EECS3342 System Specification and Refinement

Lecture Notes

Fall 2025

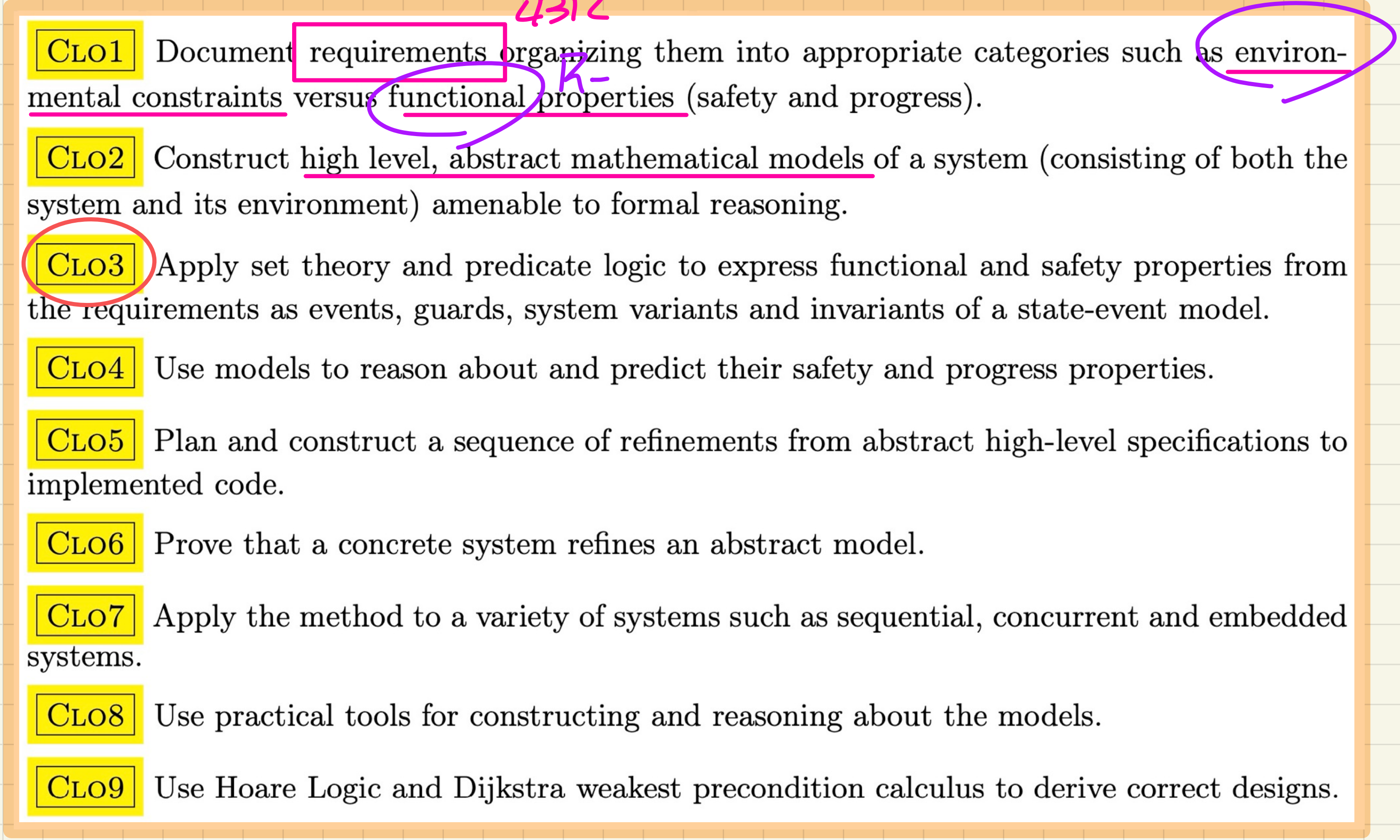
Jackie Wang

Lecture 1 - Sep. 4

Syllabus

Course Learning Outcomes (CLOs)

E -



4312

R-

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.

⇗

$\longrightarrow$ total funct.

$\not\longrightarrow$ partial funct.

Lecture 1 - Sep. 9

Syllabus & Introduction

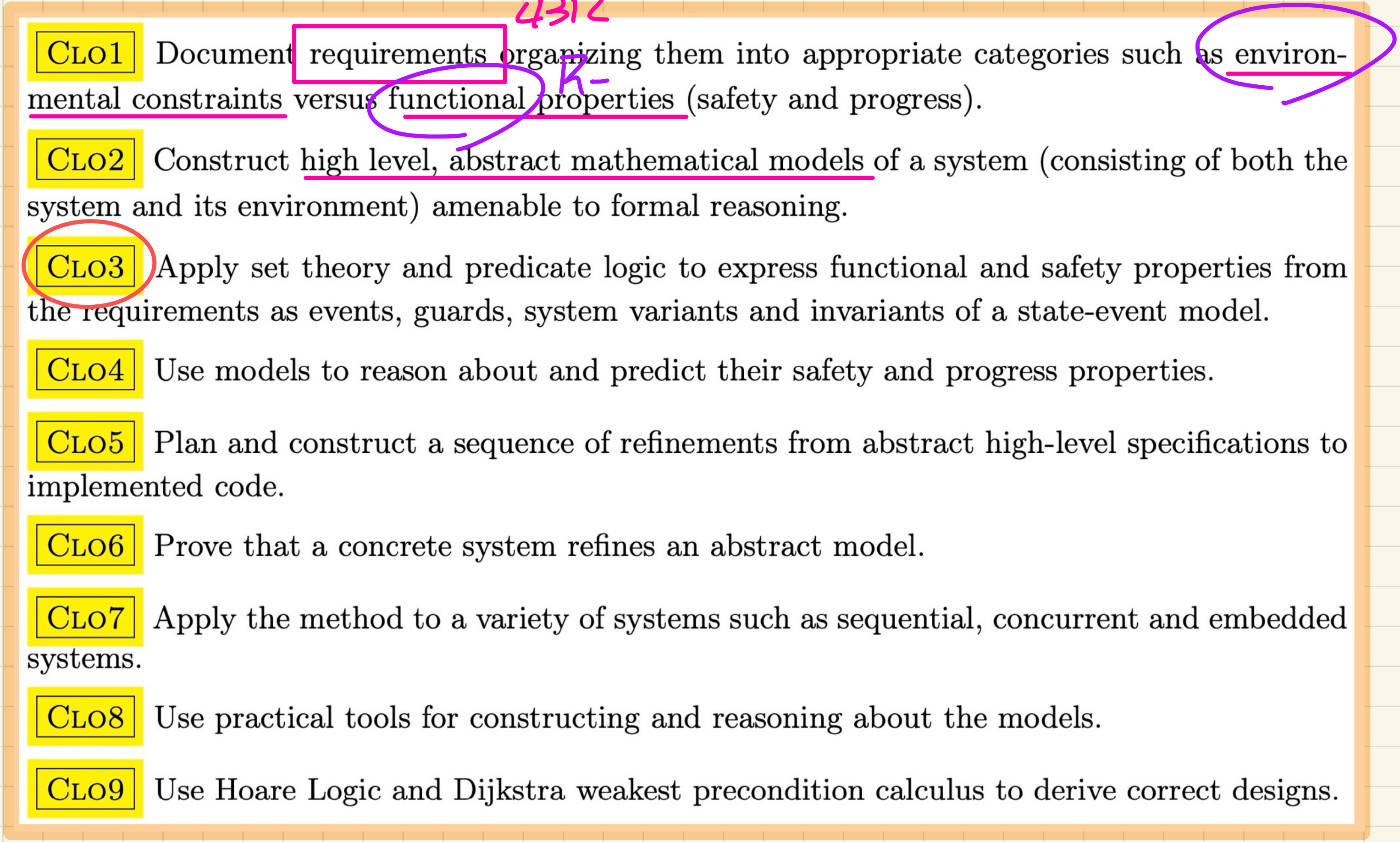
Formal Methods:
Theorem Proving vs. Model Checking

Course Learning Outcomes (CLOs)

E -

4312

R-



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.

Lecture 3 - Sep. 11

Math Review

Propositions: Commutativity vs. SCE
Implications: Contracts, Theorems
Predicates: Universal vs. Existential Q.

Announcements/Reminders

REQ8

transfer
↳ same var in two actions

- First Class (Syllabus) recording & notes posted
- Today's class: **notes template** posted
- **Event-B Summary** Document
  ↳ print copy allowed?
- **Priorities**:
  + **Lab1** → **Due**: Next Tuesday (Sep 16)
  + **Lab2** → **Due**: Tuesday (Sep 23)
- Missed Lecture 2 (Tuesday):
  + We'll dive directly into **Math Review** (**1b**).
  + **Introduction** (**1a**) will come after the review is done.

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

Exercise
SCE for ||

short circuit

p && q
p q
↳ evaluate from L to R
↳ if L evaluates to false, bypass R.

Q. Are the ∧ and ∨ operators equivalent to, respectively, && and || in Java?

well-definedness (2) a[i] ↳ $i \in 0..a.length-1$

(1) $x/y \rightarrow y \neq 0$

$p \wedge q$

① "evaluated" separately & independently

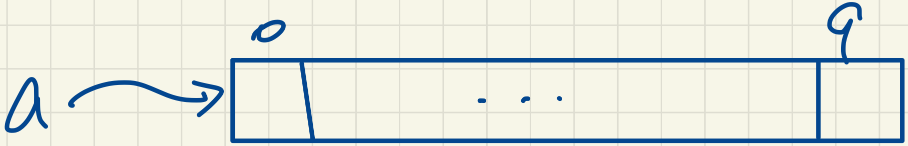
② both p and q well-defined.

