \leq d$
↳ $n+1 \in \mathbb{N} \wedge n+1 \leq d$
Inv at post-state

(B) $n' = n+1$

Axioms
Invariants Satisfied at *Pre-State*
Guards of the Event — ML_out
⊢
Invariants Satisfied at *Post-State*

INV

main-formed

P.O. for INV. preservation

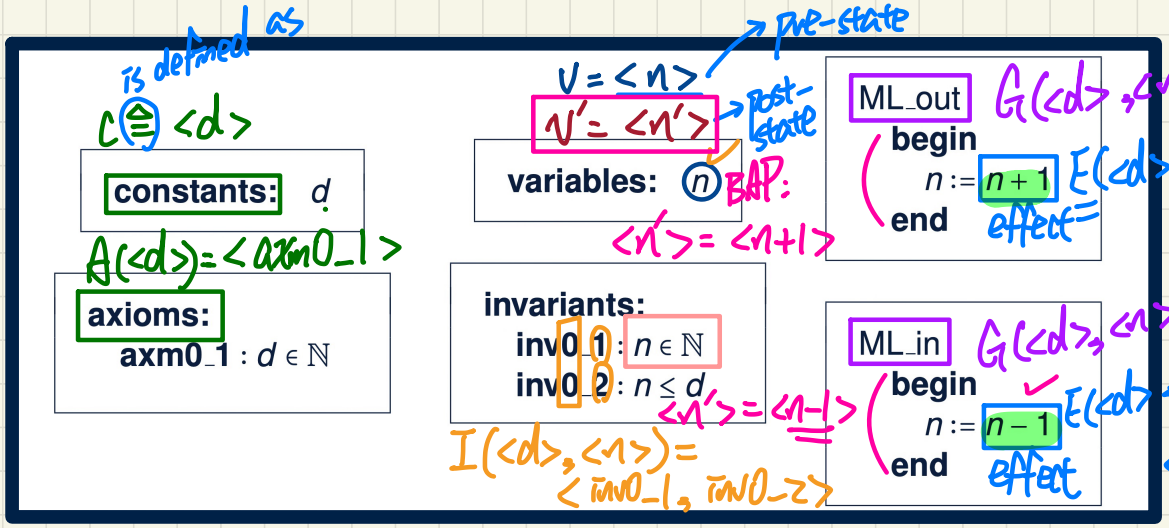
$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
True

≡ $d \in \mathbb{N}$ ∧ $n \in \mathbb{N}$ ∧ $n \leq d$ ∧ True

(new True in hypothesis section is implicitly a conjunction)

⊢ $n' \in \mathbb{N} \wedge n' \leq d$ *
$n+1 \in \mathbb{N} \wedge n+1 \leq d$

PO/VC Rule of Invariant Preservation: Components



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

choice by predicate

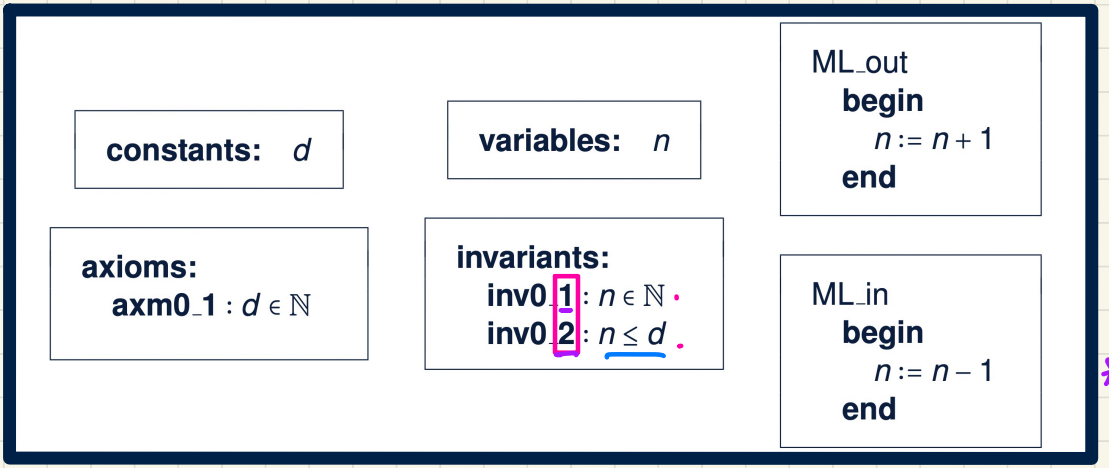
Swap.

$x, y :| x' = y \wedge y' = x$

change list of variables

post-state characterization (as a predicate)

PO/VC Rule of Invariant Preservation: Sequents



$A(c)$

$I(c, v)$ ← pre-state values

* $G(c, v)$

$\vdash$ effect of event

** $I_i(c, E(c, v))$ ← i=1, i=2

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

* Guards of some events : ML_out or ML_in (2)

** Some all invariant conditions : inv0_1 or inv0_2 (2)

Total # of POs : 2 * 2 = 4.

PO1 : ML_out / inv0_1 / INV
(evt / inv. cond / nature of PO.)

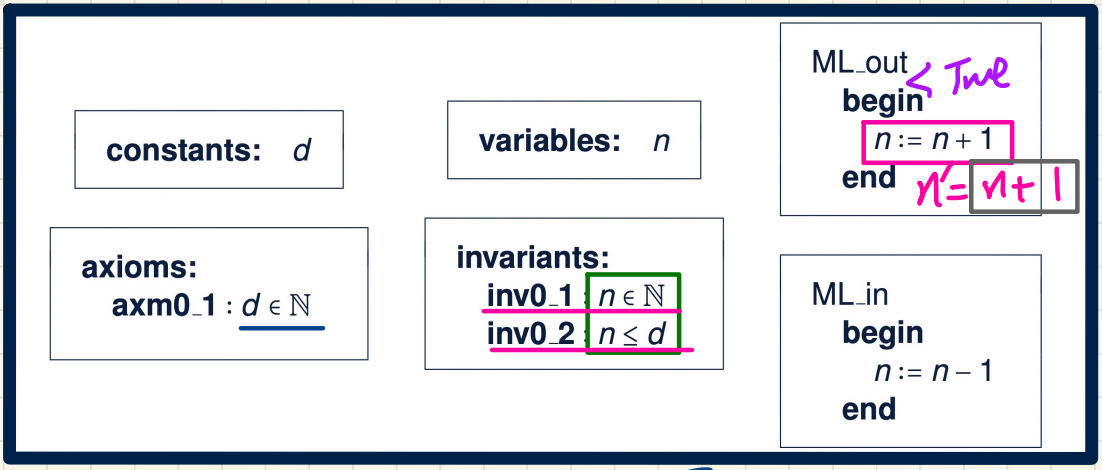
PO2 : ML_out / inv0_2 / INV

EXERCISE : PO3 & PO4 ?

M2_out / inv0_1 / INV

P.O. is related to whether or not taking a state transition of event M2_out can preserve/maintain inv0_1

PO/VC Rule of Invariant Preservation: Sequents



$A(c)$

$I(c, v)$

$G(c, v)$

$\vdash$

$I_i(c, E(c, v))$

PO1 ML_out/INV0_1/INV

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

$\top$

$\vdash$

$n' \in \mathbb{N}$

$n+1$

PO2 ML_out/INV0_2/INV

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

$\top$

$\vdash$

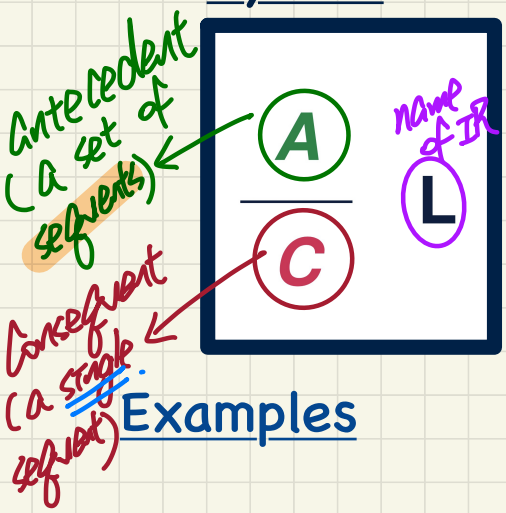
$n' \leq d$

$n+1$

Exercise : PO3 PO4

Inference Rule: Syntax and Semantics

Syntax



Semantics

$A \Rightarrow C \equiv \text{True}.$

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

Examples

$$\frac{H_1 \vdash G}{H_1, H_2 \vdash G}$$

MON = monotonicity.

To prove $H_1, H_2 \vdash G$

it's sufficient to prove

$H_1 \vdash G$

Lecture 14 - Oct. 29

Bridge Controller

Inference Rules: Proof Steps
Interpreting Unprovable Sequents

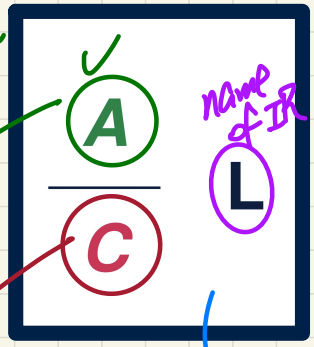
Announcements/Reminders

- ProgTest1 & WT1 results released
- Lab4 released (ProgTest2 on November 6)
  + Scheduled lab session on **October 30**.

Inference Rule: Syntax and Semantics

$\vdash G \equiv True \vdash G$

Syntax



antecedent (a set of sequents)

consequent (a single sequent)

name of IR

Semantics

$A \Rightarrow C \equiv True.$

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

$\frac{}{C}\ L$ means $\frac{True}{C}$ (meaning that C is an axiom)

To prove the consequent C, it's sufficient to prove the antecedent A (instead). rewriting.

Examples

$$\frac{H_1 \vdash G}{H_1, H_2 \vdash G}\ Mon$$

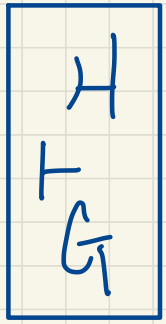
Mon = monotonicity.

you can drop hypotheses

To prove $H_1, H_2 \vdash G$

it's sufficient to prove $H_1 \vdash G$

Sequent



$H \Rightarrow G$

Inference Rule

$$\frac{A}{C}\,L \qquad A \Rightarrow C \equiv True$$

to make progress in proofs, rewrite C into A

Proof of Sequent: Steps and Structure

Outstanding Sequent to Prove

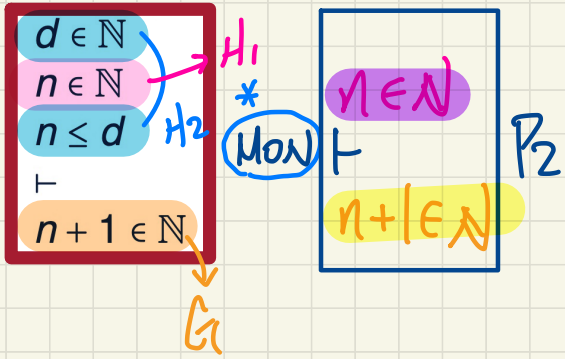
$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$\vdash$
$n + 1 \in \mathbb{N}$

ML_out/inv0_1/INV

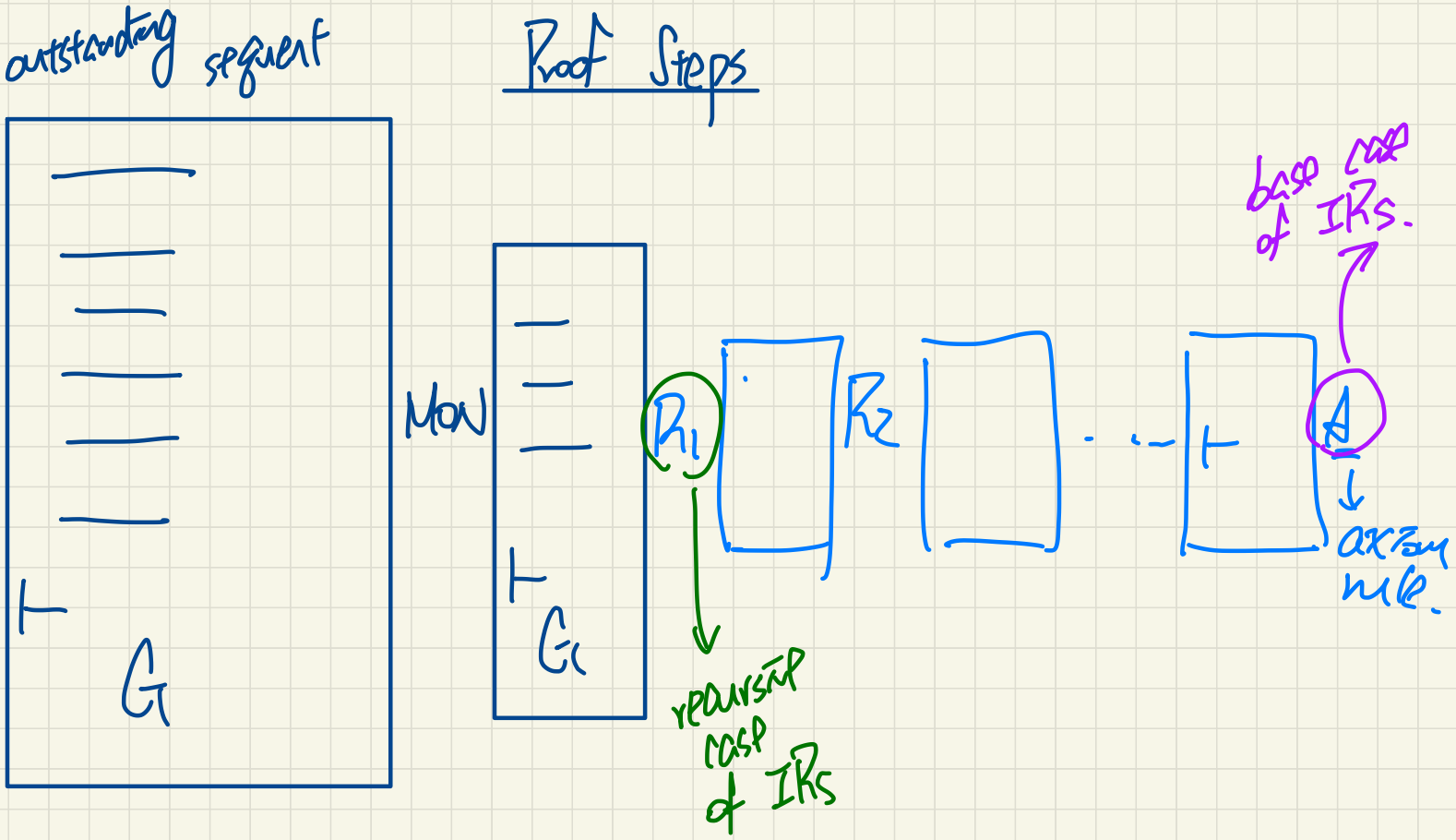
Known Inference Rules

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

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



* Analyze the goal predicate to prove: what are the relevant variables? what hypotheses are useful?


outstanding sequent
Proof Steps
⊢
G
MON
⊢
G'
R1
R2
⊢
A
base case of IRs.
axiom rule.
recursive case of IRs

Understanding Inference Rule: OR_L

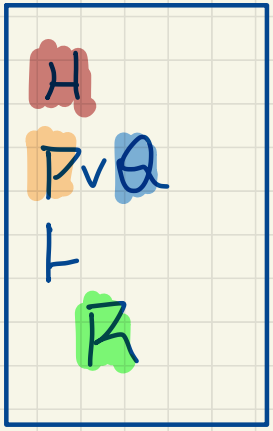
$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, P \vee Q \vdash R}\ \text{OR\_L}$$

disjunction

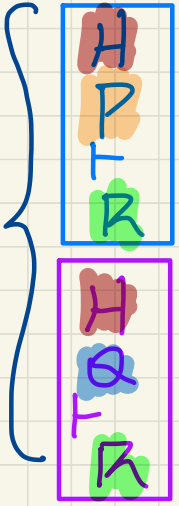
Q1. Does OR_L help us:

(A) split one sequent into two

(B) combine two sequents into one



OR_L



the rule concerns what's to the L of ⊢

Example Inference Rules

Peano Numbers Theorems

Axioms rules (base cases)

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

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

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

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

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

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

$n < m$

$\vee$

$n = m$

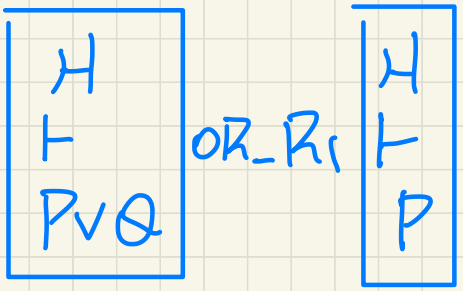
$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, P \vee Q \vdash R}\ \text{OR\_L}$$

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

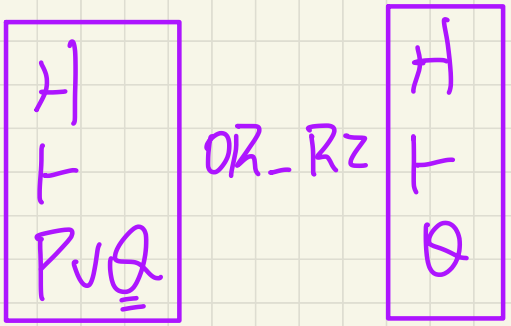
$$\frac{H \vdash Q}{H \vdash P \vee Q}\ \text{OR\_R2}$$

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

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



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



Discharging POs of original m0: Invariant Preservation

ML_out/inv0_1/INV

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

Exercise!

ML_in/inv0_1/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$\vdash$ True guard of ML_m1
$n-1 \in \mathbb{N}$

Mon

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

new guard for ML_m1: $n > 0$

unprovable

One possible resolution: add some guard condition of ML_m1, so that the extra hypothesis can make this seq. provable.