Commutativity

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

Java p && q q && p

Accessing Array



$a.length == 10$

(1) $\overset{①}{\underline{i < a.length}}$ && $\overset{②}{\underline{a[i] > 10}}$ && $\overset{③}{\underline{i >= 0}}$

say $i == -2$

① $-2 < 10$ Ⓣ

② $a[-2] > 10 \rightsquigarrow$ AIOOBE (fail)!

say $i == 100$

① $100 < 10 \rightsquigarrow$ false

↳ bypass ②, ③

guarding condition

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

exercise

guarding cond. for ||

implication

$\Rightarrow$ $\quad$ $\not\rightarrow$

$\Leftrightarrow$ $\quad$ $\not\leftrightarrow$

alt

guard_1 : ______

guard_2 : ______

⋮

guard_1

∧

guard_2

∧

⋮

Implication ≈ Whether a Contract is Honoured

DbC

$p \Rightarrow q$

antecedent → promised terms

consequence → duties

① $T \Rightarrow T \equiv T$

② $T \Rightarrow F \equiv F$ → only case the contract is violated.

③ $F \Rightarrow T \equiv T$

④ $F \Rightarrow F \equiv T$

→ If not paid, work or not will not violate the contract

③, ④

zero of $\Rightarrow$: $F \Rightarrow p \equiv T$

false proposition true

①, ②

identity of $\Rightarrow$: $T \Rightarrow p \equiv p$

true

$p \Rightarrow q$

def. of $\Rightarrow$: $p \Rightarrow q \equiv$
$\neg p \vee q$

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

(2) Converse : $q \Rightarrow p$

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

(Inverse of Converse)

$$\boxed{p \Rightarrow q \equiv \neg q \Rightarrow \neg p}$$

Most General

$$\boxed{\begin{array}{l} \forall x \cdot Q(x) \\ \\ \exists x \cdot Q(x) \end{array}}$$

Tool of spec. & proofs (e.g. Rodin)

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

$$\exists i \cdot R(i) \land P(i)$$

Predicate Logic: Quantifiers

- syntax
- base cases in programming

$$\forall i \bullet R(i) \Rightarrow P(i)$$

universal quantification

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

(any i not satisfying R not considered)

$$\exists i \bullet R(i) \land P(i)$$

existential quantification

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

range (universe of disclosure)

property to be asserted

Q. What happens if $R(i)$ corresponds to an empty range.

$R(i) \equiv False$?

Predicate Logic: Quantifiers

- syntax
- base cases in programming

zero of ⇒: $F \Rightarrow P \equiv T$.

$$\forall i \bullet \overset{\text{false}}{\boxed{R(i)}} \Rightarrow P(i)$$

what if empty range?

$$\exists i \bullet \overset{\text{false}}{\boxed{R(i)}} \land P(i)$$

what if $R(i) \equiv$ false?

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

∵ no witness can be found in empty range to prove otherwise non-positive.

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

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

Lecture 4 - Sep 16

Math Review

Implication: Alternative Exp. in English
Logical Quantifiers: Proof Strategies
Sets: Enumerations vs. Comprehension

Announcements/Reminders

- Today's class: notes template posted
- **Event-B Summary** Document
- **Priorities**:
  - + **Lab1** → **Due**: This Tuesday (Sep 16)
  - + **Lab2** → **Due**: Next Tuesday (Sep 23)

given p (ante.) is true, the only way to make $p \Rightarrow q$ true is when q is also true.

$\boxed{p \Rightarrow q}$

$p \Rightarrow q$

sufficient

necessary

p: snow storm

q: cancel class

Expressing Implications

q if p, p is sufficient for q

c1 **q unless ¬p** c2

$p \Rightarrow True \equiv True$

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

if p

if ¬p, then don't care what q is

neither c1 ¬q

nor c2 $\neg(\neg p)$ = p

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

c1: q

c2: ¬p

p only if q, q is necessary for p

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

p.

¬q

¬p

↳ don't care about q.

e.g. Formulate "$\underset{q}{x > 0}$ if $\underset{p}{y \leq 10}$"

↳ $y \leq 10 \Rightarrow x > 0$

$p \overset{if}{\underset{only\ if}{\Longleftrightarrow}} q$

1. p if q ($p \Leftarrow q$, $q \Rightarrow p$)
2. p only if q ($p \Rightarrow q$)

$\mathbb{Z}$ set of integers

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

$\mathbb{N}$ set of natural numbers

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

$\mathbb{N}_1$ $1, 2, \cdots, +\infty$ (positive integers).

3 numbers

$1..3$

8 numbers

$4..11$

$$\forall i, j \bullet i \in \boxed{\mathbb{N}} \land j \in \boxed{\mathbb{Z}} \Rightarrow \underline{P(i, j)}$$

need to consider all combinations of (i, j)

need to consider

$3 * 8 = 24$

combinations of i, j.

Logical Quantifiers: Examples

$\forall i \bullet \boxed{i \in \mathbb{N}} \Rightarrow i \geq 0$ $\quad \textcircled{T}$

$i \in \{0, 1, 2, \dots\}$

F $\textcircled{\forall} i \bullet \underline{i \in \mathbb{Z} \Rightarrow i \geq 0}$

witness : -1

$\boxed{-1 \in \mathbb{Z}} \Rightarrow \boxed{-1 \geq 0}$

T $\quad \textcircled{F} \quad$ F

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

witness : choose i, j s.t. $i = j$.

T $\textcircled{\exists} i \bullet i \in \mathbb{N} \wedge i \geq 0$ $\quad$ witness: 0

$0 \in \mathbb{N} \wedge 0 \geq 0 \equiv \textcircled{T}$.

T $\textcircled{\exists} i \bullet i \in \mathbb{Z} \wedge i \geq 0$ $\quad$ witness: 0

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

witness:

$i = 2$

$j = 3$.

$R(i) \triangleq$ student i enrolled in 3342
$P(i) \triangleq$ student i received A+

$\triangleq$ is defined as.

Logical Quantifiers: Examples

Goal: show $R(i) \Rightarrow P(i) \equiv$ True.

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

trivial ← (1) show $\neg R(i)$ ∵ zero of $\Rightarrow$: $false \Rightarrow p \equiv true$

harder ← (2) show $R(i), P(i)$ (every i sat R also sat P)

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

Goal: show $R(i) \land P(i) \equiv$ True.

(1) give a witness i that sat both R and P.

Goal: show $R(i) \Rightarrow P(i) \equiv$ False.

How to **disprove** $\forall i \bullet \underset{T}{R(i)} \Rightarrow \underset{F}{P(i)}$?

(1) give a witness j s.t. $R(j)$ but $\neg P(j)$.

How to **disprove** $\exists i \bullet R(i) \land P(i)$? harder.

Goal: show $R(i) \land P(i) \equiv$ False.

(1) show $\neg R(i)$: $false \land p \equiv false$

(2) $R(i), \neg P(i)$ every i sat R does not sat P.

Prove/Disprove Logical Quantifications

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

↳ $x \in 1..10$ → each member in this interval is > 0.

($R(x)$ is not false)

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

Exers

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

$x \in 1..10$

witness : $\exists$

Disprove using witness $x=1$

$1 \in \mathbb{Z} \wedge 1 \leq 1 \leq 10 \wedge 1 > 1$ (F)

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

Exercise

wrong statements to disprove $\exists$, not sufficient with a single witness.

Sets: Definitions and Membership

No ordering: $\underline{\{1, 2, 3\}} = \{2, 3, 1\}$

set enumeration (members are explicit).

No duplicates: $\{1, \underline{\underline{2}}, 3, \underline{\underline{2}}\}$ ×

① $\{\underline{x} \mid x \in \mathbb{N} \land 0 \le x \le 2\}$
$= \{0, 1, 2\}$

② $\{2 \cdot x \mid x \in \mathbb{N} \land 0 \le x \le 2\}$
$= \{0, 2, 4\}$

Set Comprehension

$\{$ expression: form of member in the final set $\mid$ Constraint: values sat this can be included in the set $\}$

Constraint: implicit about what set members are

v1, v2, v3

③ $\left|\{(x, y) \mid x \in \mathbb{N} \land y \in \mathbb{N} \land 1 \le x \le 2 \land 3 \le y \le 4\}\right| = 4$