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

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

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

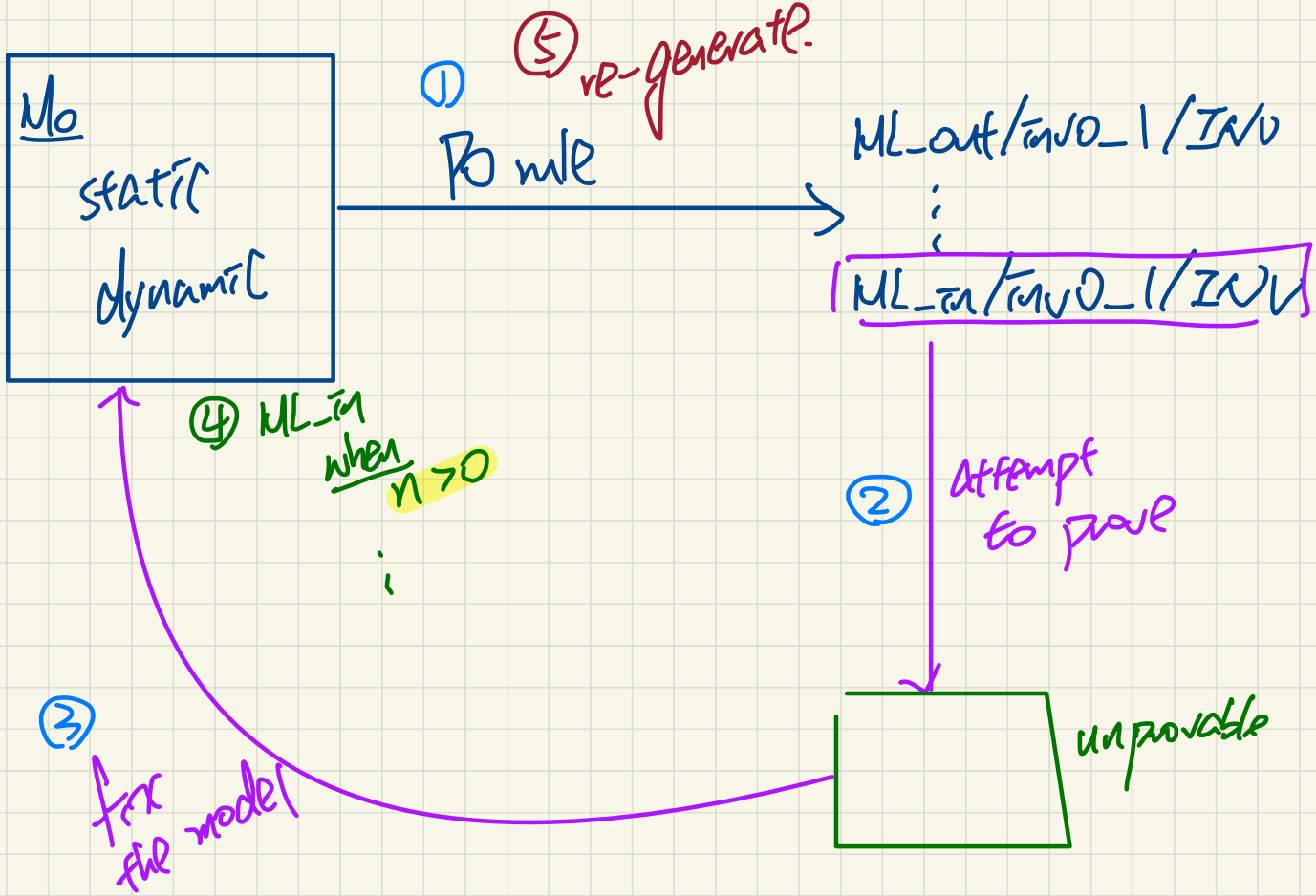
$$\frac{}{n \in \mathbb{N} \vdash n+1 \in \mathbb{N}}\ \text{P2}$$

ML_out/inv0_2/INV

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

ML_in/inv0_2/INV

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


M0
static
dynamic
① Rule
⑤ re-generate
ML_0/inv0_1/INV
ML_n/inv0_1/INV
② attempt to prove
unprovable
③ for the model
④ ML_n when n>0

Lecture 15 - Oct. 31

Bridge Controller

Revising M0: Adding Event Guards
Re-Generating/Re-Proving PO Sequents

Announcements/Reminders

- Lab4 due tomorrow at **noon**
- ProgTest2 next Wednesday, November 6

* $n \le d \vdash n-1 < d$ DEC

Discharging POs of original m0: Invariant Preservation

ML_out/inv0_1/INV

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

✗ P2 ∵ too many irrelevant hypotheses.

MON → $n \in \mathbb{N} \vdash n+1 \in \mathbb{N}$ → P2

ML_in/inv0_1/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \le d$
$\vdash$
$n-1 \in \mathbb{N}$ ← True guard of ML_tr

MON → $n \in \mathbb{N} \vdash n-1 \in \mathbb{N}$?

unprovable

new guard for ML_tr: $n > 0$

One possible resolution: add some guard condition of ML_tr, so that the extra hypotheses can make this seq. provable.

ML_out/inv0_2/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \le d$ ← True
$\vdash$
$n+1 \le d$

MON → $n \le d \vdash n < d \;\vee\; n+1 \le d$

new guard for ML_out: not provable ∵ n might be equal to d.

ML_in/inv0_2/INV

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

DEC ✗ → ARI $p \le q \equiv p < q \vee p = q$ → $d \in \mathbb{N}, n \in \mathbb{N}, n \le d \vdash n-1 < d \vee n-1 = d$

OR_R1 → $d \in \mathbb{N}, n \in \mathbb{N}, n \le d \vdash n-1 < d$ MON*

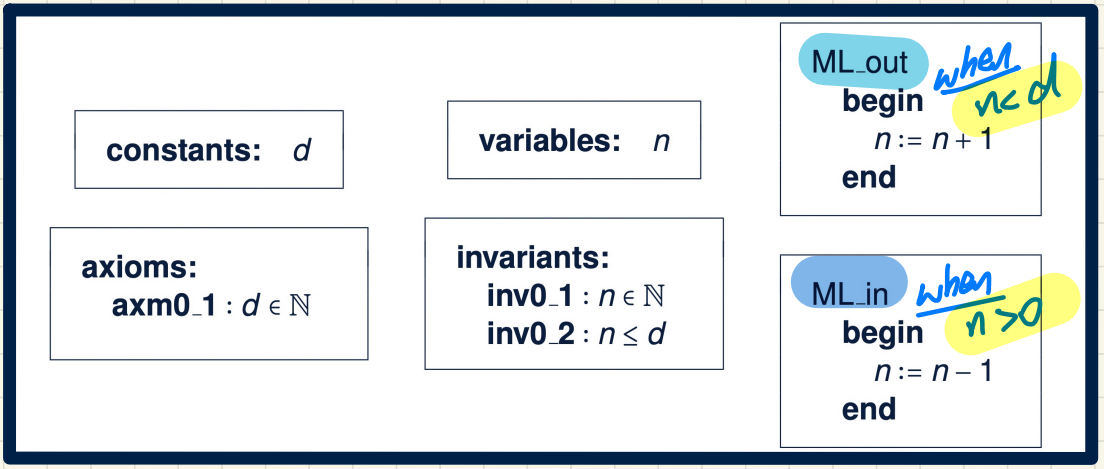
OR_R1: $\dfrac{H \vdash P}{H \vdash P \vee Q}$

MON: $\dfrac{H1 \vdash G}{H1, H2 \vdash G}$

DEC: $\dfrac{}{n \le m \vdash n-1 < m}$

P2: $\dfrac{}{n \in \mathbb{N} \vdash n+1 \in \mathbb{N}}$

PO/VC Rule of Invariant Preservation: Revised M0



$$A(c)$$
$$I(c, v)$$
$$G(c, v)$$
$$\vdash$$
$$I_i(c, E(c, v))$$

Q. How many PO/VC rules for model m0? 4

Discharging POs of revised m0: Invariant Preservation

ML_out/inv0_1/INV — Ex.

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

ML_in/inv0_1/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \le d$
$n > 0$ — new guard for ML_in
$\vdash$
$n - 1 \in \mathbb{N}$

MON:
$n > 0$
$\vdash$
$n - 1 \in \mathbb{N}$
P2

ML_out/inv0_2/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \le d$
$n < d$ — new guard for ML_out
$\vdash$
$n + 1 \le d$

MON:
$n < d$
$\vdash$
$n + 1 \le d$
INC

ML_in/inv0_2/INV — Ex.

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \le d$
$n > 0$
$\vdash$
$n - 1 \le d$

$$\frac{H \vdash P}{H \vdash P \vee Q}\ \text{OR\_R1} \qquad \frac{H1 \vdash G}{H1, H2 \vdash G}\ \text{MON}$$

$$\frac{}{n \le m \vdash n - 1 < m}\ \text{DEC} \qquad \frac{}{n < m \vdash n + 1 \le m}\ \text{INC}$$

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

Discharging PO of DLF: Second Attempt

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

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

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

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

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

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

ARI

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

MON

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

OR_L

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

OR_R1

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

HYP

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

Lecture 16 - Nov. 5

Bridge Controller

Invariant Establishment
Deadlock Freedom

Announcements/Reminders

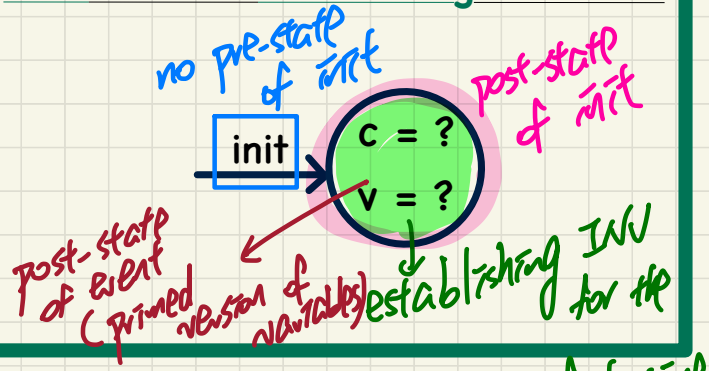
- ProgTest2 tomorrow

Initializing the System

$d \in \mathbb{N}$ $n \in \mathbb{N}$ $n \le d$ $n < d$ $\vdash$ $n+1 \in \mathbb{N}$	$d \in \mathbb{N}$ $n \in \mathbb{N}$ $n \le d$ $n < d$ $\vdash$ $n+1 \le d$	$d \in \mathbb{N}$ $n \in \mathbb{N}$ $n \le d$ $n > 0$ $\vdash$ $n-1 \in \mathbb{N}$	$d \in \mathbb{N}$ $n \in \mathbb{N}$ $n \le d$ $n > 0$ $\vdash$ $n-1 \le d$

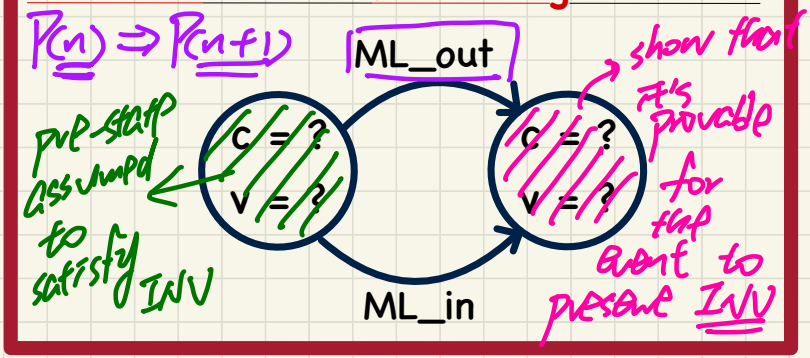
Analogy to Induction:

Base Cases ≈ Establishing Invariants

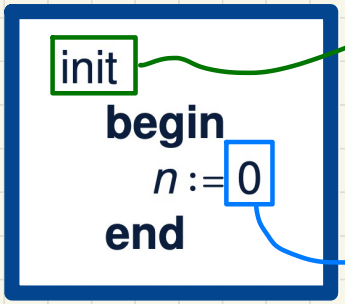


Analogy to Induction:

Inductive Cases ≈ Preserving Invariants



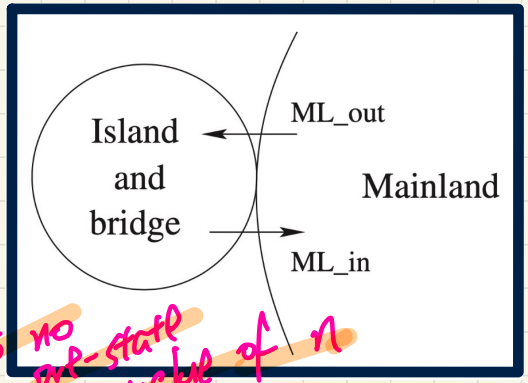
The Initialization Event



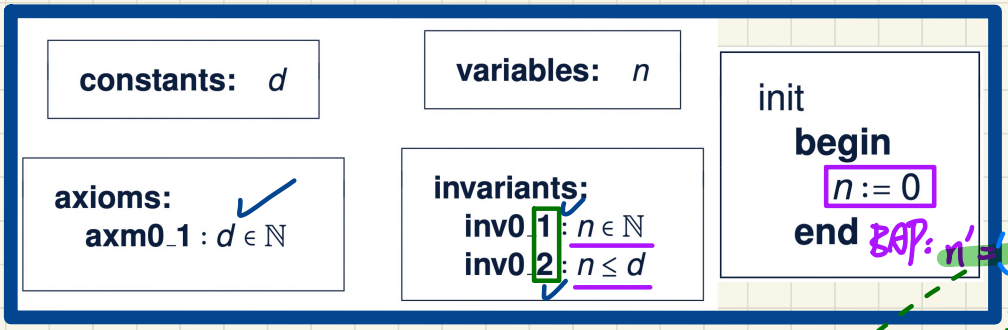
the only event that should be unguarded (initialization is always possible)

constants

$n := n + 1$ not valid init action! there's no pre-state value of n



PO of Invariant Establishment



BAP: $n' = 0$

E(c, v) → regular evt can ref. variables

effect of non-init event

Components

K(c): effect of init's actions

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

init evt can only ref. constants (no v).

Rule of Invariant Establishment

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

A(c): axioms

no invariants assumed

I_i: index of INV

Exercise:

Generate Sequents from the INV rule.

invariants not established yet.

init/inv0_1/INV

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

$0 \in \mathbb{N}$

init/inv0_2/INV

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

$0 \leq d$

Discharging PO of Invariant Establishment

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

init/inv0_1/INV

P1 ✗

MON

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

P1

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

init/inv0_2/INV

P3

↗ where n is instantiated by d

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

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

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

Bridge Controller : REACTIVE SYSTEM

↳ there's always at least one event enabled for the system to progress

unacceptable : deadlock — no event enabled to occur

$\neg G(ML\_out) \wedge \neg G(ML\_in)$ [deadlock condition]

$\equiv$

$\neg (G(ML\_out) \vee G(ML\_in))$

$G(ML\_out) \vee G(ML\_in)$ [deadlock freedom cond.]

event enableness in pre-state

PO Rule: Deadlock Freedom

REQ4	Once started, the system should work for ever.

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

$A(c)$ axioms
$I(c, v)$ invariants
$\vdash$
$G_1(c, v) \vee \cdots \vee G_m(c, v)$

DLF ≙ deadlock freedom

At least one of the m events is enabled.

- 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* $\mathrel{\widehat{=}}$ $n > 0$

Exercise: Generate Sequent from the DLF rule.

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
$\vdash \quad \underbrace{n < d}_{G(ML\_out)} \vee \underbrace{n > 0}_{G(ML\_in)}$

PO	(n) pre-state	(n') post-state
inv. est	X	✓
inv. pre.	✓	✓
DLF	✓	X

Example Inference Rules

$\top$ true
$\bot$ false

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

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

false $\Rightarrow P \equiv$ True

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

$P \Rightarrow$ True $\equiv$ True

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

replace every free occurrence of E by F

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

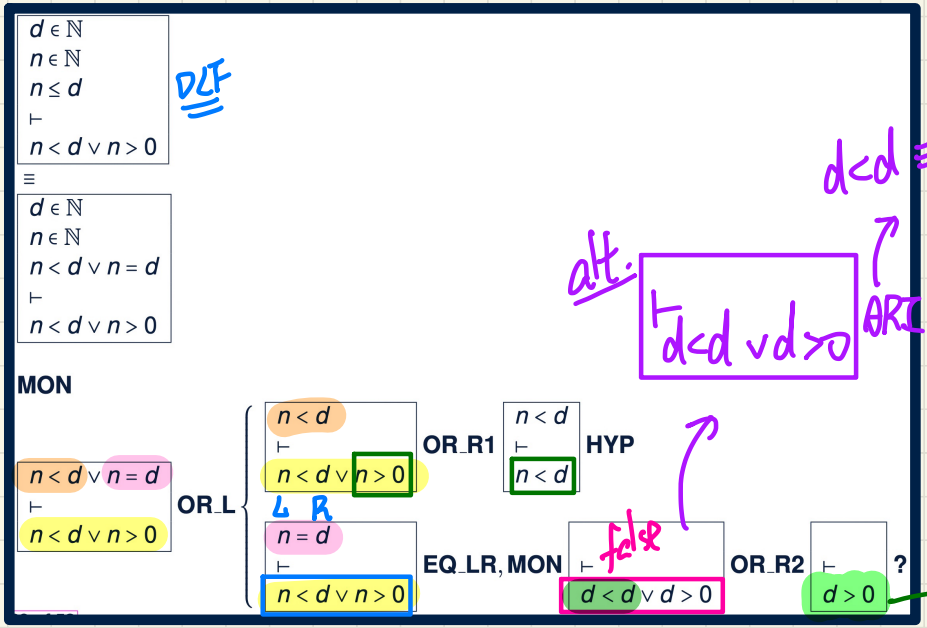
$$\frac{H(E), E = F \vdash P(E)}{H(F), E = F \vdash P(F)}\ \text{EQ\_RL}$$

Discharging PO of DLF: First Attempt

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

$$\frac{H1 \vdash G}{H1, H2 \vdash G}\ \text{MON} \qquad \frac{H, P \vdash R \quad H, Q \vdash R}{H, P \vee Q \vdash R}\ \text{OR\_L} \qquad \frac{H \vdash P}{H \vdash P \vee Q}\ \text{OR\_R1} \qquad \frac{H \vdash Q}{H \vdash P \vee Q}\ \text{OR\_R2}$$

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



d<d ≡ false

alt.