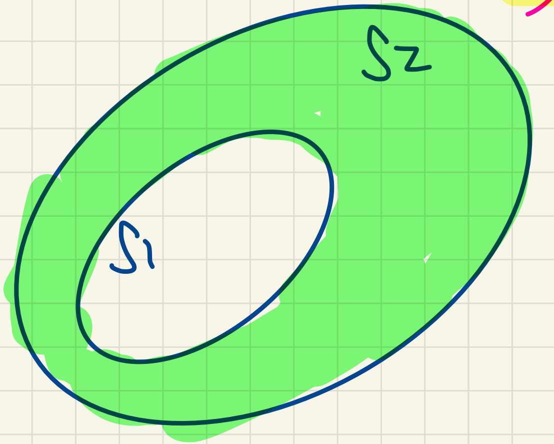
$\{(1,3), (1,4), (2,3), (2,4)\}$

Q. How many sets of size 3 can you make out of 1, 2, 3, 4, 5

Relating Sets

subset

$S1 \subseteq S2$



$S_2 \setminus S_1 = 0$

$\Rightarrow S_1 = S_2$

$|\underbrace{S_2 \setminus S_1}| > 0$

set diff

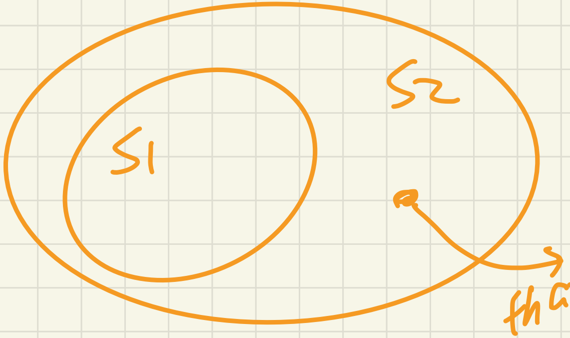
all members in S2 but not in S1

Proper subset

$S_1 \subset S_2$

$\iff S_1 \subseteq S_2 \wedge S_1 \neq S_2$

$\iff S_1 \subseteq S_2 \wedge (\exists x \cdot x \in S_2 \wedge x \notin S)$



witness that is $\in S_2$ $\notin S1$

Lecture 5 - Sep 18

Math Review

Converting ∀ and ∃ : Equational Proofs

Understanding the Choose Operator

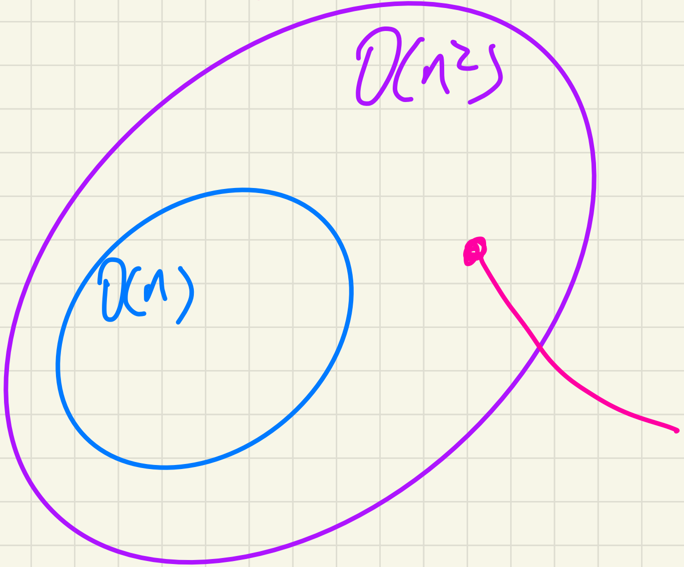
Power Sets

Announcements/Reminders

For next class
precedence of $\land$ vs. $\forall$

- Today's class: notes template posted
- **Event-B Summary** Document
- **Priorities**:
  - + **Lab1** → **Due**: This Tuesday (Sep 16)
  - + **Lab2** → **Due**: Next Tuesday (Sep 23)
- To be released:
  - + **ProgTest** guide
  - + 2 Practice Tests
  - + **Lab1** solution

$O(n) \subset O(n^2)$



e.g. $2n^2 + 3n - 4$

$n \cdot \log n$

$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Logical Quantifications: Conversions

R(x): x ∈ 3342_class

P(x): x receives A+

Axiom: $\forall x \bullet Q(x) \iff \neg \exists x \cdot \neg Q(x)$

$$(\forall\ X \bullet R(X) \Rightarrow P(X)) \Leftrightarrow \neg(\exists\ X \bullet \overset{R(x)}{R} \wedge \neg \overset{P(x)}{P}\ )$$

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

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

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

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

$$\neg \exists x \cdot \boxed{\neg(\neg R(x) \vee P(x))}$$

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

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

$$\iff \{ \neg(\neg p) \equiv p \}$$

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

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

↓

Exercise

✓

De Morgan

Relating Sets: Exercises

$m \in S$

$\neg(m \in S) \Leftrightarrow m \notin S$

$S_1 \subseteq S_2 \land S_2 \subseteq S_1 \Leftrightarrow S_1 = S_2$

$\underline{\{1,2\} \subseteq \{1,2,3\}}$

<u>but they're not equal</u>

$S \subset S$ <u>always fails</u>

↳ not-empty: $\{1,2\} \subset \{1,2\}$ ✗

↳ empty: $\emptyset \subset \emptyset$ $\underset{0}{\underline{|\emptyset|}} < \underset{0}{\underline{|\emptyset|}}$ ✗

$\emptyset \subset \underline{\underline{S}}$ sometimes holds, sometimes fails

↳ S empty →

?

↳ S not empty →

Sets: Exercises

$S_1 \setminus S_2 \rightarrow$ members in S_1 but not in $S_2 \rightarrow \neg(e \in S)$

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

Find a common pattern for defining:

1. = (numerical equality) via ≤ and ≥
2. = (set equality) via ⊆ and ⊇

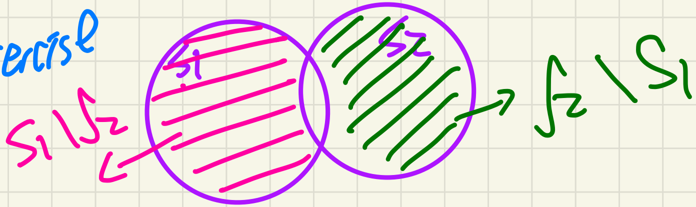
$x = y \Leftrightarrow x \leq y \land y \leq x$

$S_1 = S_2 \Leftrightarrow S_1 \subseteq S_2 \land S_2 \subseteq S_1$

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

	S	T	U
S	⊆ ⊂	⊆ ⊂	⊆ ⊂
T	⊆ ⊂	⊆ ⊂	⊆ ⊂
U	⊆ ⊂	⊆ ⊂	⊆ ⊂

Exercise



$S_1 \setminus S_2 \overset{?}{=} S_2 \setminus S_1$

① Give some witness of violation.

✓

Is set difference $\boxed{(\setminus)}$ commutative?

Exercise:

How many **sets** of size 3 can be made out of values 1, 2, 3, 4, 5?

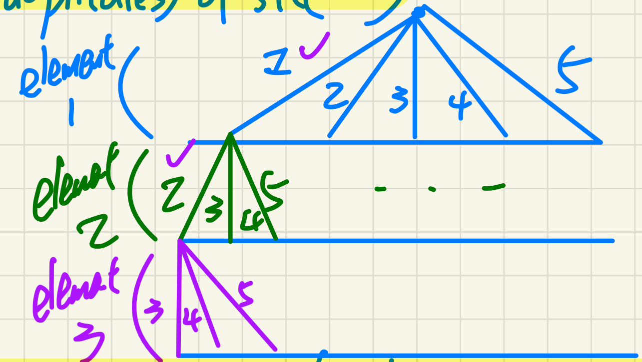
5.

$\left(\begin{matrix}5\\3\end{matrix}\right)$

(Step 1) Make sequences (with no duplicates) of size 3

$\boxed{\underline{5} \times \underline{4} \times \underline{3}}$

\# sequences



(Step 2) Disregard ordering of sequences with the same set of contents

3!

For {1, 3, 5}, we would've made sequences:

1 3 5
1 5 3
3 1 5
3 5 1
5 1 3
5 3 1

$3 * 2 * 1$

$= \boxed{3!}$ sequences that correspond to the same set.

(Step 3)

\# sets of size 3

$$\frac{5 * 4 * 3}{3!}$$

disregarding ordering.

$\binom{n}{r}$ → n choose r $(n \geq r, r \geq 0)$

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

out of n given elements,
how many ways to make a set
of card./size of r?

$$\binom{n}{r} = \binom{n}{n-r}$$

=

$$\frac{n!}{(n-r)! \cdot \underline{\underline{r!}}}$$

=

r terms

$$\frac{\boxed{n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot (n-r+1)}}{r!}$$

$[n, n-r+1]$
=
(r).

$$\binom{10}{8} = \frac{10 * 9 * 8 * 7 * 6 * 5 * 4 * 3}{8!}$$

=

$$\binom{10}{2} = \frac{10 * 9}{2!}$$

Power Set power set of S $\boxed{\mathbb{P}(S)} = \{ x \mid x \subseteq S \}$

Each member in $\mathbb{P}(S)$ is a set

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

$\mathbb{P}(\{1,2,3\}) = \{ x \mid x \subseteq \{1,2,3\} \}$

$= \{ \emptyset,$ smallest $\binom{3}{0} = 1$ (subset of card. 0)

$\{1\}, \{2\}, \{3\},$ $\binom{3}{1} = 3$ (subsets of card 1)

$\{2,3\}, \{1,3\}, \{1,2\},$ $\binom{3}{2} = 3$ (subsets of card 2)

$\{1,2,3\} \}$ largest $\binom{3}{3} = 1$ (subset of card 3)

1. smallest member in $\mathbb{P}(S)$: $\emptyset$

   $\emptyset \subseteq S$

2. largest member in $\mathbb{P}(S)$: S

   $S \subseteq S$

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

$$|\mathbb{P}(\{1,2,3\})| = \binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3}$$

Set of Tuples

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.

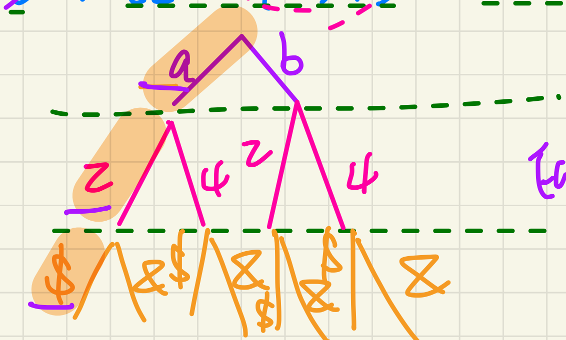
n sets.

$$\boxed{S_1 \times S_2 \times \cdots \times S_n} = \{(\overbrace{e_1, e_2, \ldots, e_n}^{n \text{ elements in the tuple.}}) \mid e_i \in S_i \wedge 1 \leq i \leq n\}$$

Example: Calculate $\{\overset{e1}{a, b}\} \times \{\overset{e2}{2, 4}\} \times \{\overset{e3}{\$, \&}\}$

$$= \{ (e1, e2, e3) \mid e1 \in \{a, b\} \wedge e2 \in \{2, 4\} \wedge e3 \in \{\$, \&\} \}$$

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



Each root-to-leaf path corresponds to a ~~n~~ 3-tuple

Relation : set of ordered $\overset{\text{tuples.}}{\text{pairs}}$

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

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

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

Lecture 6 - Sep 23

Math Review

Constructing All Relations
Domain, Range, Inverse
Image, Restrictions, Subtractions

Announcements/Reminders

- Today's class: notes template posted
- Event-B Summary Document
- Priorities:
  - + Lab1 → Review
  - + Lab2 → Due: This Tuesday (Sep 23)
- Released:
  - + ProgTest guide
  - + 2 Practice Tests
  - + Lab1 solution

Cardinality of Power Set: Interpreting Formula

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

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

$$|\mathbb{P}(S)| = \underbrace{\overset{1}{\binom{|S|}{0}}}_{\emptyset} + \binom{|S|}{1} + \binom{|S|}{2} + \cdots + \underbrace{\overset{1}{\binom{|S|}{|S|}}}_{S}$$

$$|\mathbb{P}(S)| = \underbrace{2 \times 2 \times \cdots \times 2}_{|S|} = 2^{|S|}$$

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

$2^{|S|} = 2^{③} = 8$

exclude c

include c

$$\left(\begin{array}{ccc} 0 & 0 & \textcircled{0} \\ 0 & 0 & \textcircled{1} \\ 0 & 1 & 0 \\ & \vdots & \\ 1 & 1 & 1 \end{array} \right. \begin{array}{l} \longrightarrow \emptyset \\ \longrightarrow \{c\} \\ \\ \\ \longrightarrow \{a, b, c\} \end{array}$$

relation on S and S

acceptable.

e.g. id

tuples.