⊢ d<d ∨ d>0 — OR1

⊢ false ∨ d>0 — OR1

false ∨ P ≡ P

n < d ∨ n > 0

⊢ d>0

unprovable given no additional hypotheses

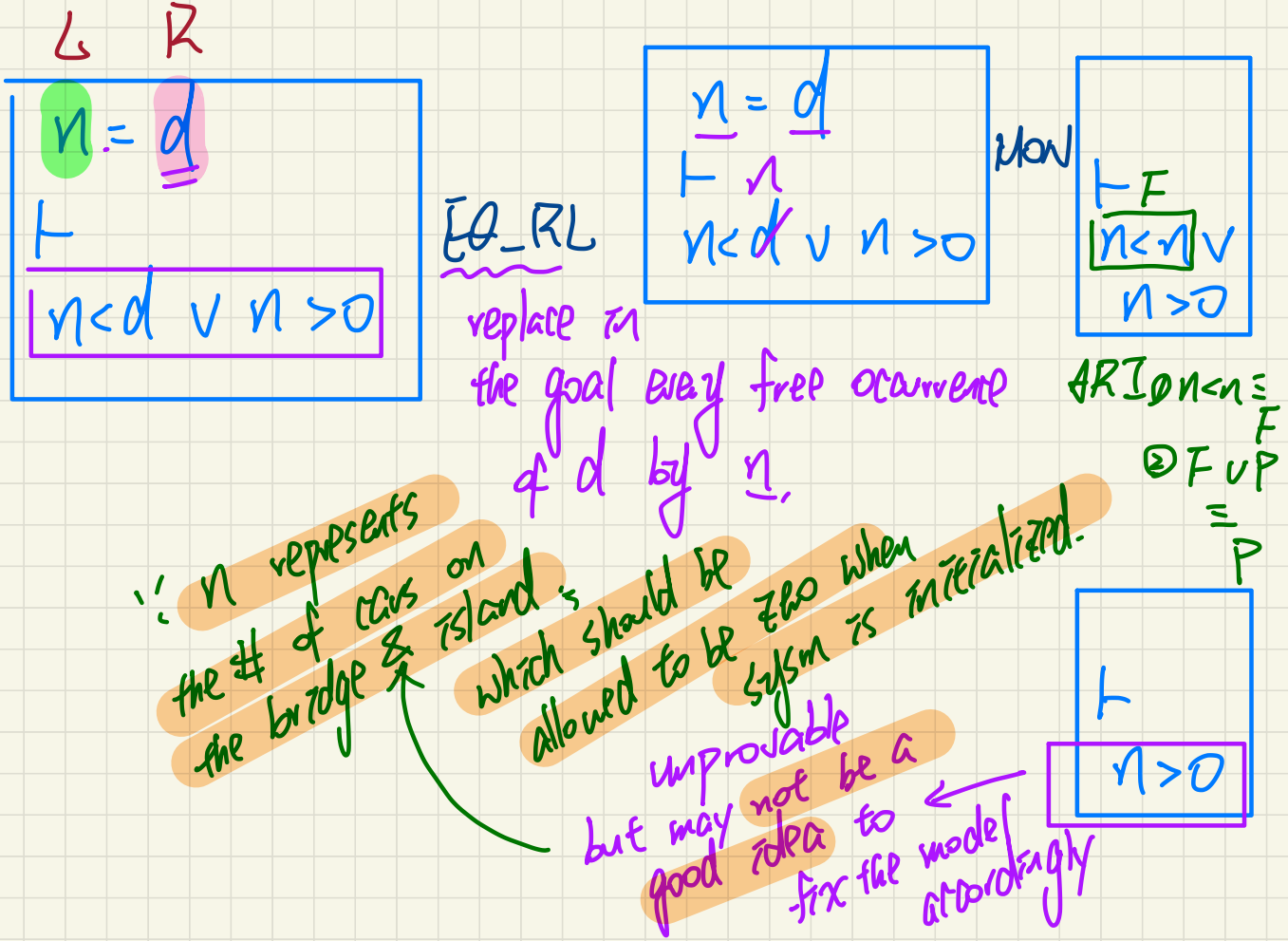
Lecture 17 - Nov. 7

Bridge Controller

Interpreting Unprovable DLF PO
First Refinement: Abstraction, State Space

Announcements/Reminders

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


L R
n = d
⊢
n < d ∨ n > 0
EQ_RL
replace n
the goal every free occurrence
of d by n.
n = d
⊢ n
n < d ∨ n > 0
MON
⊢ F
n < n ∨
n > 0
ARITH n < n ≡ F
② F ∨ P
≡ P
⊢
n > 0
! n represents
the # of ccis on
the bridge & island
which should be
allowed to be zero when
sysm is initialized.
unprovable
but may not be a
good idea to
fix the model
accordingly

Understanding the Failed Proof on DLF

constants: d

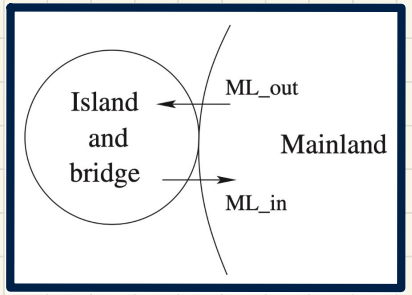
variables: n

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**



Unprovable Sequent: $\vdash d > 0$

$\vdash d > 0$

Not being able to prove $d > 0$
↳ current model may violate it: $\neg(d > 0)$ is true.

$0 < 0 \vee 0 > 0$
$F \quad F$
$\equiv F$

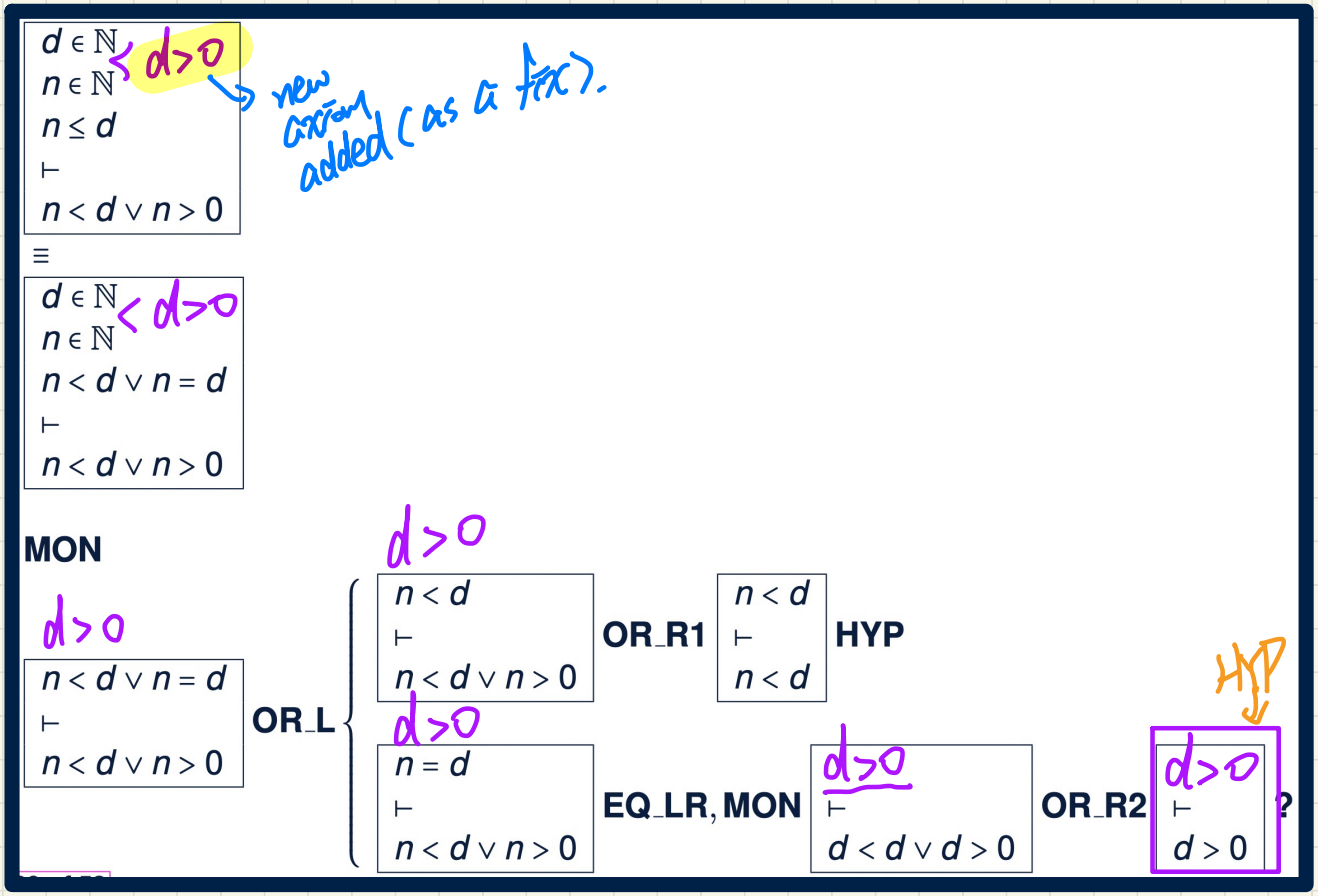
deadlocks right after init

Say $d = 0$
After init: $n = 0$
$G(ML\_out) \vee G(ML\_in)$
$n < d$ $n > 0$

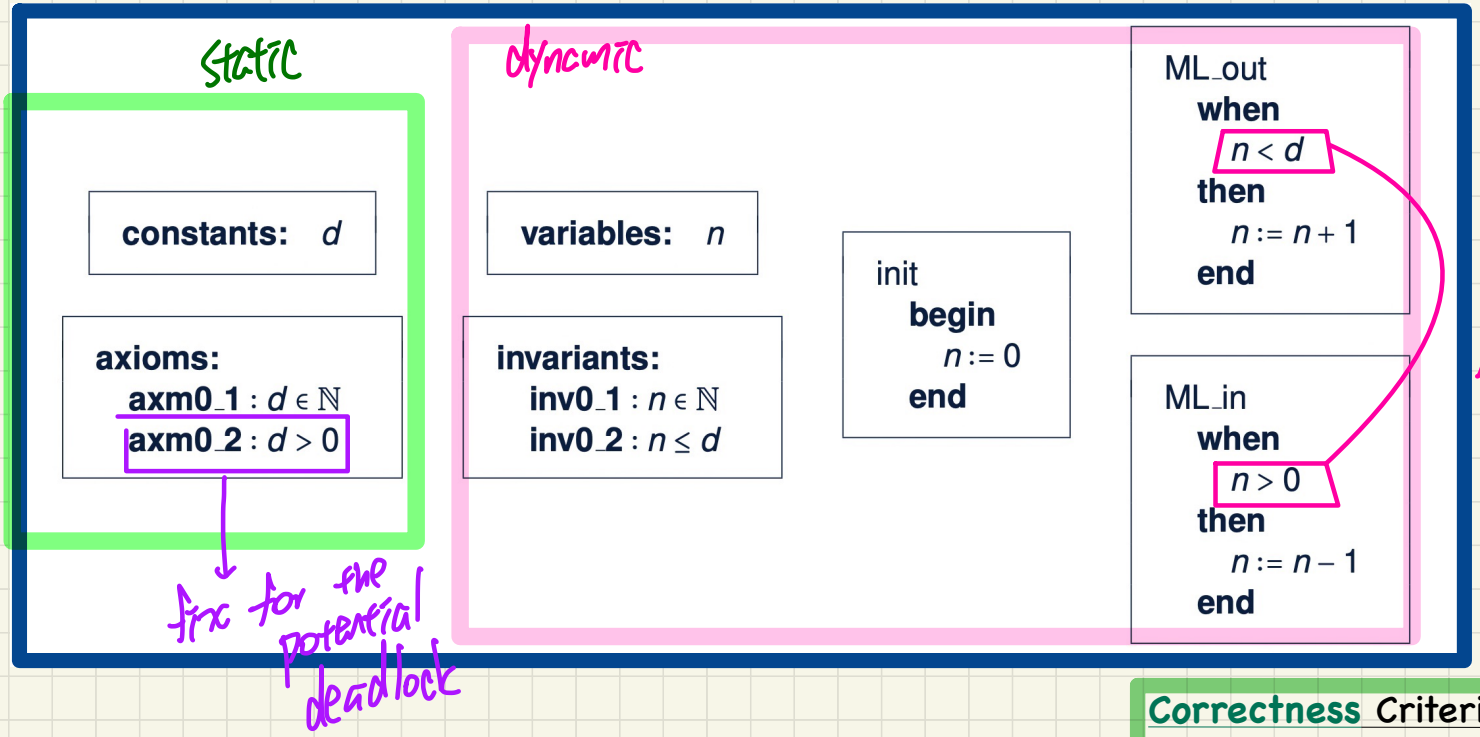
① $d \leq 0$
② $d \in \mathbb{N}$ $(d \geqslant 0)$

$d = 0$ allowed by current model

Discharging PO of DLF: Second Attempt



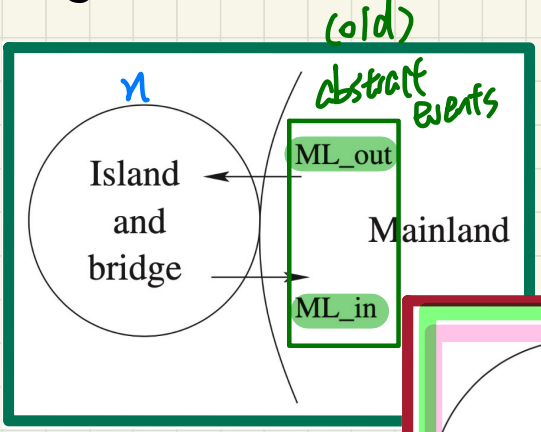
Summary of the Initial Model: Provably Correct



Correctness Criteria:

+ Invariant Establishment
+ Invariant Preservation
+ Deadlock Freedom

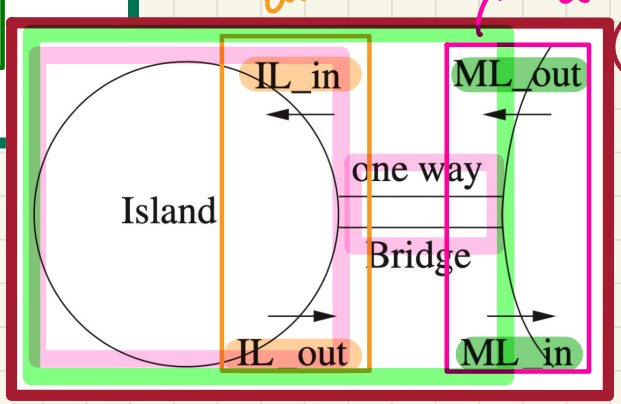
Bridge Controller: Abstraction in the 1st Refinement



m0:

initial, most abstract

added "concreteness": one-way bridge

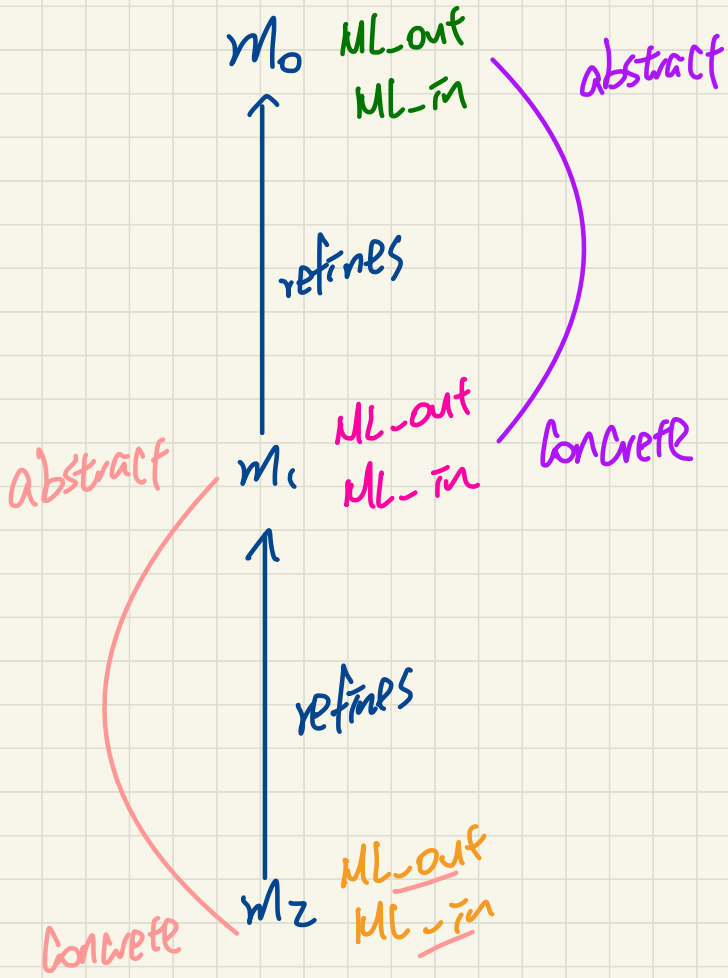


m1:

second, more concrete

traffic light still abstracted away

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


M_0
ML_out
ML_in
abstract
refines
M_1
ML_out
ML_in
concrete
abstract
refines
M_2
ML_out
ML_in
concrete

$* \quad a=0 \vee c=0 \equiv \neg(a\neq 0 \wedge c\neq 0)$

Bridge Controller: State Space of the 1st Refinement

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

Dynamic Part of Model

variables: a, b, c

invariants:

inv1_1 : $a \in \mathbb{N}$

inv1_2 : $b \in \mathbb{N}$

inv1_3 : $c \in \mathbb{N}$

inv1_4 : ??

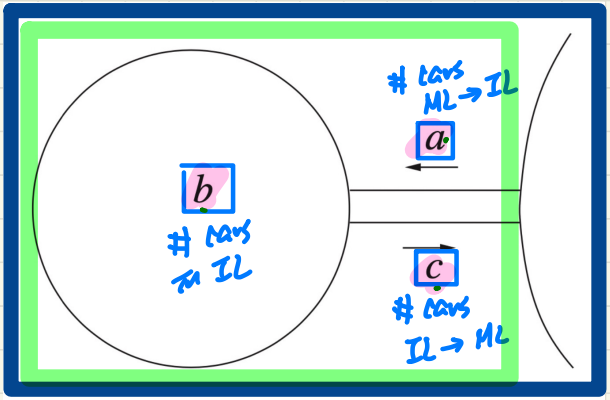
inv1_5 : ??

linking/gluing invariant

$n = a+b+c$

$a=0 \vee c=0$

def * $a*c=0$ may be hard for provers.