Relation : set of ordered pairs

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

$\{(x,y) \mid x \in S \land y \in T\}$

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

  No! ∵ $(1,a)$ is not a set.

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

- Is $\{(\overset{\notin S}{b}, \overset{\notin T}{2})\}$ a relation? No. order is wrong!

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

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

$R_1 = R_2$

What is the min relation on S and T? $\emptyset$

What is the max relation on S and T? $S \times T$

Set of Possible Relations

subsets of max relation.

max relation

- **Set** of possible **relations** on S and T: $\mathbb{P}(S \times T)$
- 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: $r \in \mathbb{P}(S \times T)$ $r \in S \leftrightarrow T$

S T

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

Hint: How many? $2^{|S\times T|} = 2^{2\times 2} = 2^4 = 16$ max rel: $\{(a,2), (a,4), (b,2), (b,4)\}$ $S \times T$

$|S \times T| = 4$

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

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

→ relations of card 2 $\binom{4}{2}$

→ relations of card. 3 $\binom{4}{3}$

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

$* \{ r \mid r \in Dep. \leftrightarrow Des \wedge |r| = 2 \}$

Exercise

$$Departure = \overset{3}{\{} toronto, montreal, vancouver \}$$

$$Destination = \overset{3}{\{} beijing, seoul, penang \}$$

$$\underbrace{airline}_{\text{a single relation}} \in \boxed{Departure \leftrightarrow Destination} \quad \mathbb{P}(Dep. \times Des.)$$

$$\binom{\boxed{|Dep. \times Des|}^{9}}{2} = 36 \quad (1) \quad |Dep. \leftrightarrow Des.| = 2^{|Dep \times Des|} = 2^{3 \times 3} = 2^{9} = \underline{\underline{512}}$$

(2) enumerate $*$ $\boxed{\text{those relations with card 2.}}$

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

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

Relational Operations: Domain, Range, Inverse

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

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

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

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

$r^{\sim}$

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

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

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

$r^{\sim}$

$$\left.\begin{array}{l} dom(r) = ran(r^{\sim}) \\ ran(r) = dom(r^{\sim}) \end{array}\right)$$ algebraic properties.

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

$r[\{a, h\}] = \underbrace{r[\{a\}]}_{\{1, 4\}} \cup \underbrace{r[\{h\}]}_{\emptyset}$

Relational Operations: Image

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

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

$S \subseteq \text{Alphabet}$

$S \subseteq \text{dom}(r)$ × not necessary.

$r[\underbrace{\{g\}}_{\subseteq \text{Alphabet}}] = \emptyset$ no value mapped from g in r.

Exercises

- **Image** of {a, b} on r?
- **Image** of {1, 2} on r? $r[\overset{\not\subseteq \text{Alphabet}}{\{1, 2\}}]$ 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**. $r[\text{dom}(r)] = \text{ran}(r)$
- Calculate r's **domain** via an **image**. $r^\sim[\text{ran}(r)] = \text{dom}(r)$

$\text{ran}(r) \rightarrow \text{dom}(r^\sim)$

$r \in S \longleftrightarrow T$

$s \subseteq S \qquad t \subseteq T$

a new relation.

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

	domain	range
Restriction	* $s \lhd r$	$r \rhd t$
Subtraction	$s ⩤ r$	$r ⩥ t$

Each of these operators returns a new relation.

$S = \{a, b\}$

Relational Operations: Restrictions vs. Subtractions

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

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

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

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

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

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

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

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

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.

$$\underset{\text{relation}}{r} \mathbin{\lhd\!\!+} \underset{\text{relation}}{t} = \{(d, r') \mid (d, r') \in t \ \underline{\underline{\lor}} \ ((\underline{\underline{d}}, r') \in r \land \underline{\underline{d}} \notin \mathrm{dom}(t))\}$$

$$= \{(d, r') \mid (d, r') \in t\} \ \underline{\underline{\cup}} \ \{(d, r') \mid (d, r') \in r \land d \notin \mathrm{dom}(t)\}$$

Lecture 7 - Sep 25

Math Review

Relational Overriding
Functional Property
Partial Functions vs. Total Functions

Announcements/Reminders

- Today's class: notes template posted
- **Event-B Summary** Document
- **Priorities**:
  - + **Lab1** → Review
  - + **Lab2** → Review
- Released:
  - + **ProgTest** guide
  - + 2 Practice Tests and solutions
  - + **Lab1**, **Lab2** solutions
  - + Possible change of **ProgTest** venue – to be confirmed

Relational Operations: Overriding

$$\{x \mid P(x) \vee Q(x)\}$$
$$= \{x \mid P(x)\} \cup \{x \mid Q(x)\}$$

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

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.

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

r overridden by t

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

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

$$= \underline{\underline{t}} \ \cup\ (\underbrace{dom(t)}_{\{a, c\}} \mathbin{\lhd\!\!\!-} r)$$

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

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

(a, 3), (c, 4)

t

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

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

Lab1 b : Account $\rightarrow\!\!\!\!\!+$ $\mathbb{Z}$

transfer from acc1 to acc2

original rel before transfer

After Transfer

b := b $\lhd\!\!\!+$ {acc1 ↦ ..., acc2 ↦ ...}

proposed changes

$r \lhd\!\!\!+ t$

changed version of r according to t

basically r, except all pairs with first elements in dom(t), they must agree with t.

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)}

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

Hint: What range of value should be included?

$$\boxed{r[s]} \overset{X}{=} s \lhd r$$

$$\subseteq \mathrm{ran}(r)$$

$$\boxed{① \ s \lhd r}$$

$$② \ s ⩤ r$$

$$③ \ r \rhd s$$

$$④ \ r ⩥ s$$

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

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?

$$r \lhd\!\!\!+ \, t = t \cup \{\mathrm{dom}(t) ⩤ r\}$$

Functional Property

$r \in S \leftrightarrow T$

isFunctional(r) ⇔

range values.

$\forall s, t1, t2 \bullet$

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

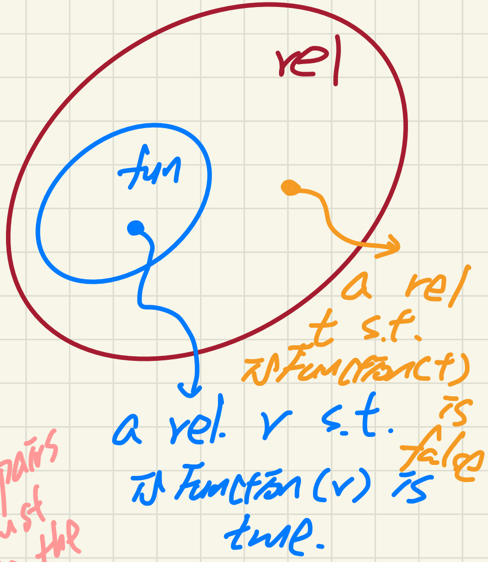
a domain value

$\Rightarrow$

two pairs sharing the same 1st elements are $\in r$

2nd elements in two pairs must be the same.

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



Q: Smallest relation satisfying the functional property.

$\emptyset$ → can't find witness of violating func. property

You cannot have the same domain value s mapping to two distinct range values t1 and t2

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

T $(a,1) \in r \land (a,3) \in r$

witness of violation ← $\Rightarrow 1 = 3$ F

Functional Property

* Each domain value maps to at most one range value

$$\text{isFunctional}(r) \Leftrightarrow$$

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

two pairs with distinct range values but same domain value cannot exist in the same relation.

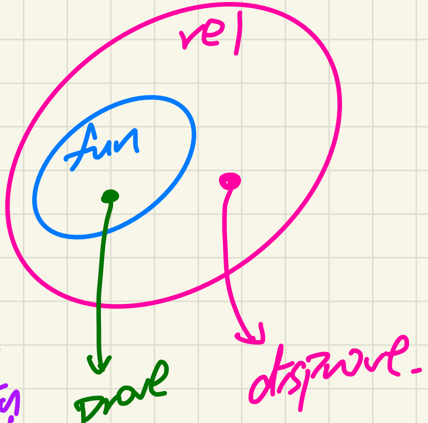
$$\forall\, s, t1, t2 \bullet$$

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

$$\Rightarrow$$

* **

$$\left(\boxed{(s, t1) \in r \land (s, t2) \in r \Rightarrow t1 = t2} \right)$$



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

Q: Rewrite the functional property using **contrapositive**.

Prove

① Show that $r = \emptyset$ $(F \Rightarrow \_ \equiv T)$

② Go over all pairs in r, show that each dom. value maps to no more than one values.

Disprove

Find $(s,t_1) \in r$ $(s,t_2) \in r$ but $t1 \neq t2$

$\nrightarrow$ partial

$\rightarrow$ total

Partial Functions vs. Total Functions

$r \in S \leftrightarrow T$

partial

$r \in S \nrightarrow T \Leftrightarrow (\ \text{isFunctional}(r) \land \boxed{\text{dom}(r) \subseteq S}\ )$

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

total

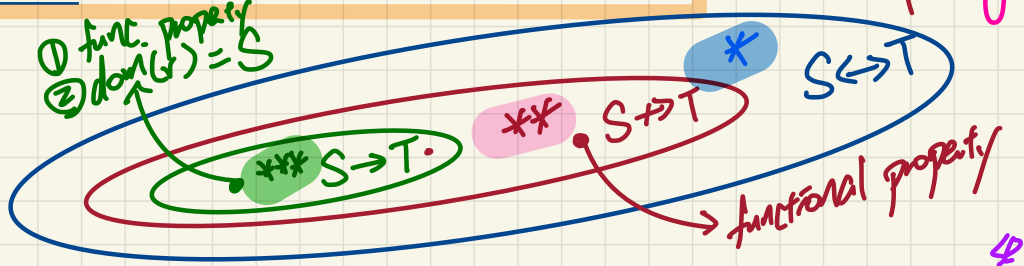
$\boxed{\text{dom}(r) \subset S} \lor \boxed{\text{dom}(r) = S}$

case 1: there's some dom value that does not has corresponding range value

case 2: every dom value has its defined range value.

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

Every function is a partial function.



set of all possible partial functions between S and T

e.g., $\{\ \{(2, a), (1, b)\}, \{(2, a), (3, a), (1, b)\}\ \} \subseteq \boxed{\underset{S}{\{1, 2, 3\}} \nrightarrow \underset{T}{\{a, b\}}}$

only check func. property.

*** e.g., $\{(2, a), (3, a), (1, b)\} \in \boxed{\{1, 2, 3\} \rightarrow \{a, b\}}$

$\text{dom}(r) = \{1, 2, 3\} = S \land \text{functional}$ ✓

** e.g., $\{(2, a), (1, b)\} \notin \underset{S}{\{1, 2, 3\}} \rightarrow \{a, b\}$

$\text{dom}(r) = \{1, 2\} \subset S$

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

violates the fun. property $\Rightarrow$ not a partial funct.

$$\text{e.g., } \{\ \overset{r1}{\underline{\{(2, a), (1, b)\}}}, \overset{r2}{\underline{\{(2, a), (3, a), (1, b)\}}}\ \} \subseteq \underset{S}{\underline{\{1, 2, 3\}}} \nrightarrow \underset{T}{\underline{\{a, b\}}}$$

$$\Longleftrightarrow$$

$$r1 \in S \nrightarrow T$$
$$\wedge$$
$$r2 \in S \nrightarrow T$$

$$\text{e.g., } \{\ \boxed{\{(2, a), (1, b)\}}, \underline{\{(2, a), (3, a), (1, b)\}}\ \} \overset{\in}{\cancel{\subseteq}} \boxed{\{1, 2, 3\} \nrightarrow \{a, b\}}$$

a set of sets of ordered pairs

a set of ordered pairs

a set of ordered pairs

a set where each member is a set of ordered pairs (sat. fun property)

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

$T = \{a, b\}$

$f(n) = 2n^2 + 3n - 4$

✓ (1) $f(n)$ is $O(n^2)$ ← most accurate.

✓ (2) $f(n)$ is $O(n^3)$

✓ (3) $f(n)$ is $O(2^n)$

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

✓ ① r is a relation.

✓ ② r is a partial function.

(most acc). ✓ ✓ ③ r is a total function.

Q1. Correct. Q2. Most accurate?

Lecture 8 - Sep 30

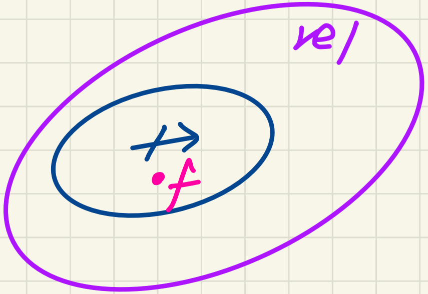
Math Review

Rel Image vs. Func Application
Modelling: Rel vs. Partial vs. Total Func
Injection, Surjection, Bijection

Announcements/Reminders

- Today's class: notes template posted
- Event-B Summary Document
- Priorities:
  - + Lab1 → Review
  - + Lab2 → Review
- Change of ProgTest venue - WSC106/108
- Released:
  - + ProgTest guide
  - + 2 Practice Tests and solutions
  - + Lab1, Lab2 solutions

Relational **Image** vs. Functional **Application**



A function is a relation.

$$\boxed{f} \in \overset{\boxed{S}}{\boxed{\{1, 2, 3\}}} \nrightarrow \overset{T}{\boxed{\{a, b\}}}$$

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

relational range

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

$$\boxed{dom(f) \subset \underbrace{\{1, 2, 3\}}_{S}} \qquad dom(f) = \{1, 2, 3\}$$

at least one value from S that does not have the corresponding range value.

Exercises:

treating f as a relation

$f[\{\underline{3}\}] = \{a\}$

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

$f[\{2\}] = \emptyset$

$\in S \notin dom(f)$

functional application

$f(3) = a$

$f(1) = b$

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

In Rodin

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

$$\boxed{f[\{c\}]}$$

$c \in \mathbb{Z}$

always well-defined

② $g : \mathbb{Z} \nrightarrow \mathbb{Z}$

$$\boxed{g(c)}$$

$c \in \mathbb{Z}$

not always well-defined

↳ one PO generated :

$c \in \mathrm{dom}(g)$.

Cardinality of relational image

$$r \in S \leftrightarrow T$$

r is also a function

$$\Rightarrow |r[\{s\}]| \le 1$$

$$s \in S$$

$$\begin{cases} = 1 \rightsquigarrow s \in \mathrm{dom}(r) \\ = 0 \rightsquigarrow s \notin \mathrm{dom}(r). \end{cases}$$

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 ∈ Employee <-> Location appropriate?

No. $e_1 \mapsto l_1 \in where\_is \wedge e_1 \mapsto l_2 \in where\_is$

Is where_is ∈ Employee → Location appropriate?

↳ No ∵ some employees may not be in the company.

Is where_is ∈ Employee ⇸ Location appropriate?

YES.

Functions

	(dom) injective	(ran) surjective	(dom, ran) bijective
partial ⇸	partial injection ⤔	partial surjection ⤀	n.a.
total →	↣	↠	⤖

** Contrapositive: $s1 \neq s2 \Rightarrow \neg((s1,t) \in f \wedge (s2,t) \in f)$

Injective Functions

same range value

func. property

`isInjective` (f)

$\iff$

$\forall s, t1, t2 \bullet (s \in S \wedge t1 \in T \wedge t2 \in T) \Rightarrow ((s,t_1) \in f \wedge (s,t_2) \in f \Rightarrow t_1 = t_2)$

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

inj. property

cannot have two distinct dom. values mapping to the same range value.

e.g. $\overset{= f}{\{(\underset{s_1}{1}, \underset{t}{b}), (\underset{s_2}{2}, \underset{t}{b})\}}$

$\hookrightarrow \left(\begin{array}{l}(1,b) \in f \\ \wedge \\ (2,b) \in f\end{array}\right) \overset{\text{false.}}{\boxed{\Rightarrow}} 1 = 2$

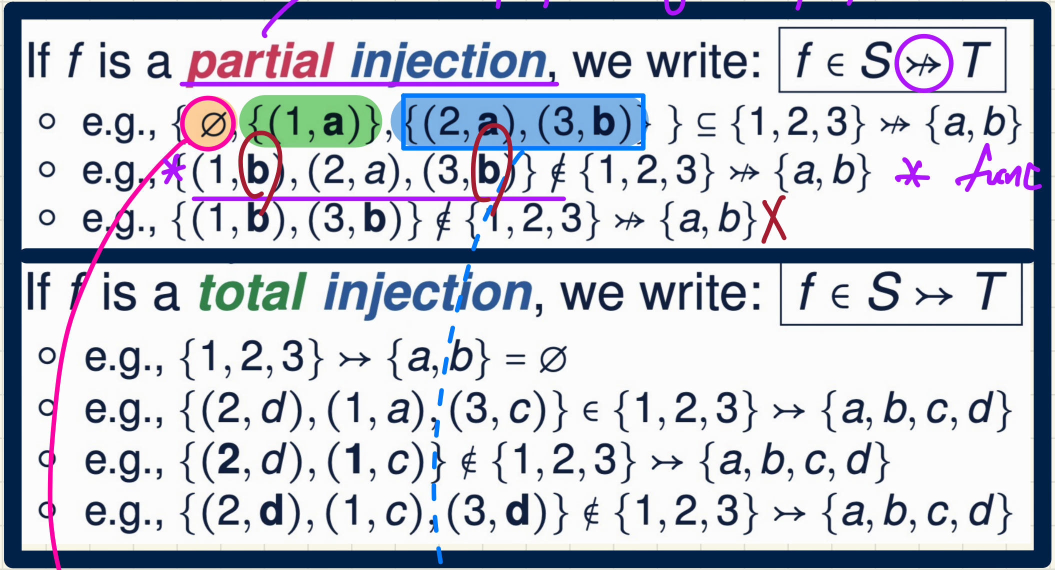
If f is a **partial injection**, we write: $\boxed{f \in S \rightarrowtail\!\!\!\!\!\!+\; T}$

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

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

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

func$^{(\forall)}$ prop. ∧ injective$^{(\forall)}$ prop.



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

- e.g., $\{\varnothing, \{(1,\mathbf{a})\}, \{(2,\mathbf{a}),(3,\mathbf{b})\}\} \subseteq \{1,2,3\} \mathrel{\rightarrowtail\!\!\!\!\!\!\!+} \{a,b\}$
- e.g., $\{(1,\mathbf{b}),(2,a),(3,\mathbf{b})\} \notin \{1,2,3\} \mathrel{\rightarrowtail\!\!\!\!\!\!\!+} \{a,b\}$ * func ✓ inj ✗ ∵ distinct dom values 1 and 2 both map to b.
- e.g., $\{(1,\mathbf{b}),(3,\mathbf{b})\} \notin \{1,2,3\} \mathrel{\rightarrowtail\!\!\!\!\!\!\!+} \{a,b\}$ ✗

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., $\{(2,\mathbf{d}),(1,c),(3,\mathbf{d})\} \notin \{1,2,3\} \rightarrowtail \{a,b,c,d\}$

sat. both func. & inj. properties (∀) trivially

distinct dom values map to distinct ran values ⇒ injective.