Static Part of Model

constants: d

axioms:

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

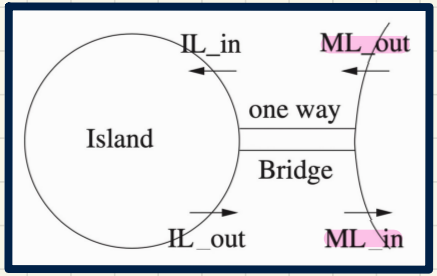
axm0_2 : $d > 0$

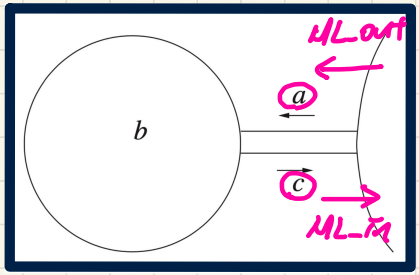
Exercises

inv1_4: linking abstract (n) & concrete (a, b, c) states

inv1_5: bridge is one-way

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





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

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

abstract guard from M0: n < d

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

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

(3) first concrete grd: c = 0

constants: d

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

** (1) (2) (3)

(2) a + b < d

(3) n < d

a + b + c = 0

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

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
    ??
  then
    c := c − 1
  end

Lecture 18 - Nov. 12

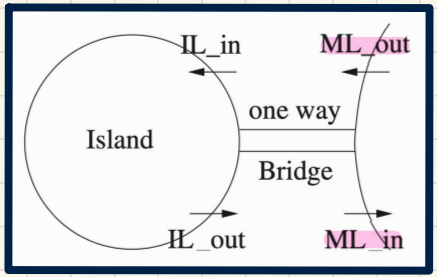
Bridge Controller

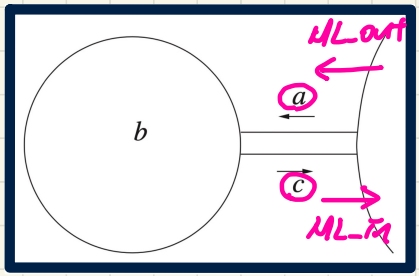
Abstract vs. Concrete Transitions
Predicates: Stronger vs. Weaker
Why Guard Strengthening?

Announcements/Reminders

- ProgTest2 results to be released by Monday, Nov 18
- Guide to be released for WrittenTest2 on Wednesday
- No scheduled lab session tomorrow.

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





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

```
ML_out
  when
    c = 0
    ??
  then
    a := a + 1
  end
```

abstract guard from Mo: $n < d$

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

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

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

constants: d

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

$a + b \le d$

$n < d$

$a + b + c = n$

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

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

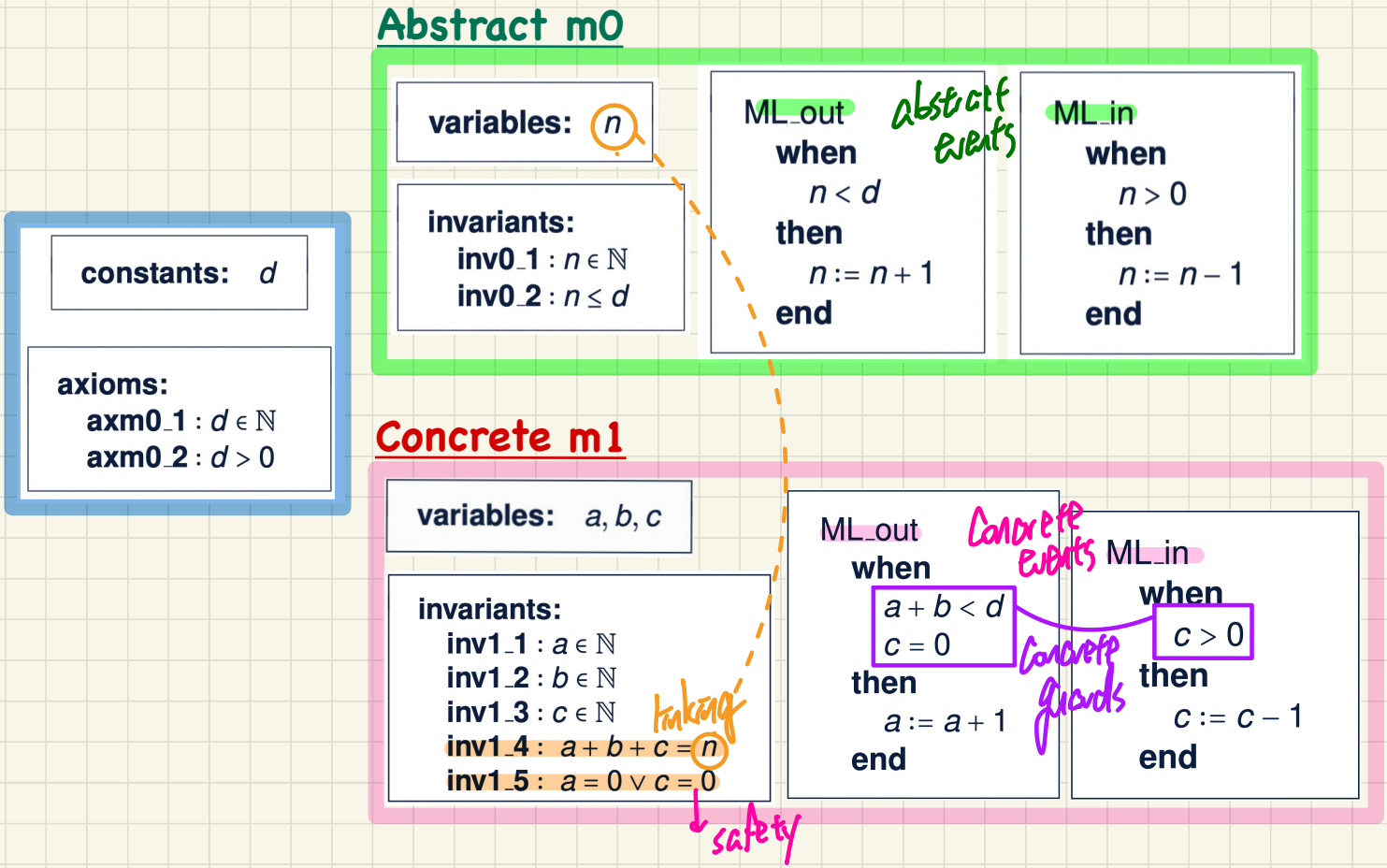
Q. Is it necessary to add a new guard "$a = 0$"?

No. $c > 0$

$a = 0 \lor c = 0$

$a = 0$

States, Invariants, Events: Abstract vs. Concrete



Consider : <init, ML_out, ML_in> → execs

Bridge Controller: Abstract vs. Concrete State Transitions

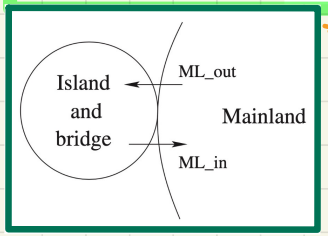
Abstract m0

variables: n

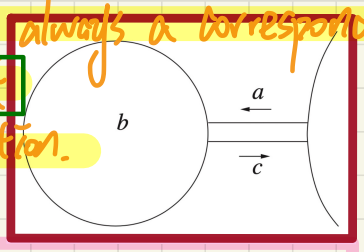
invariants:
- inv0_1 : $n \in \mathbb{N}$
- inv0_2 : $n \leq d$

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

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



* For each concrete transition, there's always a corresponding abstract transition.



Concrete m1

variables: a, b, c

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

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

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

abstract init. state

init → d = 2, n = 0

(abs) ML_out → d = 2, n = 1

inv1_4: 0+0+0=0 ✓

inv1_4: 1+0+0=1 ✓

d = 2
n initialized to 0

Scenario
- car leaving ML
- car entering ML

d = 2
a, b, c initialized to 0

init → d = 2, a = 0, b = 0, c = 0

ML_out (conc.) → d = 2, a = 1, b = 0, c = 0

Concrete init state

Before-After Predicates of Event Actions: 1st Refinement

Concrete

Events

ML_in

when

$0 < c$

then

$c := c - 1$

end

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

ML_out

when

$a + b < d$

$c = 0$

then

$a := a + 1$

end

Recall: concrete variables are: a, b, c

- Pre-State
- Post-State
- Sate Transition

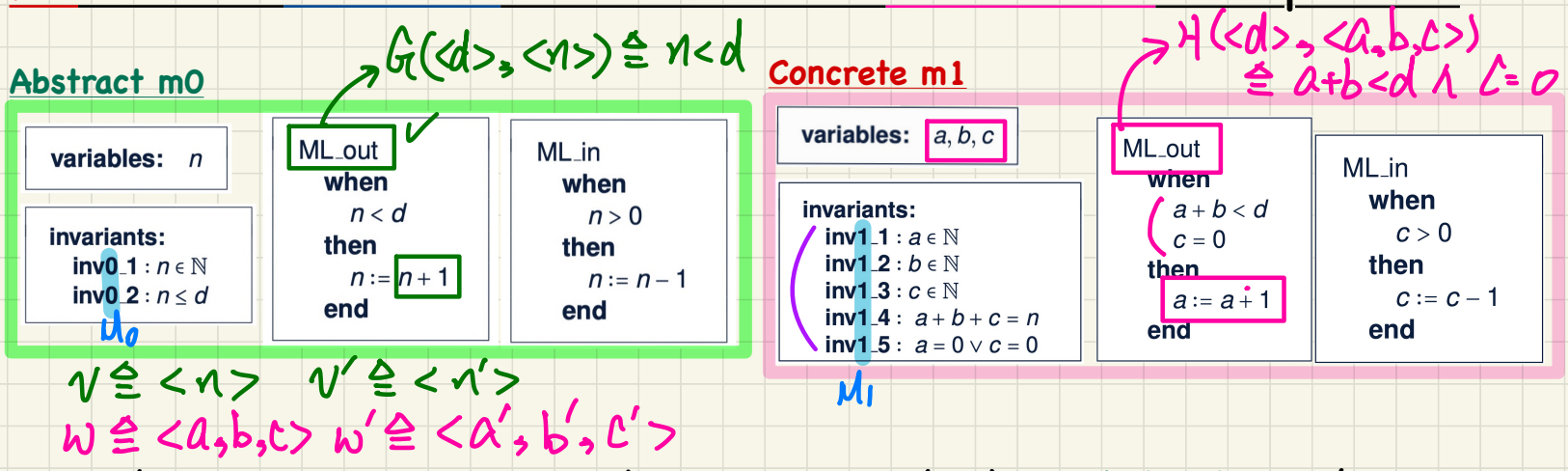
Before–after predicates

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

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

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

PO Rule of Invariant Preservation in Refinement: Components



v and v': **abstract** variables in pre-/post-states

w and w': **concrete** variables in pre-/post-states

I(c, v): list of **abstract** invariants

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

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

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

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

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

E(<d>, <n>) of ML_out : <n+1>

F(<d>, <a,b,c>) of ML_out : <a+1, b, c>

J(<d>, <n>, <a, b, c>) ≙ <inv1_1, ..., inv1_5>

Predicates: Weaker vs. Stronger

$P \Rightarrow Q$

↳ P is "stronger" than Q

↳ Q is "weaker" than P

$P(x) \triangleq x > 0$

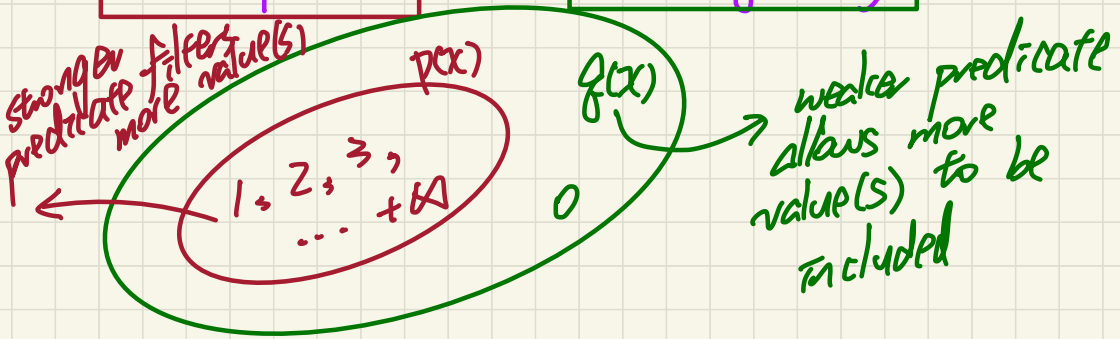
$Q(x) \triangleq x \geq 0$

$P(x) \Rightarrow Q(x)$

$Q(x) \not\Rightarrow P(x)$

Satisfying Values

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



Refinement: Why Guard Strengthening

(3) $\neg G \Rightarrow \neg H$

$\neg G$: abstract event disabled

$\neg H$: corresponding concrete event disabled

(2) $H \Rightarrow G$

H: concrete guard

G: abstract guard

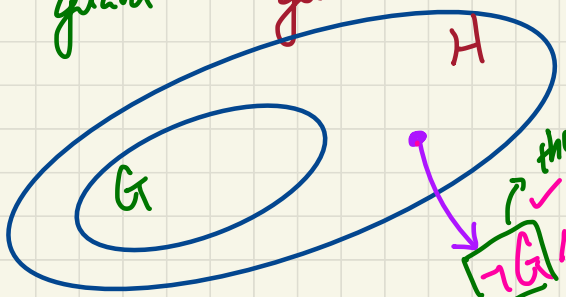
if a concrete transition enabled, the corresponding abstract transition is always enabled as well

Principle: a refinement's (M_1) behaviour should be consistent with the abstract model's (M_0) behaviour (⇒ no new behaviour introduced by the concrete events) (⇒ no new behaviour by ref.).

(1) $G \Rightarrow H$: Why not?

G: abstract guard

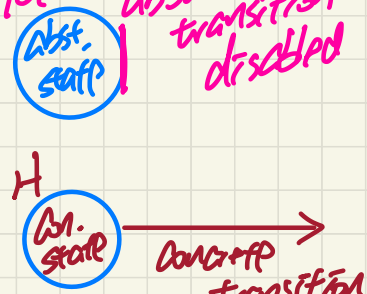
H: concrete guard



$\neg G \wedge H$

$\neg G$: the abstract version of event is disabled

H: the corresponding concrete version of event is enabled.



PO/VC Rule of Guard Strengthening: 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

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

$A(c)$ ✓
$I(c, v)$ ✓
$J(c, v, w)$
$H(c, w)$
$\vdash$
$G_i(c, v)$ ← ML_out, ML_in.

Concrete m1

variables: a, b, c

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

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

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

ML_out/GRD
$d \in \mathbb{N}$ axm0_1
$d > 0$ axm0_2
$n \in \mathbb{N}$ inv0_1
$n \leq d$ inv0_2
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \vee c = 0$
$a + b < d \quad c = 0$

ML_in/GRD ← exercise.

$\vdash n < d$

Q. How many PO/VC rules for model m1? → # of concrete guards

Lecture 19 - Nov. 14

Bridge Controller

Discharging Guard Strengthening POs
Invariant Preservation: Concrete Events
Commuting Diagram: Simulation
New Events: IL_in, IL_out

Announcements/Reminders

- **Lab5** to be released on by next Tuesday's class (Nov 19)
  (due on Tuesday, December 3)
- **WrittenTest2** next Wednesday, November 20
  + Guide
  + Practice Questions
- **Bonus** Opportunity coming: Formal Course Evaluation

Discharging POs of m1: Guard Strengthening in Refinement

ML_out/GRD

$$\frac{H1 \vdash G}{H1, H2 \vdash G}\ \text{MON} \qquad \frac{}{H, P \vdash P}\ \text{HYP}$$

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

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

abstract I
- $n \in \mathbb{N}$
- $n \leq d$

Con. Inv.
- $a \in \mathbb{N}$
- $b \in \mathbb{N}$
- $c \in \mathbb{N}$
- $a + b + c = n$
- $a = 0 \vee c = 0$

Con. guards of ML_out
- $a + b < d$
- $c = 0$

$\vdash$

abst. guard of ML_out
- $n < d$

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

EQ_LR, MON → $a+b+0=n,\ a+b<d \vdash n<d$

$a+b+0 = a+b=n$ ↗ ARI

$a+b=n,\ a+b<d \vdash n<d$ (R)

EQ_LR, MON → $n<d \vdash n<d$

HYP

Discharging POs of m1: Guard Strengthening in Refinement

ML_in/GRD

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

$$\frac{H1 \vdash G}{H1, H2 \vdash G}\ \text{MON} \qquad \frac{}{H, P \vdash P}\ \text{HYP} \qquad \frac{}{\bot \vdash P}\ \text{FALSE\_L}$$

$$\frac{H(F), E = F \vdash P(F)}{H(E), E = F \vdash P(E)}\ \text{EQ\_LR} \qquad \frac{H, P \vdash R \quad H, Q \vdash R}{H, P \vee Q \vdash R}\ \text{OR\_L}$$

ARI
$n>0$
$\vdash$
$n>0$
HYP

MON

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

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

$c \leq n$
$c>0$
$\vdash$
$n>0$

*

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

MON

$c=0$
$c>0$
$\vdash$
$n>0$

$c=0 \wedge c>0$
$\equiv \bot$

ARI

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

FALSE_L

for concrete evts

PO/VC Rule of Invariant Preservation: Sequents

Abstract m0 * $a'+b'+c'=n'$ $(a+1)+b+c=n+1$

variables: n

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

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

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

$A(c)$ axioms
$I(c, v)$ abst. I
$J(c, v, w)$ conc. I.
$H(c, w)$ con. guard
$\vdash$
$J_i(c, E(c, v), F(c, w))$ $i \in 1..5$

Concrete m1 ** $a'=0 \vee c'=0$ $a=0 \vee c-1=0$

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 $a=0 \vee c=0$

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

ML_in
when $c > 0$
then $c := c-1$
end $c'=c-1 \wedge a'=a$, $b'=b$

Concrete invariant satisfied in the post state.