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

↳ function

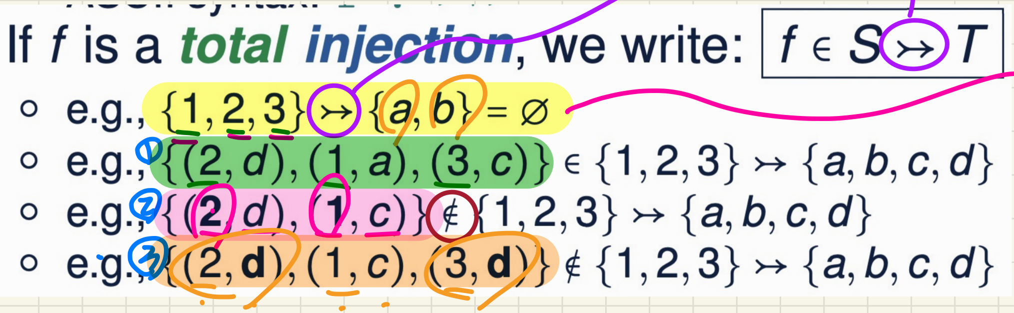
not injective

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

↳ not a function!

$S \longleftrightarrow T$: set of all relations

→ set of all possible total injections.



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

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

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

↓ violates inj. prop

witness

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

$f \in S \leftrightarrow T$

Surjective Functions

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

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

① funct. prop.

② surj. prop.

If f is a ***partial surjection***, we write: $\boxed{f \in S \text{ ⤀ } T}$

- e.g., $\{\{(1,\mathbf{b}),(2,\mathbf{a})\},\{(1,\mathbf{b}),(2,\mathbf{a}),(3,\mathbf{b})\}\} \subseteq \{1,2,3\} \text{ ⤀ } \{a,b\}$
- e.g., $\{(2,\mathbf{a}),(1,\mathbf{a}),(3,\mathbf{a})\} \notin \{1,2,3\} \text{ ⤀ } \{a,b\}$ → func not surj.
- e.g., $\{(2,\mathbf{b}),(1,\mathbf{b})\} \notin \{1,2,3\} \text{ ⤀ } \{a,b\}$

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

- e.g., $\{$ ① $\{(2,a),(1,b),(3,a)\},$ ② $\{(2,b),(1,a),(3,b)\}\} \subseteq \underline{\{1,2,3\}} \twoheadrightarrow \underline{\{a,b\}}$
- e.g., ③ $\{(\mathbf{2},a),(\mathbf{3},b)\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$
- e.g., ④ $\{(2,\mathbf{a}),(3,\mathbf{a}),(1,\mathbf{a})\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$

	total	func	surj.	
①	✓	✓	✓	✓
②	✓	✓	✓	✓
③	✗	✓	✓	✗
④	✓	✓	✗	✗

Bijective Functions

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

all possible bijections

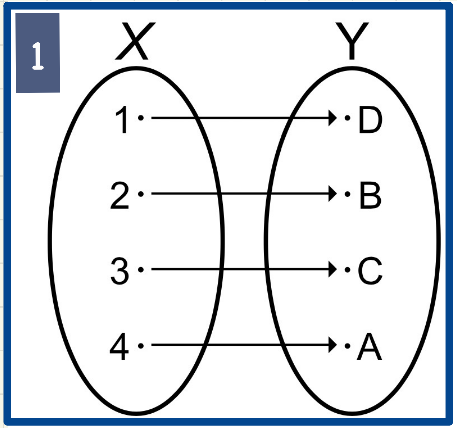
- e.g., $\{1,2,3\} \mathrel{\rightarrowtail\!\!\!\!\twoheadrightarrow} \{a,b\} = \varnothing$ → ∵ no injective function can be made
- e.g., $\{\{(1,a),(2,b),(3,c)\},\{(2,a),(3,b),(1,c)\}\} \subseteq \{1,2,3\} \mathrel{\rightarrowtail\!\!\!\!\twoheadrightarrow} \{a,b,c\}$
- e.g., ① $\{(\mathbf{2},b),(\mathbf{3},c),(\mathbf{4},a)\} \notin \{1,2,3,4\} \mathrel{\rightarrowtail\!\!\!\!\twoheadrightarrow} \{a,b,c\}$
- e.g., ② $\{(1,\mathbf{a}),(2,b),(3,c),(4,\mathbf{a})\} \notin \{1,2,3,4\} \mathrel{\rightarrowtail\!\!\!\!\twoheadrightarrow} \{a,b,c\}$
- e.g., ③ $\{(1,\mathbf{a}),(2,\mathbf{c})\} \notin \{1,2\} \mathrel{\rightarrowtail\!\!\!\!\twoheadrightarrow} \{a,b,c\}$

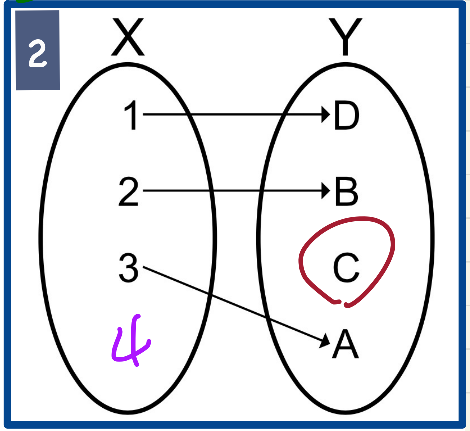
	total + func	inj	surj.	
①	✗	✓	✓	✗
②	✓	✗	✓	✗
③	✓	✓	✗	✗

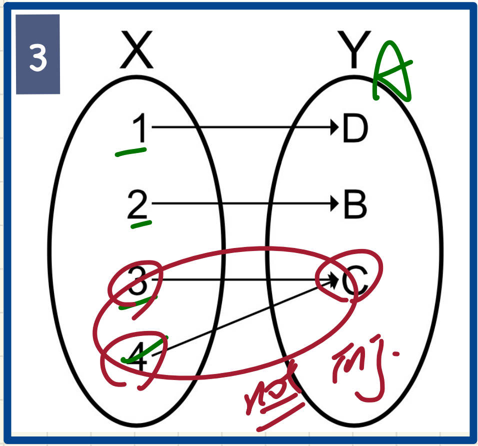
Exercise

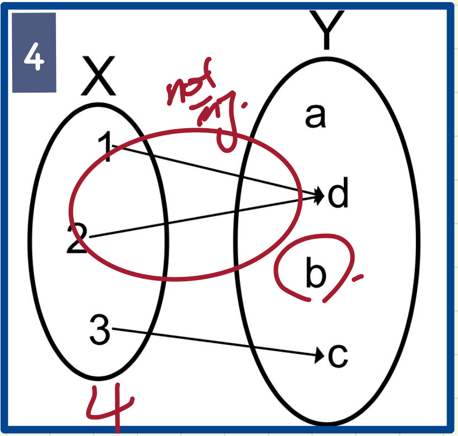
$X = \{1, 2, 3, 4\}$

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









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

Lecture 10 - Oct 7

Bridge Controller

Modelling Decision: Formulating Arrays
Correct by Construction
State Space of a Model
M0: Abstraction, Context, Machine

Announcements/Reminders

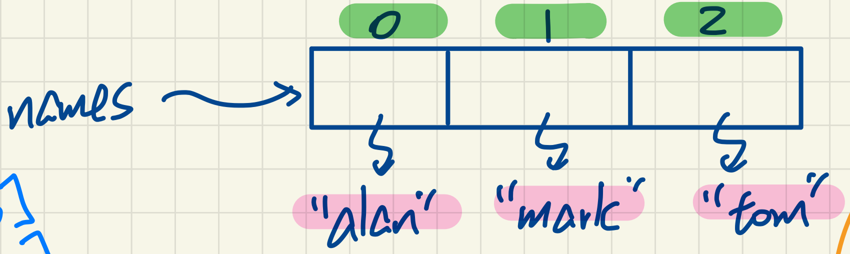
- Today's class: notes template posted
- Last Thursday's class:

  A lecture video on formal background to be released
- ProgTest being graded
- WrittenTest1 (Oct 22) coming after the reading week

Formalizing Arrays as Functions

0 1 2

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



names[0]
" [1]
" [2]

✓ (3) $names \in \mathbb{Z} \psurj \boxed{\mathbb{Z}}$ (partial surjection)

infinite

↳ $ran(names) = \mathbb{Z}$ → infeasible

Instead: $names \in \mathbb{Z} \psurj \boxed{Int}$ 32-bit integer

→ feasible if memory allows.

$names \in \mathbb{Z} \boxed{\pfun} String$

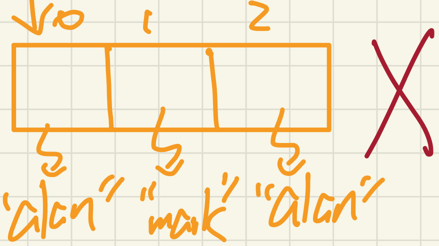
only certain indices are applicable

$names \in \mathbb{Z} \pinj String$

↳ distinct indices map to distinct strings (no dup.)

dom

ran



(4) $a \in \mathbb{Z} \;⤀\; \boxed{\text{String}}$ ⇝ the set of all possible strings.

$\text{ran}(a) = \text{String}$ → feasible

(5) → related to (3)

↳ int data type

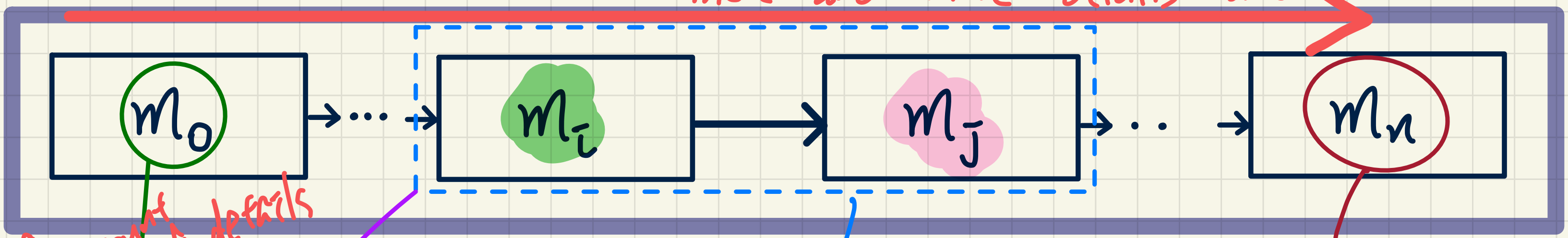
bridge controller: <m_0, m_1, m_2>

Correct by * Construction

→ * of a series of models

$n+1$ models

more and more details added



initial, most abstract model

containing the least amount of details

↳ simplest.