ML_out/inv1_4/INV
- $d \in \mathbb{N}$
- $d > 0$
- $n \in \mathbb{N}$
- $n \le d$
- $a \in \mathbb{N}$ $b \in \mathbb{N}$ $c \in \mathbb{N}$
- $a+b+c=n$
- $a=0 \vee c=0$
- $a+b<d$
- $c=0$
- $\vdash$
- * $(a+1)+b+c=n+1$

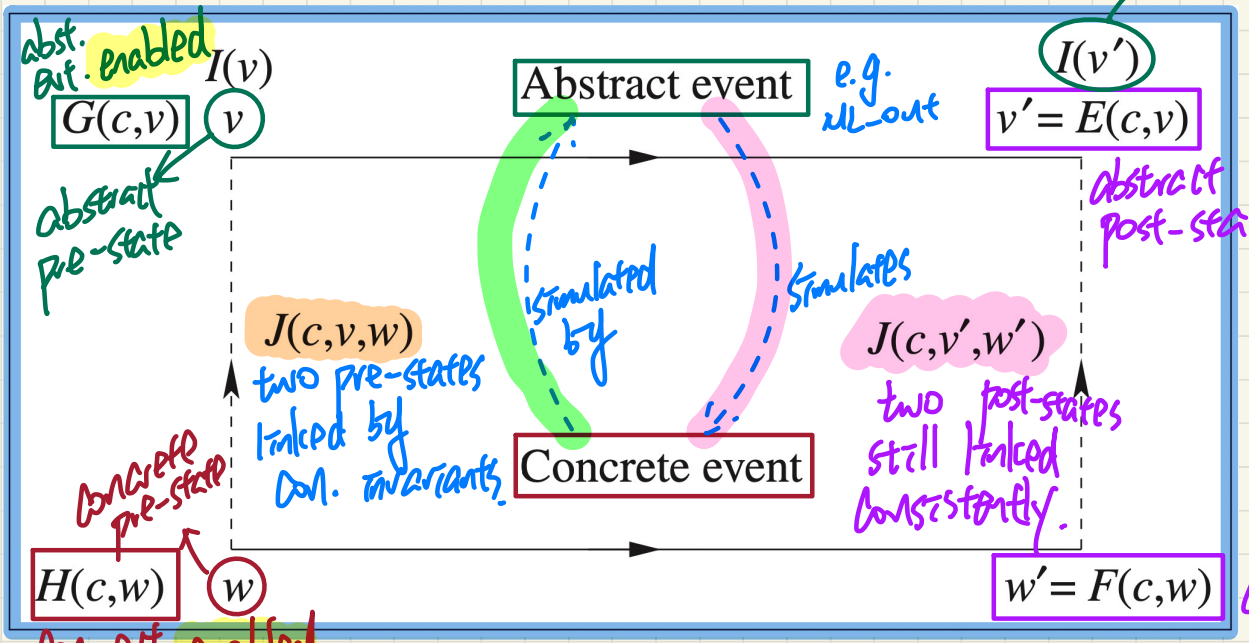
ML_in/inv1_5/INV
- $d \in \mathbb{N}$
- $d > 0$
- $n \in \mathbb{N}$
- $n \le d$
- $a \in \mathbb{N}$ $b \in \mathbb{N}$ $c \in \mathbb{N}$
- $a+b+c=n$
- $a=0 \vee c=0$
- $c>0$
- $\vdash$
- ** $a=0 \vee c-1=0$

2 con. evts * 5 inv. = 10

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

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

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

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

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

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

Discharging POs of m1: Invariant Preservation in Refinement

ML_in/inv1_5/INV

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

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

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

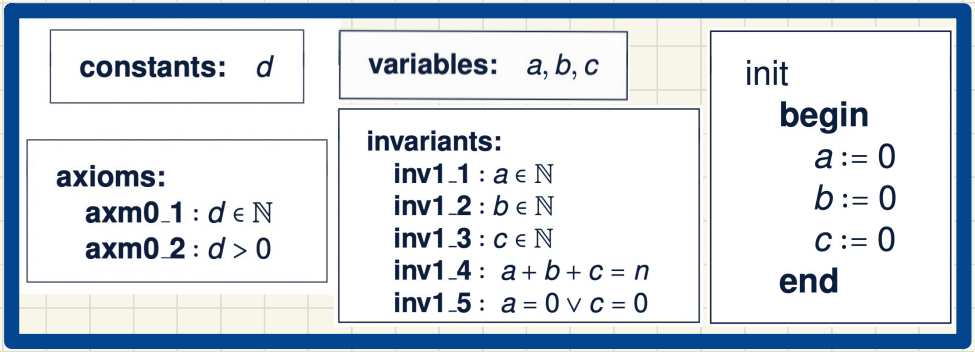
$$\frac{H1 \vdash G}{H1, H2 \vdash G}\ \text{MON}$$

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

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

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

PO of Invariant Establishment in Refinement



Components

K(c): effect of abstract init

L(c): effect of concrete init

Rule of Invariant Establishment

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

Exercise:

Generate Sequents from the INV rule.

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

Discharging PO of Invariant Establishment in Refinement

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

init/inv1_4/INV

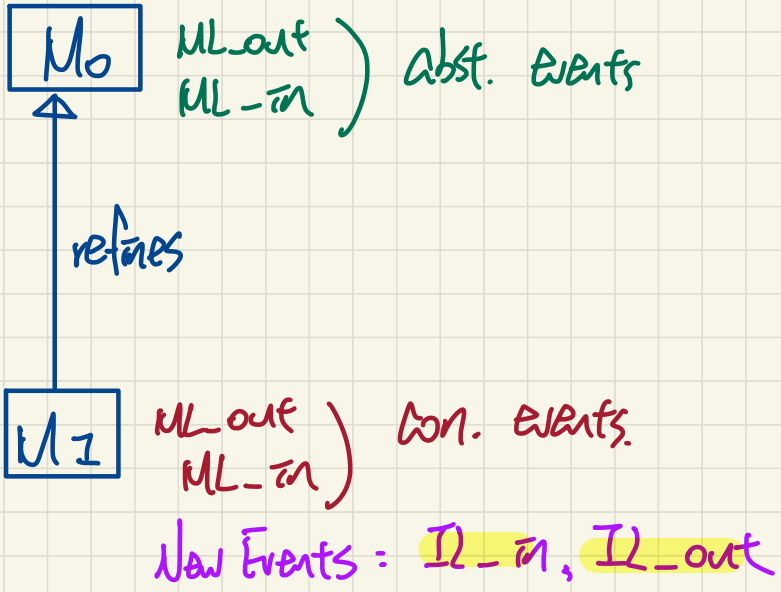
$$\frac{H1 \vdash G}{H1, H2 \vdash G}\ \text{MON}$$

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

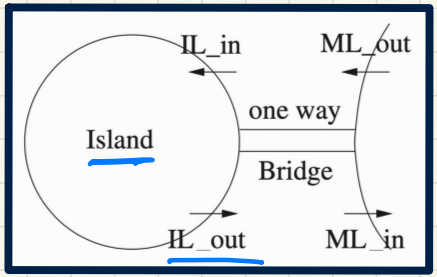
$$d \in \mathbb{N}$$
$$d > 0$$
$$\vdash$$
$$0 = 0 \lor 0 = 0$$

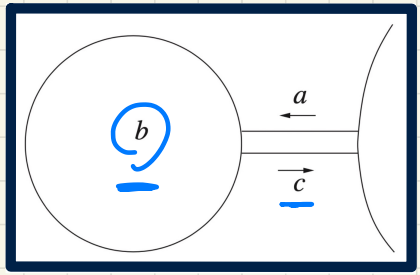
init/inv1_5/INV

Events


M0
ML_out
ML_in
abst. events
refines
M1
ML_out
ML_in
con. events
New Events : IL_in, IL_out

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





constants: d

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

variables: a, b, c

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

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

IL_in
when
??
then
??
end

① $a > 0$
② $c = 0$

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

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

IL_out
when
??
then
??
end

① $b > 0$
② $a = 0$

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

Before-After Predicates of Event Actions: 1st Refinement

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

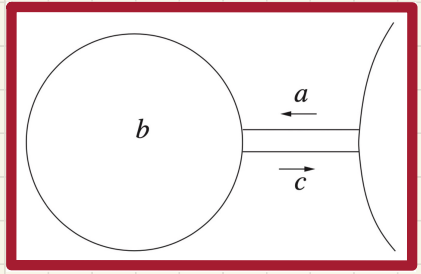
$a' = a - 1$

$\wedge\ b' = b + 1$

$\wedge\ c' = c$

- Pre-State
- Post-State
- Sate Transition

Concrete State Space

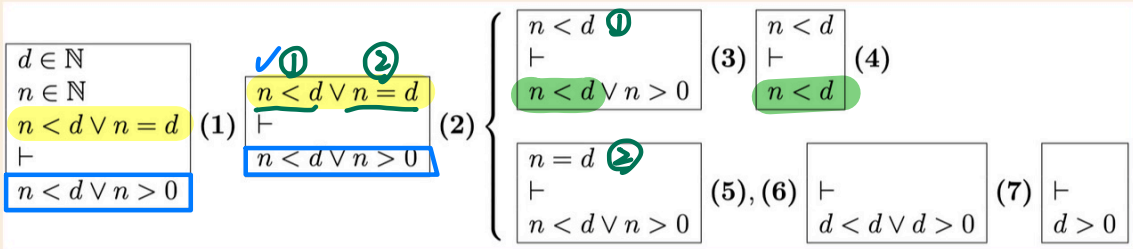


Review Q & A - Nov. 17

Written Test 2

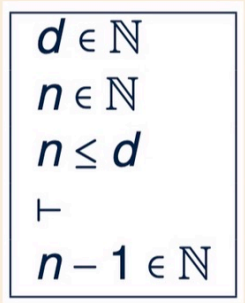
Bridge Controller Lecture Practice Questions

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

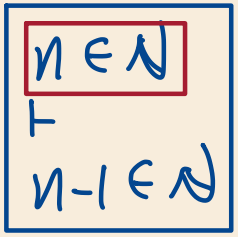


Each of Labels (1) to (7) denotes the justification for transforming the two neighboring sequents. Drag and drop the appropriate inference rule; otherwise, drag and drop "NONE" to indicate that no rule can be used for the transformation.

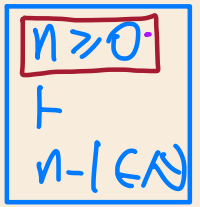
- (1): [MON]
- (2): [OR_L]
- (3): [OR_R1]
- (4): [HYP]
- (5): [EQ_LR]
- (6): [MON]
- (7): [OR_R2]



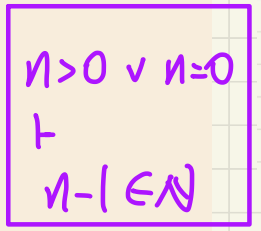
MON



P3 ARI



ARI



You are required to attempt proving the above sequent, using the inference rules as listed here: https://www.eecs.yorku.ca/~jackie/teaching/lectures/2022/W/EECS3342/notes/inference-rule-examples-1.pdf

- 1st inference rule to use (if applicable): [MON]
- 2nd inference rule to use (if applicable): [?]
- 3rd inference rule to use (if applicable): [n.a.]
- 4th inference rule to use (if applicable): [n.a.]

Q13. $n = 10$ should not be chosen.

Lecture 20 - Nov. 19

Bridge Controller

Concrete, New Events vs. Abstract skip
Livelock, Divergence
Invariant vs. Variant
Tracing of Abstract/Concrete Transitions

Announcements/Reminders

- Lab5 released (due on Tuesday, December 3)
- WrittenTest2 tomorrow
- Bonus Opportunity coming: Formal Course Evaluation

old events and new events.

PO/VC Rule of Invariant Preservation: Sequents

Abstract m0

constants: d

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

variables: n

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

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

Concrete m1

variables: a, b, c

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

IL_in (new event) — corresponds to "skip", $n' = n$
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

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

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

IL_in/INV1_4/INV

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

$a + b + c = n$
$a = 0 \vee c = 0$
$a > 0$

IL_in/INV1_5/INV
(exercise)

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

new events (2) * # invariants (5) = 10.

Discharging POs of m1: Invariant Preservation in Refinement

IL_in/inv1_4/INV

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

$$\frac{H1 \vdash G}{H1, H2 \vdash G}\ \textbf{MON} \qquad \frac{}{H, P \vdash P}\ \textbf{HYP}$$

Discharging POs of m1: Invariant Preservation in Refinement

ML_in/inv1_5/INV

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

$$\frac{}{\bot \vdash P}\ \textbf{FALSE\_L} \qquad \frac{H1 \vdash G}{H1, H2 \vdash G}\ \textbf{MON} \qquad \frac{H \vdash Q}{H \vdash P \lor Q}\ \textbf{OR\_R2}$$

$$\frac{}{H, P \vdash P}\ \textbf{HYP} \qquad \frac{H(F), E = F \vdash P(F)}{H(E), E = F \vdash P(E)}\ \textbf{EQ\_LR} \qquad \frac{H, P \vdash R \quad H, Q \vdash R}{H, P \lor Q \vdash R}\ \textbf{OR\_L}$$

new events : IL_in, IL_out.

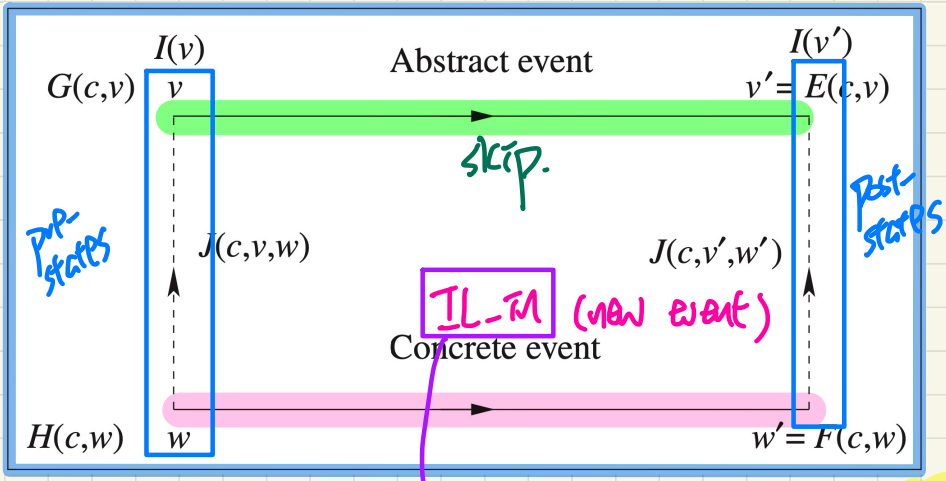
Visualizing Invariant Preservation in Refinement

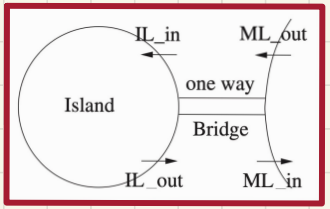
Each **new** state transition (from w to w')
should be simulated by
an **abstract dummy** state transition (from v to v')

abstract event simulating any new event

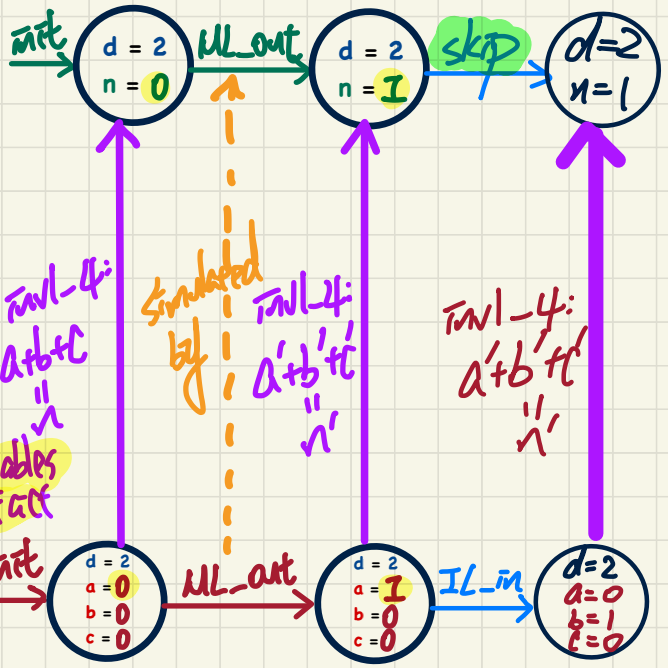
skip
begin
end

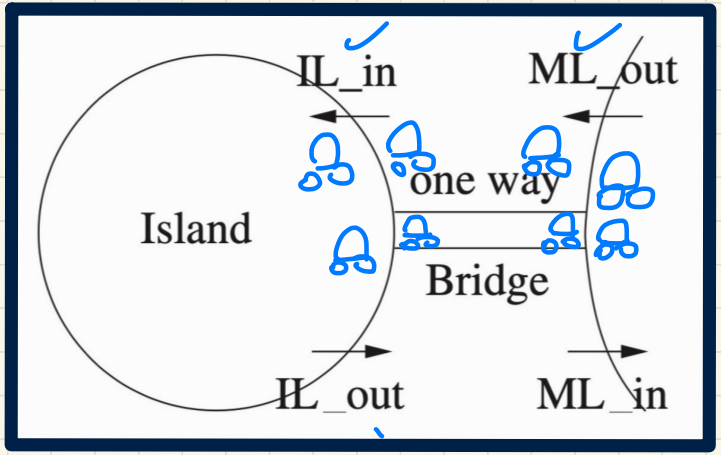
n'=n
BAP.





1. only modifies concrete variables (a, b, c) — variables intact
2. as if new event kept the abstract





Exercise Show the abstract & concrete transitions of:

< init , ML_out , IL_in , IL_out , ML_in >

concrete transitions

< init , ML_out , skip , skip , ML_in >

abstract transitions

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

(unconditional)

while(1){ }

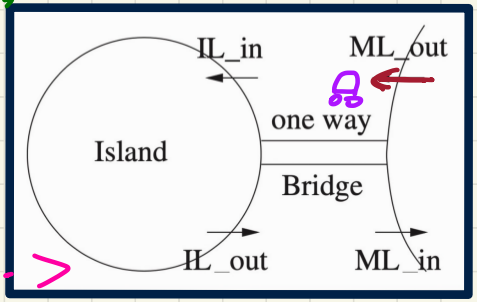
solution: have a measure on imposing an upper bound on the # of interleavings of new events

Abstract transitions: <init, ML_out, skip, skip, skip, skip, ... >

new events occurring indefinitely.