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

m_i is refined by ||| m_j.

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

final, most concrete model

↳ closest to the ultimate working implementation (e.g. C)

To show that m_j is a refinement of m_i, a list of Proof obligations need to be discharged.

↳ instead, distribute properties into different models.

State Space of a Model

→ concreteness ↑
→ state space ↑

* 1. typing constraints
2. invariant

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

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

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

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

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

Init. $(\underbrace{10,000}_{c}, \underbrace{20,000}_{L}, \phi)$

C1: $(10,000, 20,000, \{\text{"bill"} \mapsto 5000\})$ → sat. both typing constraint and invariant.

bottleneck of model checking

what if no invariant

C2: $(10,000, 20,000, \{\text{"bill"} \mapsto -5M\})$

should not be allowed.

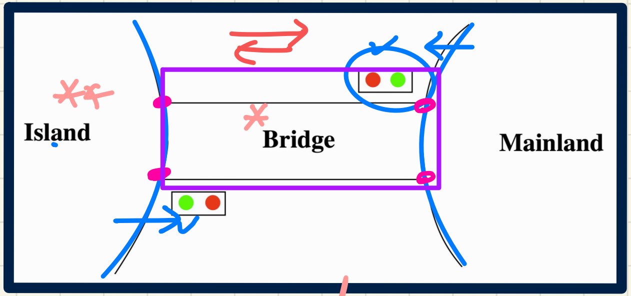
Q2. How large exactly is this initial model's state space?

State Space = { var/const list | typing constraints ∧ invariant }

very easily infinite (combinatorial explosion)

Bridge Controller:

Requirements Document



$* + ** \leq d.$

M2

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.

M3

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.

M1

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 property

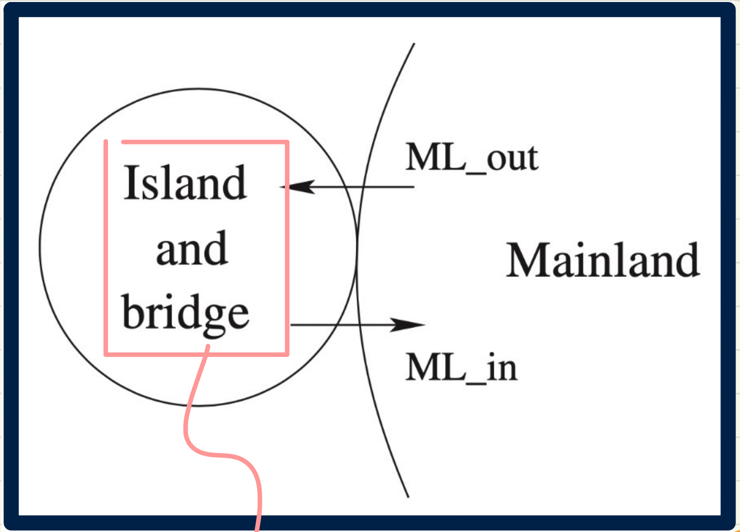
M0

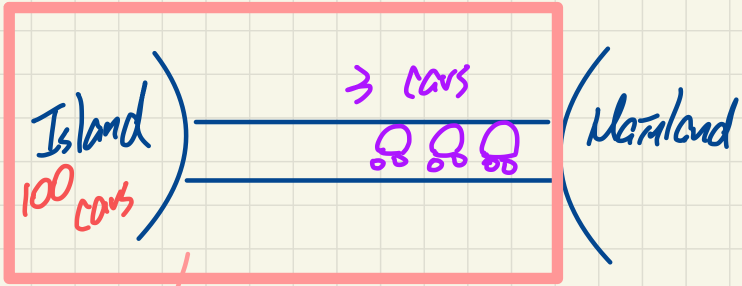
the only REQ to focus on the initial model

Bridge Controller: Abstraction in the Initial Model

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

in M_0, consider them as not separable!





n : number of cars in the Island & bridge

Island-bridge compound

$n = \underbrace{100}_{\text{island}} + \underbrace{3}_{\text{bridge}} = 103.$

Bridge Controller: State Space of the Initial Model

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

Static Part of Model

max # cars

constants: d

axioms:

$d \geq 0$

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

Dynamic Part of Model

variables: n

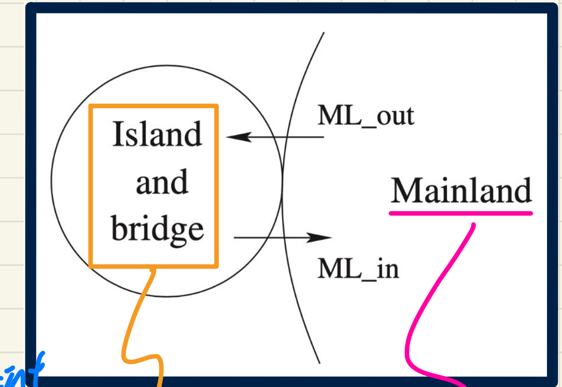
invariants:

inv0_1 : $n \in \mathbb{N}$

inv0_2 : $n \leq d$

typing constraint

desired property



$n \leq d$

current # cars

max #

cars unspecified/unbounded.

Bridge Controller: State Transitions of the Initial Model

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

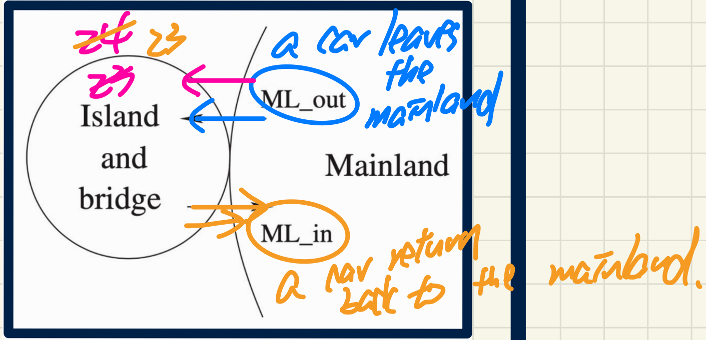
constants: d

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

problematic.

variables: n

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



Mainland

ML_out
begin
$n := n + 1$ ✓
end

no guards (always enabled).

ML_in
begin
$n := n - 1$ ✓
end

State Transition Diagram on an Example Configuration

$d = 2$
n initialized to 0

→ Is there a trace of event that can lead to inv. violation?

$d = 2$
$n =$

Lecture 11 - Oct 9

Bridge Controller

Before-After Predicates
Sequents: Syntax and Semantics
Inv. Preservation: PO/VC as as Sequent

Announcements/Reminders

- Today's class: notes template posted
- Last Thursday's class:
  A lecture video on formal background to be released
- **ProgTest** being graded
- **WrittenTest1** (Oct 22) coming after the reading week
  + Guide and example questions to be released
  + An in-person review session

* if an unsafe state is possible → something wrong with the model !

Bridge Controller: State Transitions of the Initial Model

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

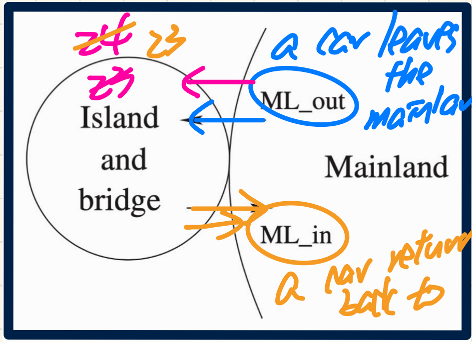
constants: d

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

problematic.

variables: n

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



Mainland

ML_out
begin
$n := n + 1$ ✓
end

ML_in
begin
$n := n - 1$ ✓
end

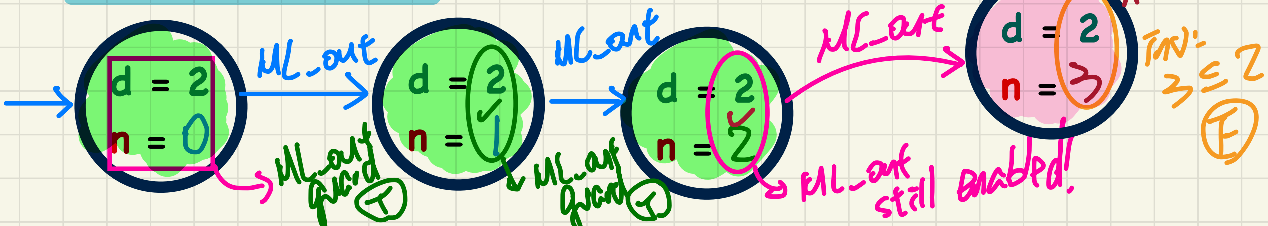
no guards (always enabled).

State Transition Diagram on an Example Configuration

d = 2
n initialized to 0

$n \leq d$

Is there a trace of event that can lead to inv. violation?