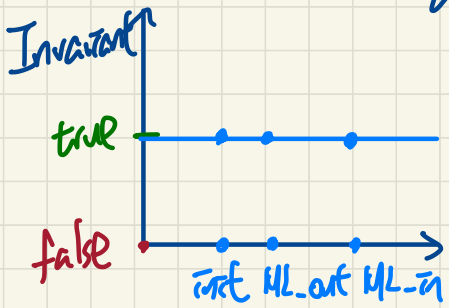
Concrete transitions: <init, ML_out, IL_in, IL_out, IL_in, IL_out, ... >

a possible trace that's problematic.

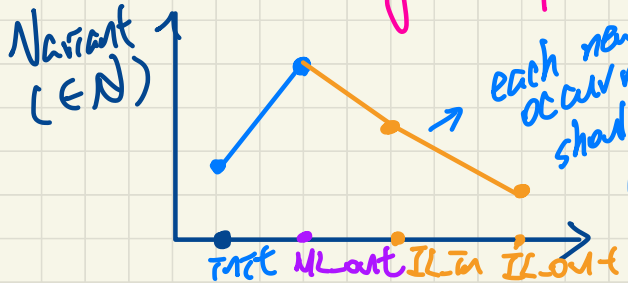


Invariant vs. Variant

Invariant : Boolean expression that should always hold (after each event occurrence)



Variant : Integer expression that may change after event occurrence.



each new event occurrence should strictly decrease the value of variant ⇒ # new events in interleaving is finite ⇒ no livelock

Use of a Variant to Measure New Events Converging

fixed

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

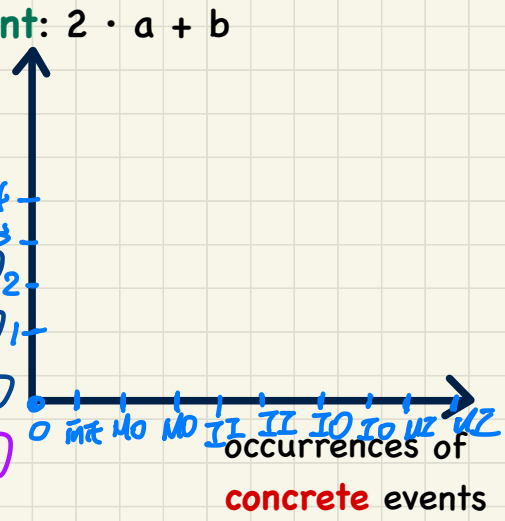
ML_in
when
$c > 0$
then
$c := c - 1$
end

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

Variants for New Events: 2 · a + b

<init,	ML_out,	ML_out,	IL_in,	IL_in,	IL_out,	IL_out,	ML_in,	ML_in >
a = 0	a = 1	a = 2	a = 1	a = 0	a = 0	a = 0	a = 0	a = 0
b = 0	b = 0	b = 0	b = 1	b = 2	b = 1	b = 0	b = 0	b = 0
c = 0	c = 0	c = 0	c = 0	c = 0	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



Lecture 21 - Nov. 21

Bridge Controller

Proof Obligations of System Variant

Announcements/Reminders

- **Lab5** released (due on Tuesday, December 3)
- **WrittenTest2** results to be released on Wednesday
- **Exam** review sessions polling
- **Bonus** Opportunity coming: Formal Course Evaluation

LiveLock / Divergence

→ Caused by an infinite interleaving of

new events / Concrete model, busy looping in the abstract model

→ Variant ($\in \mathbb{N}$) is

~ not the cause of livelock of a model

~ just a measure on if livelock is present in your model

→ 2 POs → provable means the model livelocks & needs to be fixed

Use of a Variant to Measure New Events Converging

fixed

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

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

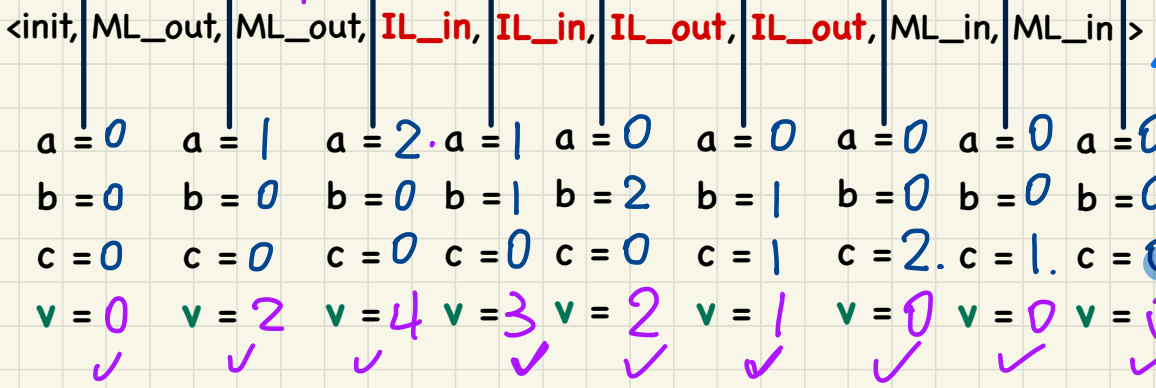
IL_in
when
$a > 0$
then
$a := a - 1$
$b := b + 1$
end

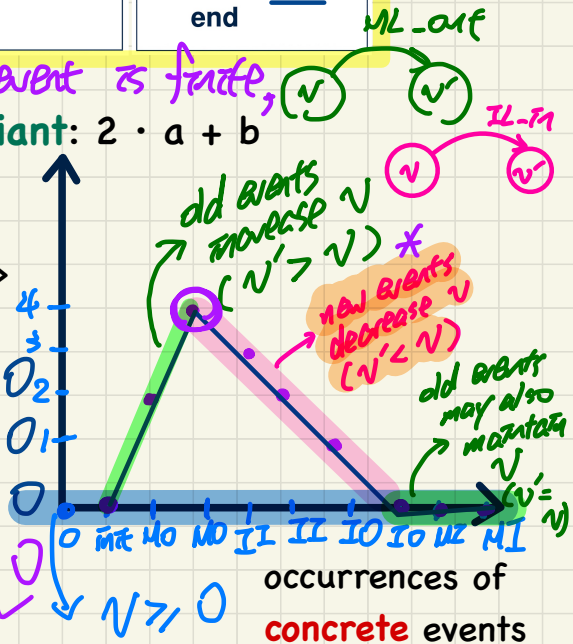
IL_out
when
$b > 0$
$a = 0$
then
$b := b - 1$
$c := c + 1$
end

* Given that the starting value of v of the first new event is finite,

Variants for New Events: 2 · a + b

the # of interleaved new events is also finite.





PO of Convergence/Non-Divergence/Livelock Freedom

Variants for New Events: 2 · a + b

Variant Stays Non-Negative

$A(c)$
$I(c, v)$
$J(c, v, w)$
$H(c, w)$
$\vdash$ $V(c, w) \in \mathbb{N}$

guard of new event

NAT

IL_in/NAT

new event

$v \geq 0$

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

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

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

* $v' < v$ $(b+1)$

$2 \cdot a' + b' < 2 \cdot a + b$

$(a-1)$

variant: V(c, w)

v

$V(c, w)$

v

$V(c, F(c, w))$

IL_in/NAT

new event

occurrences of **new** events

A New Event Occurrence Decreases Variant

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

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

effect of new event

v v

VAR

IL_in/VAR

new event

v's value strictly decreased.

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

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

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

Exercise Given variant: $a + b$

(1) Re-trace the value of v using the same trace (and plot the diagram). Can the same patterns be observed?

(2) Formulate the VAR and VAT POs. ($2 * 2 = 4$ sequents)

VAT, VAR — IL-in, IL-out

(3) Are they provable?

Example Inference Rules

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

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

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

$H \Rightarrow P \vee Q$

$\equiv \{ P \equiv \neg\neg P \}$

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

$\equiv \{ P \Rightarrow Q \equiv \neg P \vee Q \}$

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

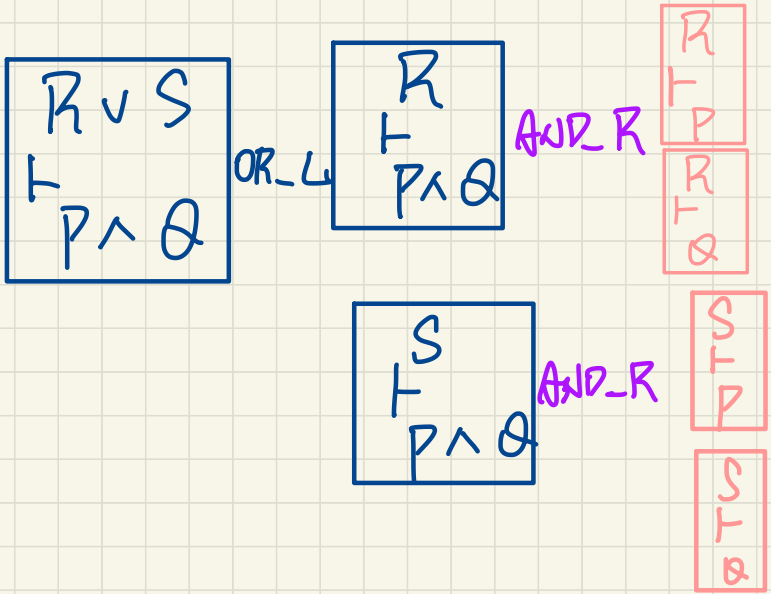
$\equiv \{$ shunting $\}$

$H \wedge \neg P \Rightarrow Q$

Question

$R \vee S \vdash P \wedge Q$

# of maximum sequents to prove instead?



Lecture 22 - Nov. 26

Bridge Controller

Relative Deadlock Freedom
2nd Refinement: Variables & Invariants
Exam Info

Announcements/Reminders

- Lab5 released (due on Tuesday, December 3)
- WrittenTest2 results to be released tomorrow
- Exam review sessions polling

Idea of Relative Deadlock Freedom

$A(c)$
$I(c, v)$
$J(c, v, w)$

$G_1(c, v) \vee \cdots \vee G_m(c, v)$

$\vdash$

$H_1(c, w) \vee \cdots \vee H_n(c, w)$

disjunction of all abstract guards

disjunction of all concrete guards

DLF

relative DL freedom in concrete model.

Guard Strengthening (GRD)

H_i ⊢ G_i

concrete guard of some event

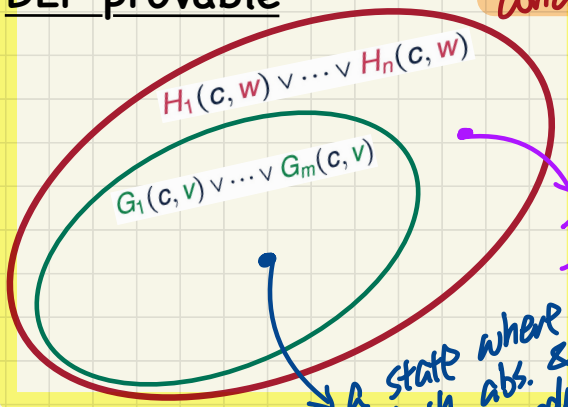
abstract guard

Principles

1. DL is bad!
2. a refinement should not introduce a DL scenario not existing in the abstract model.

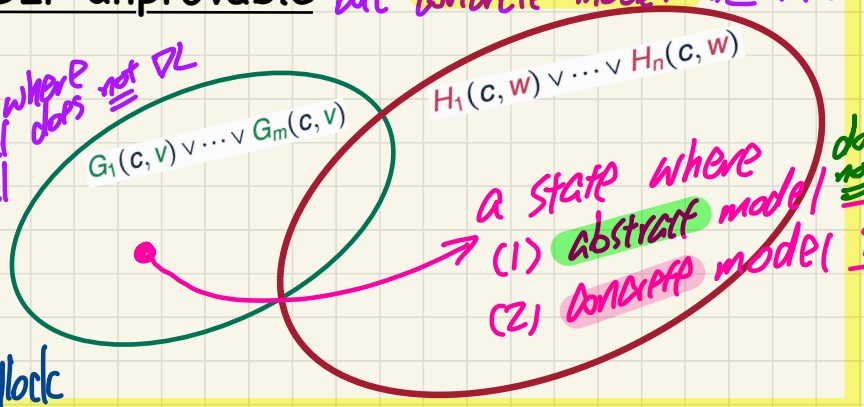
It's unacceptable if there's a state s_1 s.t. abstract model does not DL in s_1 but concrete model DL in s_1.

DLF provable



a state where
- con. model does not DL
- abs. model DL

DLF unprovable



PO of Relative Deadlock Freedom

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_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

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

DLF

$d \in \mathbb{N}$, $d > 0$) A
$n \in \mathbb{N}$, $n \le d$) I
$a \in \mathbb{N}$, $b \in \mathbb{N}$, $c \in \mathbb{N}$, $a + b + c = n$, $a = 0 \lor c = 0$) J

$n < d \lor n > 0$

$(a + b < d \land c = 0) \lor (c > 0) \lor (a > 0) \lor (b > 0 \land a = 0)$

Discharging POs of m1: Relative Deadlock Freedom

Part 1

$$\frac{H1 \vdash G}{H1, H2 \vdash G}\ \textbf{MON} \qquad \frac{H(F), E = F \vdash P(F)}{H(E), E = F \vdash P(E)}\ \textbf{EQ\_LR} \qquad \frac{H, \neg P \vdash Q}{H \vdash P \vee Q}\ \textbf{OR\_R}$$

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

Discharging POs of m1: Relative Deadlock Freedom

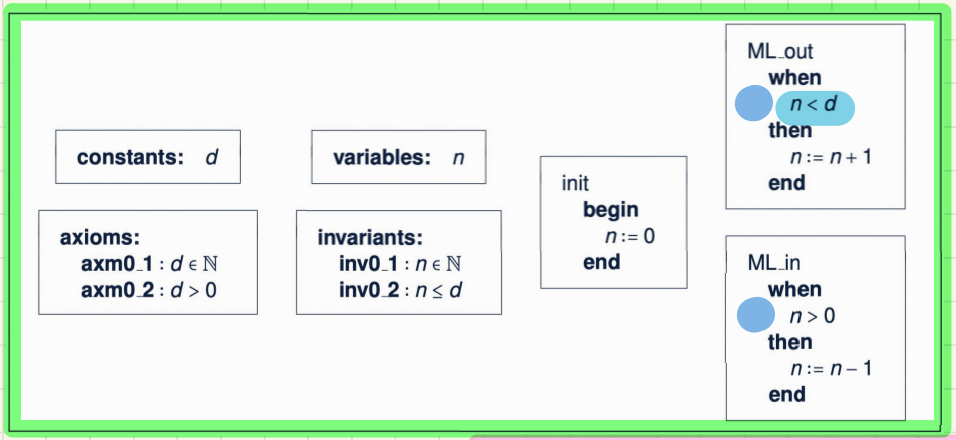
Part 2

$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, P \vee Q \vdash R}\ \text{OR\_L} \qquad \frac{H \vdash P}{H \vdash P \vee Q}\ \text{OR\_R1} \qquad \frac{}{P \vdash E = E}\ \text{EQ}$$

$$\frac{H \vdash P \qquad H \vdash Q}{H \vdash P \wedge Q}\ \text{AND\_R} \qquad \frac{H \vdash Q}{H \vdash P \vee Q}\ \text{OR\_R2} \qquad \frac{}{H, P \vdash P}\ \text{HYP}$$

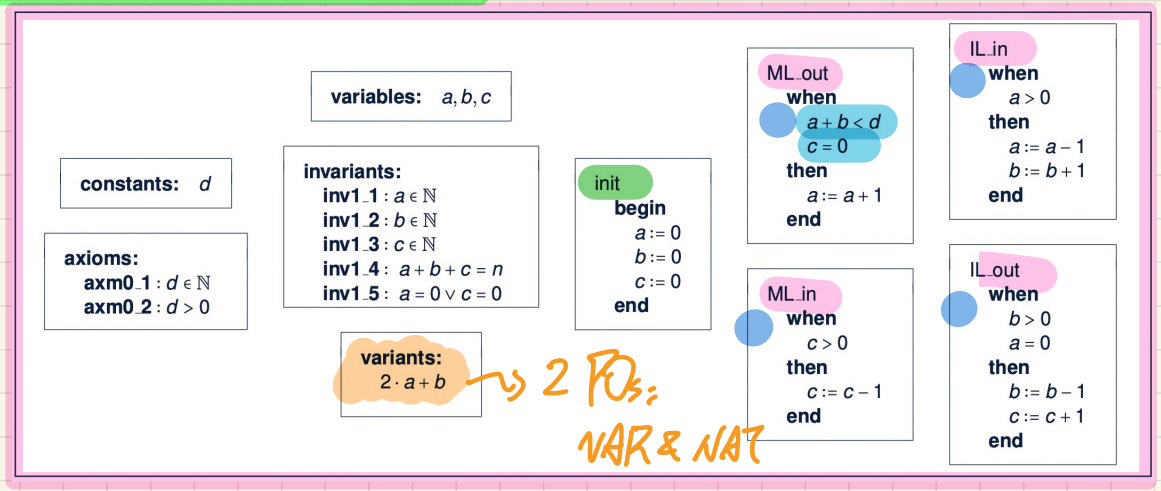
$$d > 0$$
$$b = 0 \vee b > 0$$
$$\vdash$$
$$b < d \wedge 0 = 0$$
$$\vee \quad b > 0 \wedge 0 = 0$$

Initial Model and 1st Refinement: Provably Correct



Abstract m0

Concrete m1



2 POs: NAT & NAT

Correctness Criteria:

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

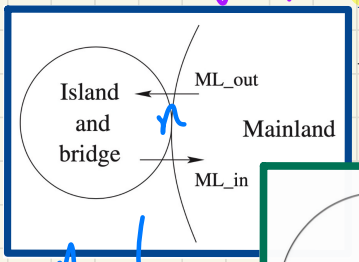
Bridge Controller: Abstraction in the 2nd Refinement

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.

without these assumptions, may hard to predict the model

m0:

more abstract than m1

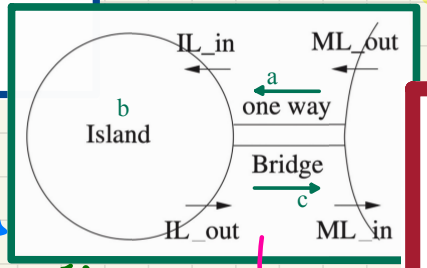


correctness of refinement

1. INV. est. & preservation 4. convergence
2. guard strengthening
3. relative DLF

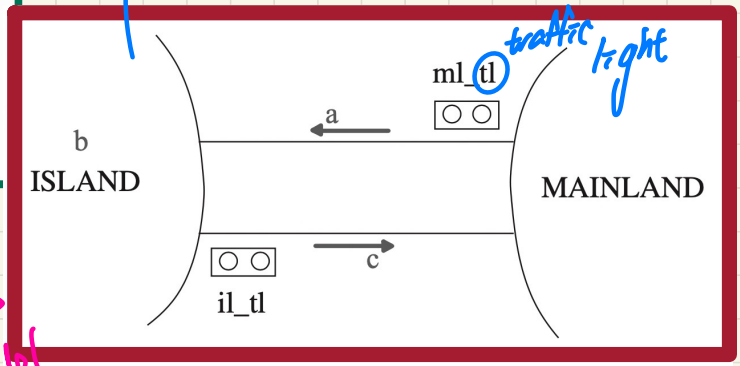
m1:

more concrete than m0, more abstract than m2



No to m1's abs. var. is not accessible by m1 ↓ event actions & guards.

superposition. abs. var. a, b, c are "inherited" and accessible by the con. model m2.



m2:

more concrete than m1

Bridge Controller: State Space of the 2nd Refinement

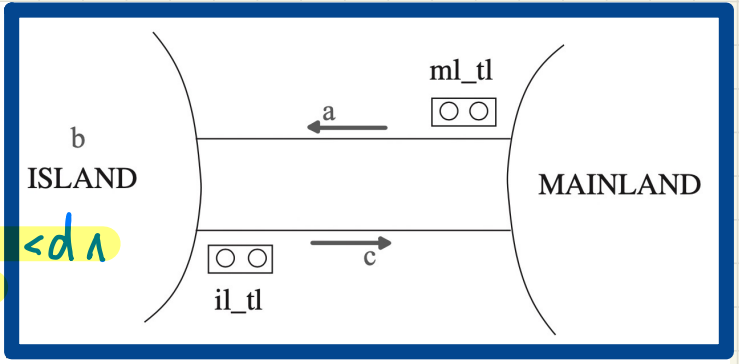
ENV1	The system is equipped with two traffic lights with two colors: green and red.
ENV2	The traffic lights control the entrance to the bridge at both ends of it.
ENV3	Cars are not supposed to pass on a red traffic light, only on a green one.

*Why not $a+b \leq d$?

Dynamic Part of Model

variables:
a, b, c
ml_tl
il_tl

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



Static Part of Model

sets: *COLOR*

constants: *red, green*

axioms:
axm2_1 : *COLOR* = {*green, red*}
axm2_2 : *green* ≠ *red*

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

Exam Info

- When: 7pm to 10pm, Sunday, December 15
- Where: TC Sobeys
- Coverage: **Everything** (lecture materials & labs)
  + slides, iPad notes
- Format: Mostly Written
  + explanations/justifications
  + write math expressions
  + calculations, proofs
- Restrictions:
  + **One-sided**, computer-typed, **min** 10pt data sheet
  + No sketch paper (Exam booklet includes it)
  + **No** calculator
- What you should bring:
  + Valid, Physical Photo ID (strict)
  + Water/Snack

ASCII vs. Math.
- should be computer-typed in the first instance
- ok to add some hand-written info in the last min.

↳ - question booklet
- answer booklet

Lecture 23 - Nov. 28

Bridge Controller

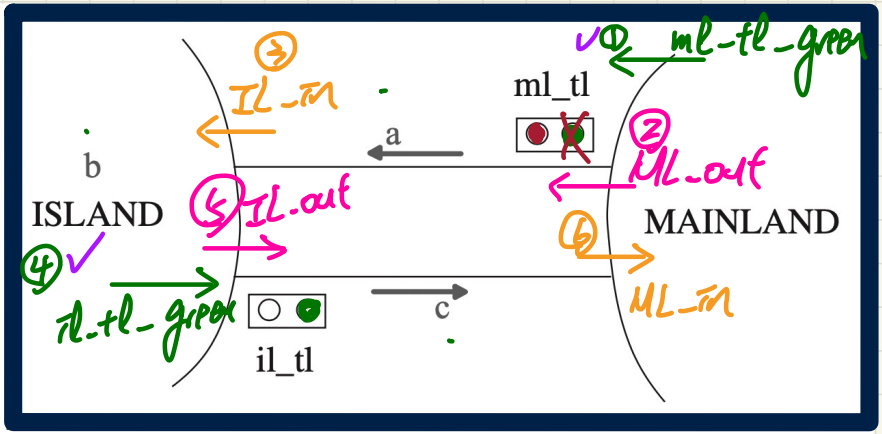
2nd Refinement: Splitting Guards
Adding Invariant to Prove INV

Announcements/Reminders

- **Lab5** released (due on Tuesday, December 3)
- **WrittenTest2** results released
- **Exam** review sessions polling
- Data Sheet:
  + Hand-Writing & Screenshots allowed
  + Font size requirement: ≥ 10pt

Bridge Controller: "Old" and "New" Events

Only ① and ④ require safety & capacity checks

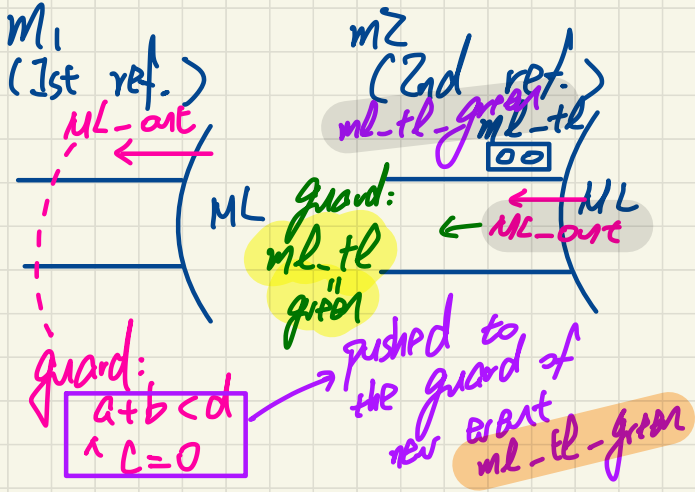


Single Car Travel:

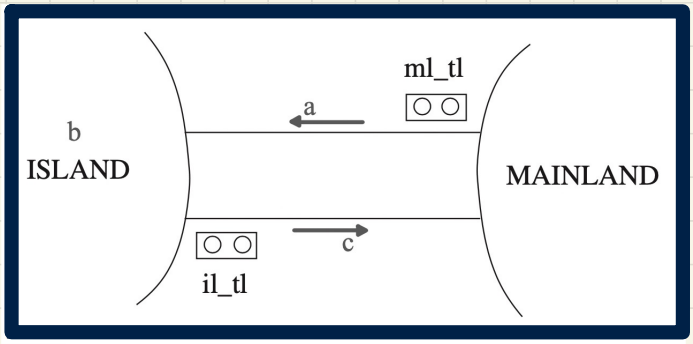
<init> → ml_tl = red.
il_tl

① ml_tl_green, ML_out,
② IL_in,
③ il_tl_green, IL_out, ML_in>

a, b, c are computer variables
↳ drivers should not access their values
↳ ML_out, IL_out) drivers should only be concerned about traffic lights colours.



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



sets: *COLOR*

constants: *red*, *green*

axioms:
axm2_1 : *COLOR* = {*green*, *red*}
axm2_2 : *green* ≠ *red*

variables:
a, *b*, *c*
ml_tl
il_tl

invariants:
inv2_1 : *ml_tl* ∈ *COLOUR*
inv2_2 : *il_tl* ∈ *COLOUR*
inv2_3 : *ml_tl* = *green* ⇒ *a* + *b* < *d* ∧ *c* = 0
inv2_4 : *il_tl* = *green* ⇒ *b* > 0 ∧ *a* = 0

in order for these two events not to be enabled at the same time, both tl's cannot be green

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

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

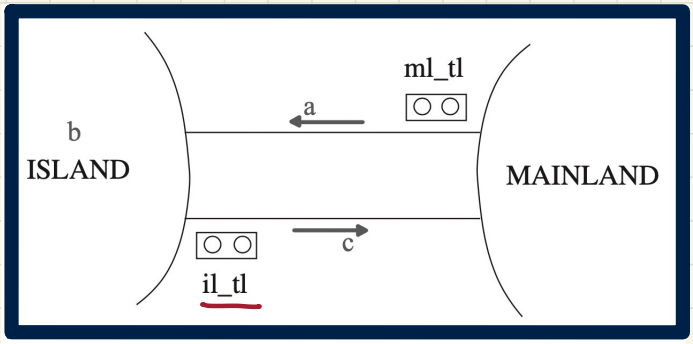
ml_tl = green

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

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

il_tl = green

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



sets: *COLOR*

constants: *red*, *green*

axioms:
axm2_1 : *COLOR* = {*green*, *red*}
axm2_2 : *green* ≠ *red*

variables:
a, *b*, *c*
ml_tl
il_tl

invariants:
inv2_1 : *ml_tl* ∈ *COLOUR*
inv2_2 : *il_tl* ∈ *COLOUR*
inv2_3 : *ml_tl* = *green* ⇒ *a* + *b* < *d* ∧ *c* = 0
inv2_4 : *il_tl* = *green* ⇒ *b* > 0 ∧ *a* = 0

ML_tl_green:

turn the traffic light ml_tl to green

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

abstract guard of ML_out in ml_1

$a + b < d$
$c = 0$

$ml\_tl = red$

IL_tl_green:

turn the traffic light il_tl to green

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

abstract guard of IL_out in ml_2

$b > 0$
$a = 0$

$il\_tl = red$

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

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

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

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

ML_out/inv2_4/INV

Concrete guards of *ML_out*

Concrete invariant **inv2_4** with *ML_out*'s effect in the post-state

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

Exercise: Specify IL_out/inv2_3/INV

Example Inference Rules

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

Modus Ponens

$(P \wedge (P \Rightarrow Q)) \Rightarrow Q$

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

$H \wedge P \Rightarrow Q$

$H \Rightarrow (P \Rightarrow Q)$

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

Contra-positive

$\neg P \Rightarrow Q$

$\equiv$ {Contra-pos}

$\neg Q \Rightarrow \neg(\neg P)$

Discharging POs of m2: Invariant Preservation

First Attempt

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

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

ML_out/inv2_4/INV

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

MON

Outstanding/Unprovable Sequent

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

$True \Rightarrow False \equiv False$) Contradiction.

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

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

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

IL_out/inv2_3/INV

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

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

MON

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

IMP_R

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

IMP_L

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

AND_L

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

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

exercise!

Fixed

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

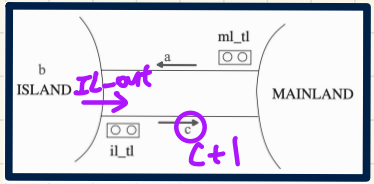
IL_out/inv2_3/INV

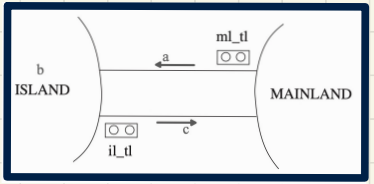
$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \vee c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
$il\_tl = green \Rightarrow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

$c=0 \wedge c=1 \equiv$ false
$c=0$ contradiction!

ML_out/inv2_4/INV

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





Unprovable Sequent:

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

⟨ init	, ML_tl_green	, ML_out	, IL_in	, IL_tl_green	, IL_out	, ML_out ⟩
$d = 2$ $a' = 0$ $b' = 0$ $c' = 0$ $ml\_tl' = red$ $il\_tl' = red$	$d = 2$ $a' = 0$ $b' = 0$ $c' = 0$ **ml_tl' = green** $il\_tl' = red$	$d = 2$ **a' = 1** $b' = 0$ $c' = 0$ $ml\_tl' = green$ $il\_tl' = red$	$d = 2$ **a' = 0** **b' = 1** $c' = 0$ $ml\_tl' = green$ $il\_tl' = red$	$d = 2$ $a' = 0$ $b' = 1$ $c' = 0$ $ml\_tl' = green$ **il_tl' = green**	$d = 2$ $a' = 0$ **b' = 0** **c' = 1** $ml\_tl' = green$ $il\_tl' = green$	$d = 2$ **a' = 1** $b' = 0$ $c' = 1$ $ml\_tl' = green$ $il\_tl' = green$

possible state

Fixing m2: Adding an Invariant

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

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

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

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

only different compared with the previous one for the change.

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*	$ml\_tl = green$
⊢	
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

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

MON

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

IMP_R

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

OR_L

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

EQ_LR, MON

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

Approach 1: NOT_L

Approach 2: green = red ⇒ false

exercise

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

IMP_L

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

AND_L

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

AND_R

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

HYP

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

EQ_LR, MON

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

ARI

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

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

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

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

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

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

Assignment

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

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

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

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

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

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

Lecture 24 - Dec. 3

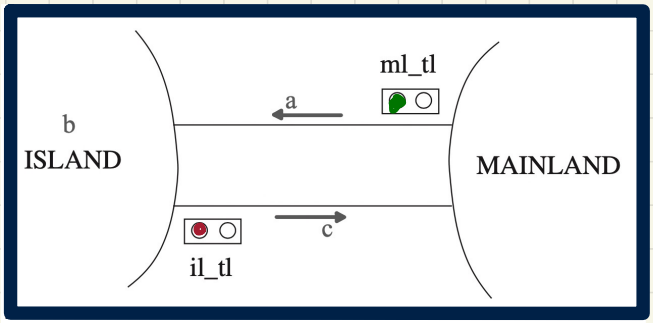
Bridge Controller

Adding Actions
Splitting Events
Preventing Livelock/Divergence
Proving Livelock/Divergence Freedom

Announcements/Reminders

- **Lab5** due today
- **Exam** review sessions and office hours TBA
- Sample exam questions to come
- Data Sheet:
  + Hand-Writing & Screenshots allowed
  + Font size requirement: ≥ 10pt

Fixing m2: Adding Actions



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

ml_tl' = g ∧ il_tl' = r

Added inv2_5 : | * b
new inv.
ML_arr ML_tl IL_arr IL_tl
ML_tl_green/inv2_5/INV
ml_tl = red ∨ il_tl = red

	axm0_1	$d \in \mathbb{N}$
	axm0_2	$d > 0$
	axm2_1	$COLOUR = \{green, red\}$
	axm2_2	$green \neq red$
m0	inv0_1	$n \in \mathbb{N}$
m0	inv0_2	$n \leq d$
	inv1_1	$a \in \mathbb{N}$
	inv1_2	$b \in \mathbb{N}$
	inv1_3	$c \in \mathbb{N}$
m1	inv1_4	$a + b + c = n$
m1	inv1_5	$a = 0 \vee c = 0$
	inv2_1	$ml\_tl \in COLOUR$
	inv2_2	$il\_tl \in COLOUR$
m2	inv2_3	$ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
m2	inv2_4	$il\_tl = green \Rightarrow b > 0 \wedge a = 0$
	inv2_5	$ml\_tl = red \vee il\_tl = red$

IL_tl = red

ML_tl_green
IL_tl_green.

Concrete grd. of ML_tl_gn.: ml_tl = red, a + b < d, c = 0

⊢ * green = red ∨ red = red.

green ml_tl' = red ∨ red il_tl' = red

Exercise: Specify IL_tl_green/inv2_5/INV

INV2_3 : ml_tl = g ⇒ a+b<d ∧ c=0

INV2_4 : tl_tl = g ⇒ b>0 ∧ a=0

Discussed (Thursday)

tl_tl = g ⇒ a=0

ML_out / INV2_4 / INV

IL_out / INV2_3 / INV

ml_tl = g ⇒ c=0

safety

To Discuss (Today)

ml_tl = g ⇒ a+b<d

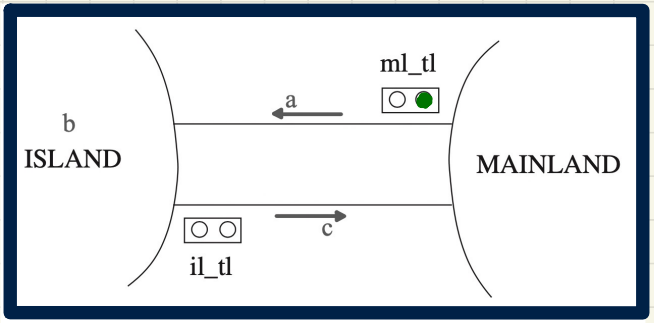
ML_out / INV2_3 / INV

IL_out / INV2_4 / INV

tl_tl = g ⇒ b>0

capacity

Invariant Preservation: ML_out/inv2_3/INV



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

$a' = a + 1$

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

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

Exercise: Specify IL_out/inv2_4/INV

$b' = b - 1$
$b - 1 > 0$

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*	$ml\_tl = green$
⊢	
Concrete invariant **inv2_3** with *ML_out*'s effect in the post-state	$ml\_tl = green \Rightarrow (a + 1) + b < d \wedge c = 0$

post-state of ML_out
$(a' = a + 1)$

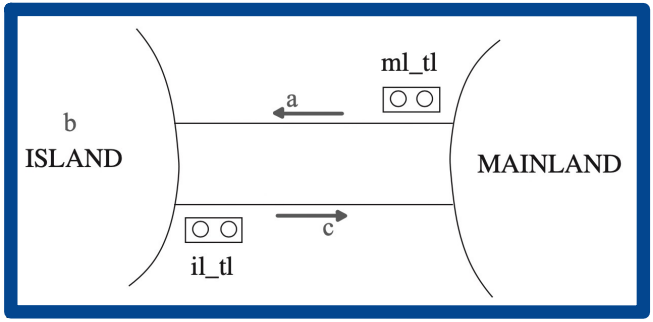
Discharging POs of m2: Invariant Preservation

First Attempt

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

MON

ML_out/inv2_3/INV



Unprovable

$a+b<d$
$c=0$
$ml\_tl = g.$
$\vdash (a+1)+b<d$

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

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

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

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

IMP_R

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

IMP_R ↓ IMP_L.

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

AND_L

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

AND_R

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

??

$a + b < d$
$c = 0$
$ml\_tl = green$
$\vdash$
$c = 0$

HYP

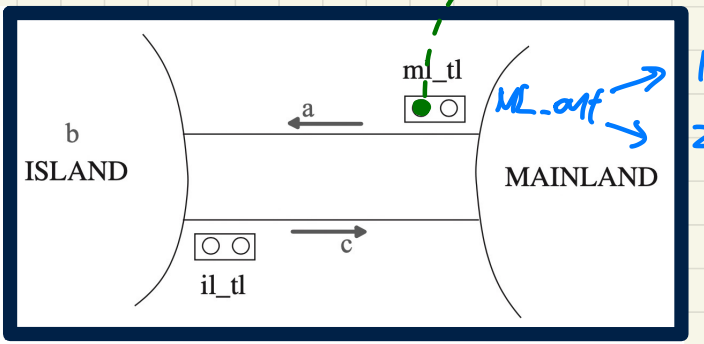
Understanding the Failed Proof on INV

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



Unprovable Sequent:

$$a + b < d$$
$$\wedge \quad c = 0$$
$$\wedge \quad ml\_tl = green$$
$$\vdash$$
$$(a + 1) + b < d$$

for $x+1<y$, $x < y \Rightarrow ? \; x+1 < y$ — not true in general

x can't be equal to $y-1$!

e.g. $3+1 \not< 4$

$d = 3, b = 0, a = 0$	$[(a+1)+b < d$ 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*** $]$

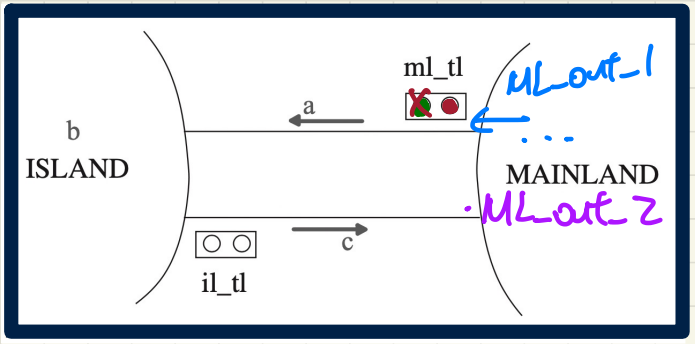
$(a+1)+b \neq d$ ↳ no need to turn ml_tl to red right away.

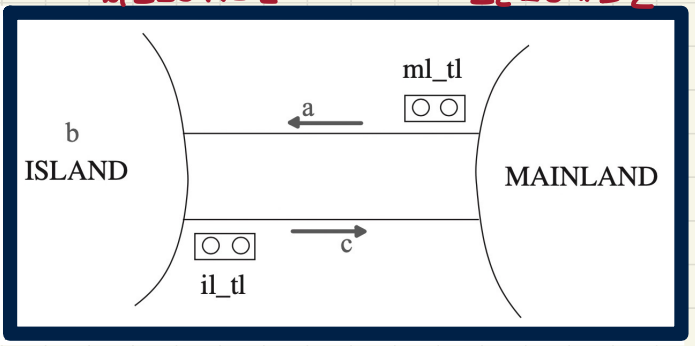
ML_ant allowing one now car on the bridge

$(a+1)+b = d$ need to turn ml_tl to red right away.

Fixing m2: Splitting Events

m1: ML_out
m2: ML_out_1, ML_out_2
IL_out
IL_out_1, IL_out_2





ML_out_1
when
ml_tl = green
a + b + 1 ≠ 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

disable ML_out_1

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

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

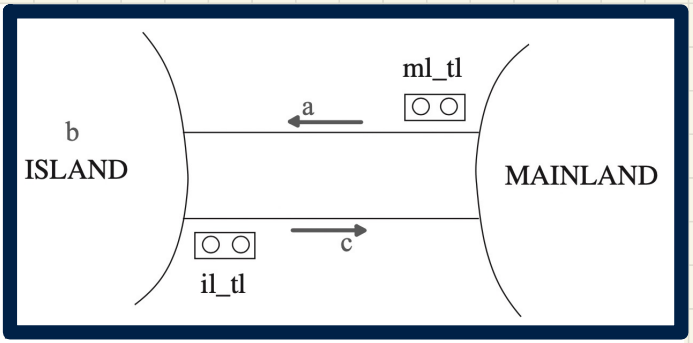
as soon as the capacity limit is reached, turn ml_tl to red

Current m2 May Livelock

infinite interleaving of new events

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



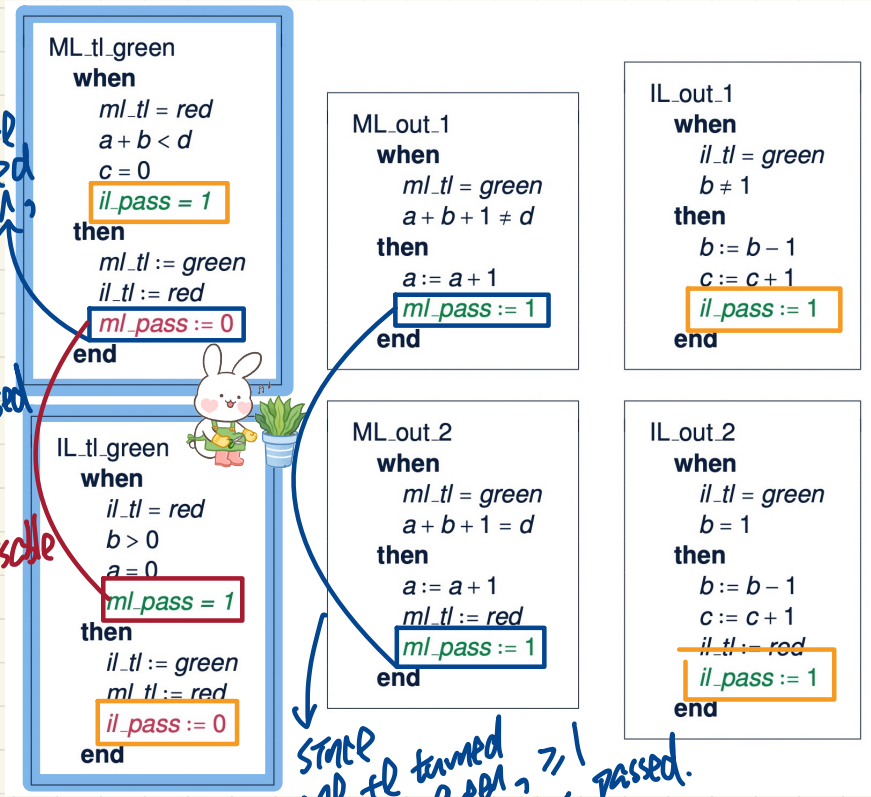
The current m2 diverges ∵ there's one valid trace of infinite interleaving of new events

starting point of livelock

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

Fixing m2: Regulating Traffic Light Changes

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



disabled IL_tl_green both new events enabled.

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
>		

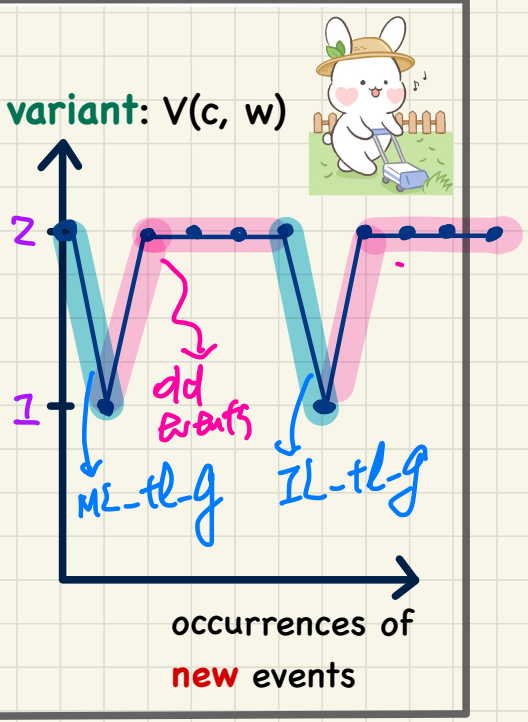
Fixing m2: Measuring Traffic Light Changes

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	2
IL_in,	1	1	2
IL_in,	1	1	2
IL_tl_green,	1	0	1
IL_out_1,	1	1	2
IL_out_2,	1	1	2
ML_in,	1	1	2
ML_in	1	1	2
>			

variants: *ml_pass + il_pass*



PO of Convergence/Non-Divergence/Livelock Freedom

A New Event Occurrence Decreases Variant

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

✓ post-state value of var.

pre-state value of var.

Variants: ml_pass + il_pass

ML_tl_green/VAR

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

$ml\_pass' = 0$
$\wedge\ tl\_pass' = tl\_pass$

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

$* \quad ml\_pass' + tl\_pass' < ml\_pass + tl\_pass$

PO of Relative Deadlock Freedom

Abstract m1

variables: a, b, c

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

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

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

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

IS#1

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

IS#2

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

IS#3

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

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

Abstract 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

Concrete m2

constants: d

sets: $COLOR$

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

variables:
a
b
c
ml_tl
il_tl
ml_pass
il_pass

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** : $il\_tl = green \Rightarrow b > 0 \wedge a = 0$
- **inv2_5** : $ml\_tl = red \vee il\_tl = red$
- **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: $ml\_pass + il\_pass$

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

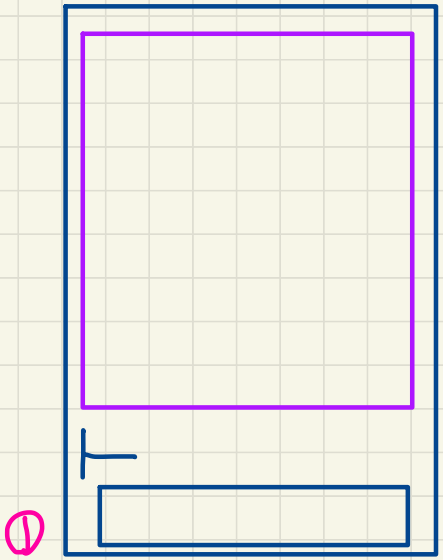
IL_out_2
when
$il\_tl = green$
$b = 1$
then
$b := b - 1$
$c := c + 1$
$il\_tl := red$
$il\_pass := 1$
end

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

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

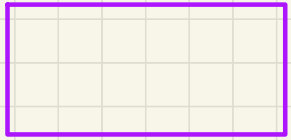
Review Q & A - Dec. 13

Exam Review Q&A



② select (or sketch) the predicated related to the variables

MON



① Examine what's to be proved

(1) structure ($\Rightarrow$, $>$, $=$)

(2) variables involved

③ From there, see how these hypotheses are related to the goal.

1. data sheet
2. inference rules applicable

$P \equiv \neg(\neg P)$

Discharging POs of m2: Invariant Preservation

Second Attempt

ML_out/inv2_4/INV

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

MON

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

IMP_R

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

IMP_L

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

AND_L

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

AND_R

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

OR_L

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

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

EQ_LR, MON

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

HYP

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

EQ_LR, MON

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

ARI

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

exercise

$\neg(green = red)$

Approach 1: NOT_L

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

Approach 2: green = red 也false

NOT_L
$green = red$ (F)
$tl\_tl = green$
$1 \neq 0$ (T)
$\vdash$
$green = red$ (F)

ARI

Contra-positive.
$\neg P \Rightarrow Q \equiv \neg Q \Rightarrow P$

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

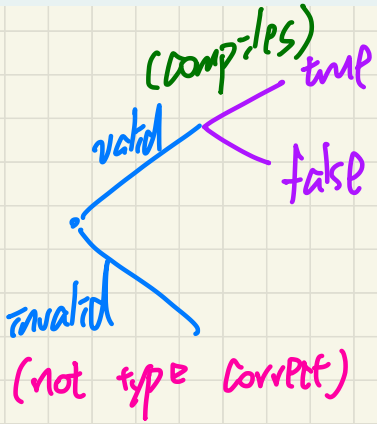
$$\frac{H(F), E = F \vdash P(F)}{H(E), E = F \vdash P(E)}\ \text{EQ\_LR}$$

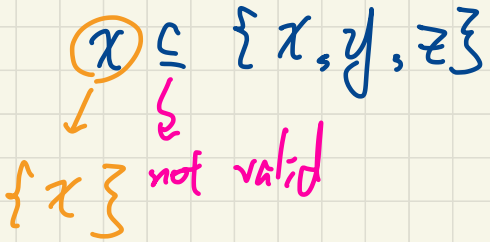
$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, P \vee Q \vdash R}\ \text{OR\_L}$$

{} : {a, b, c, d} +->> {23, 46, 69}	Choose...
{(a, 23), (b, 46), (c, 69), (d, 81)} <: {a, b, c, d} >+> {23, 46, 69, 81, 98, 101}	Choose...
{} : {a, b, c, d} >+> {23, 46, 69}	Choose...
{(1, a), (2, b), (3, c), (4, a)} /: {1, 2, 3, 4} >->> {a, b, c}	Choose...
{} <: {a, b, c, d} >-> {23, 46, 69}	Choose...

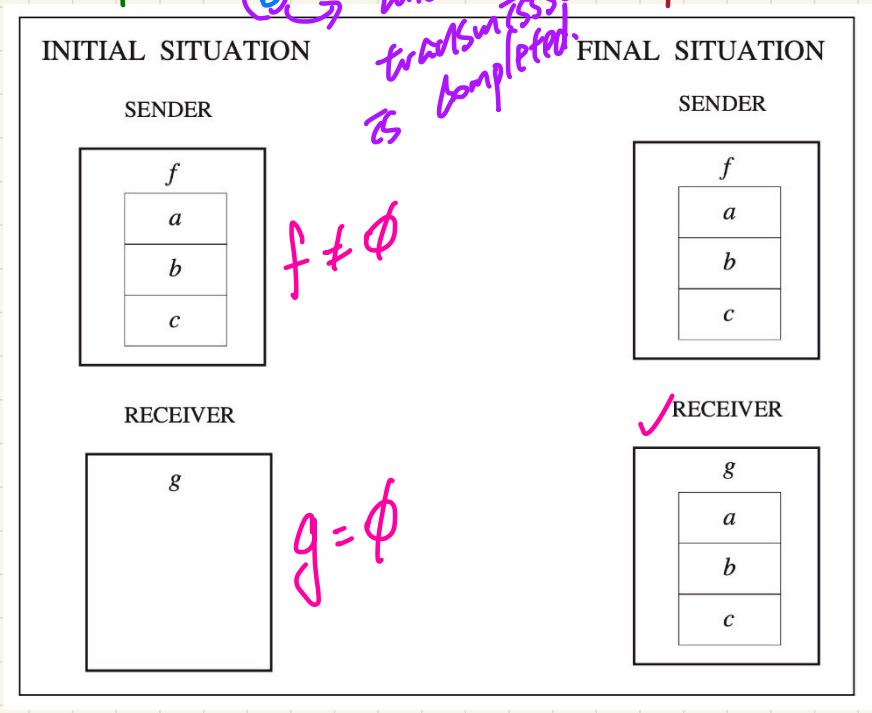
set of all possible partial injections

subset





pre-trans. b = false whether or not transmission is completed. post-trans. b = true.



$f \neq \phi$

$g = \phi$

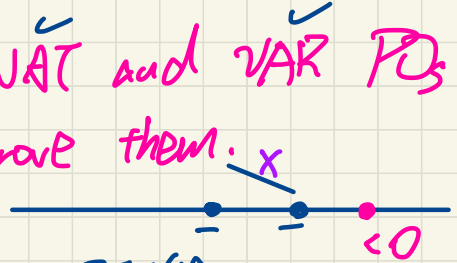
$f' = f$

$g' = f$

- Find a trace to prove some given variant?
  - even if this witness shows that the NAT and VAR properties are satisfied, it's not sufficient.

$$\forall \text{ traces} \cdot \text{NAT} \wedge \text{VAR}$$

To prove a given variant, state NAT and VAR POs and prove them.



- To disprove a variant being valid/appropriate, find a witness trace which violates either NAT or VAR.

Assume a model consisting of the following components (where A1, I1, G1, and G2 are some valid predicates referring to the declared constants and/or variables):

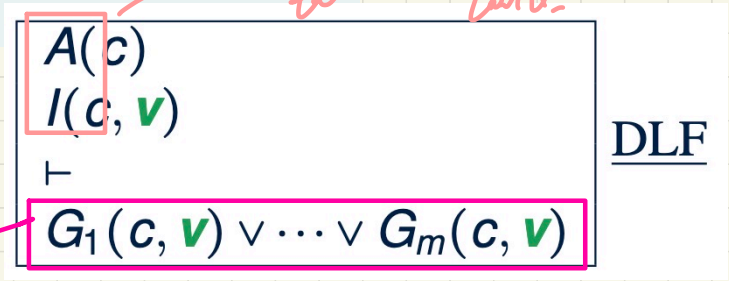
- An axiom: A1
- An invariant: I1
- An event e1 with guard: G1
- An event e2 with guard: G2

Before the change: DLF condition is $G_1 \vee G_2$

After the change: $(G_1 \wedge P) \vee G_2$

Consider each of the following 9 possible changes, introduced to the above model, **in isolation**:

1. Adding a new axiom A2 (where A2 is a valid predicate)
2. Changing event e1's guard to "G1 & P" (where P is some valid predicate)
3. Changing event e1's guard to "G1 or P" (where P is some valid predicate)
4. Removing axiom A1
5. Removing e2's guard G2 (so that it has no guard)
6. Adding a new, second guard G2' (where G2' is a valid predicate) to event e2
7. Adding a new invariant I2 (where I2 is a valid predicate)
8. Adding a new event e3 with guard G3 (where G3 is a valid predicate)
9. Removing invariant I1



I hope you enjoyed learning with me 🤗

All the best to you !