Two independent concerns
~ safe state
~ enabled event

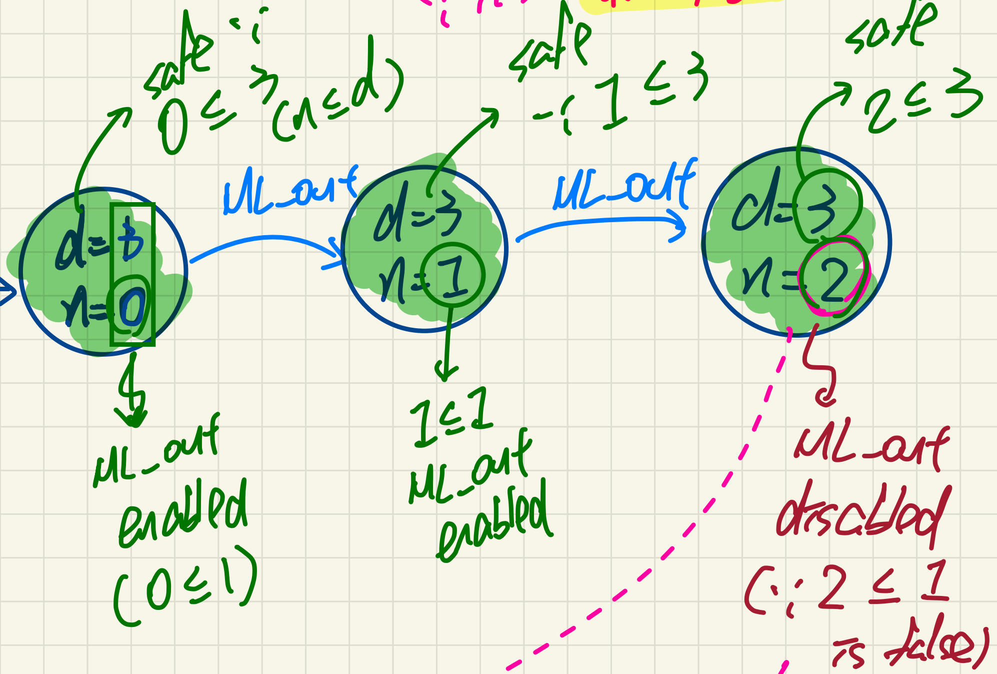
action for ML_out:

$\overset{}{n \underset{\text{becomes}}{:=} n + 1}$

guard for ML_out

$n \le \underset{d}{\boxed{1}}$

ML_out cannot happen anymore
∵ it's disabled



even though it's safe for n to be incremented once more, *

Before-After Predicates of Event Actions

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

from Rodin

post-state value of n

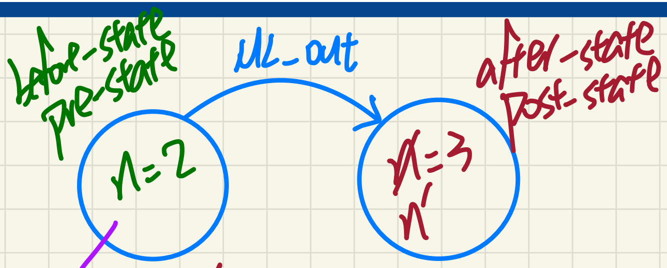
becomes

pre-state value of n plus 1

Post-state value

pre-state value

- **Pre-State**
- **Post-State**
- **Sate Transition**



$n' = n + 1$

guard of ML_out should be enabled!

For each variable x:

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

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

Exercise: Event Actions vs. Before-After Predicates

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

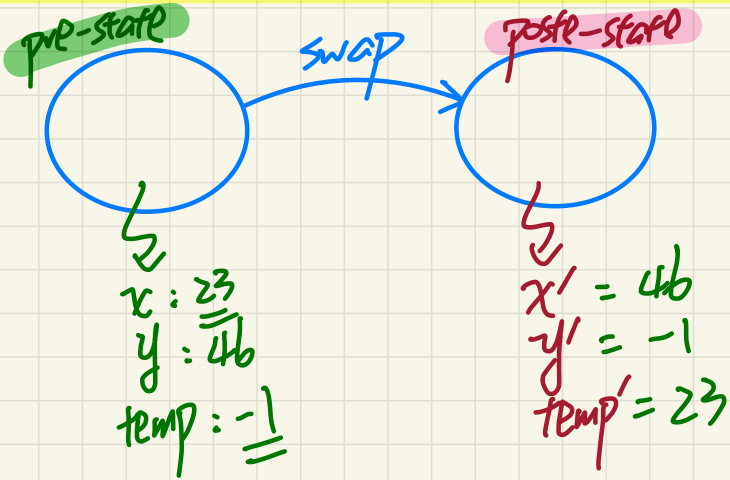
```
swap
  begin
    temp := x
    x := y
    y := temp
  end
```

Before-After Predicate

$temp' = x$

$\wedge\ x' = y$

$\wedge\ y' = temp$



Fix

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

BAP:

$x' = y$

$\wedge\ y' = x$

Design of Events: Invariant Preservation

↓ to be formulated as a proof obligation

↓ stated as a sequent .

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 : $n \leq d$

↓ Inv .

sat typing constraints

$n \in \mathbb{N}$

Desire : $\forall \overset{n}{state} \cdot state \in StateSpace$

$\Rightarrow Inv(state)$

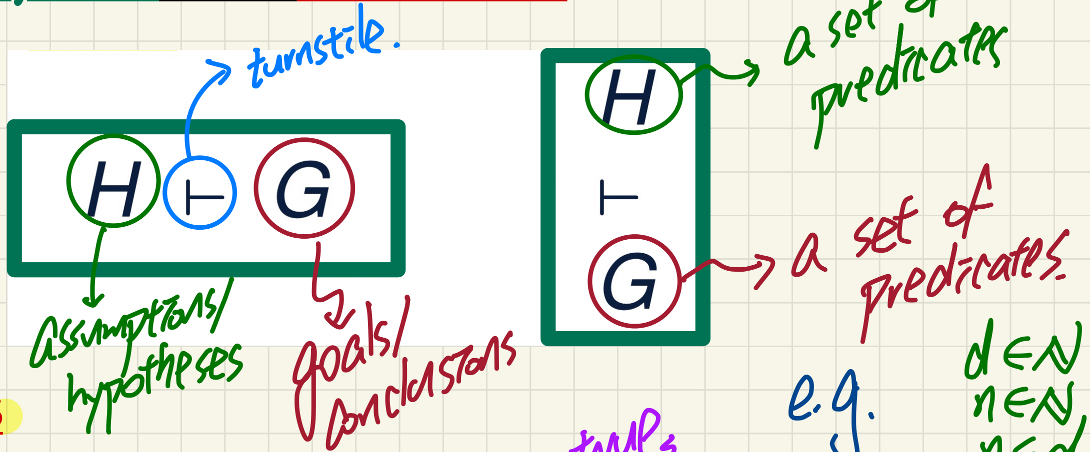
$n \leq d$

$T \Rightarrow F \equiv F$

To disprove: find a witness $state \in StateSpace$ but $\neg Inv(state)$

Sequents: Syntax and Semantics

Syntax



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

meaning of a sequent.

Semantics

true or false

$\boxed{H \vdash G} \iff H \Rightarrow G$ → assuming that H is true, G is provable.

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

$\boxed{\vdash G}$

① $\boxed{\text{True} \vdash G} \iff \text{True} \Rightarrow G \iff G$ ⇝ prove G without any assumption

② $\boxed{\text{False} \vdash G} \iff \text{False} \Rightarrow G \iff \text{True}$ ⇝ no hypotheses means it's proved!

verification condition

PO/VC Rule of Invariant Preservation

→ ML_out

Proof obligation

constants: d

variables: n

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

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

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

guard: true

BAP: $n' = n+1$

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

Inv: pre-state version
$n \in \mathbb{N}$
$n \leq d$

Inv': post-state version
$n' \in \mathbb{N}$
$n' \leq d$

* To eliminate the "primes", consider the BAP:

$\underset{n+1}{\cancel{n'}} \in \mathbb{N} \wedge \underset{n+1}{\cancel{n'}} \leq d$

Axioms
Invariants Satisfied at *Pre-State*
Guards of the Event ML_out
$\vdash$
Invariants Satisfied at *Post-State*

INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \leq d$
true ✓
$\vdash$ * $n' \in \mathbb{N} \wedge n' \leq d$
↳ $n+1 \in \mathbb{N} \wedge n+1 \leq d$

Lecture 12 - Oct 21

Bridge Controller

Before-After Predicate, Inv. Preservation
Formal Model Components
WrittenTest1 Review

Announcements/Reminders

- Today's class: notes template posted
- **ProgTest** being graded
- **WrittenTest1** (Oct 22) tomorrow

Transition of an Event

$$\boxed{b : ACCOUNT \nrightarrow \mathbb{Z}}$$

$$\underset{\text{Invariant}}{\underset{\sim}{I}} : \forall \underset{x}{\cancel{a}} \cdot \underset{x}{\cancel{a}} \in dom(b) \Rightarrow b(\underset{x}{\cancel{a}}) \geq -C$$

✓

$\boxed{\text{withdraw}}$

a : ACCOUNT

v : $\mathbb{N}$

<u>where</u>

$a \in dom(b)$

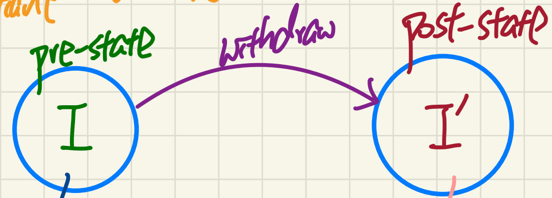
<u>begin</u>

$\overset{\text{only var. modified}}{\textcircled{b}} := \boxed{b \mathbin{\lhd\!\!\!+} \{ a \mapsto b(a) - v \}}$

effect of event.

<u>end</u>

BAP : $b' = \boxed{b \mathbin{\lhd\!\!\!+} \{ a \mapsto b(a) - v \}}$



withdraw <u>guard</u> evals to true.

Invariant maintained (I')

Axiom

$a \in dom(b)$ → guard of withdraw

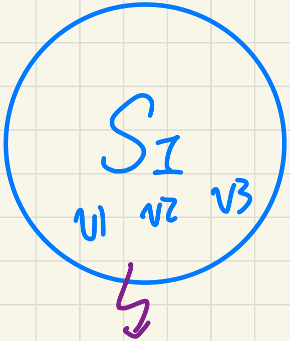
$$\cancel{I}\ \forall \overset{x}{\cancel{a}} \cdot \overset{x}{\cancel{a}} \in dom(b) \Rightarrow b(\underset{x}{\cancel{a}}) \geq -C$$

$\vdash$ ✓

$$\cancel{I'} \qquad b \mathbin{\lhd\!\!\!+} \{ \ldots \}$$

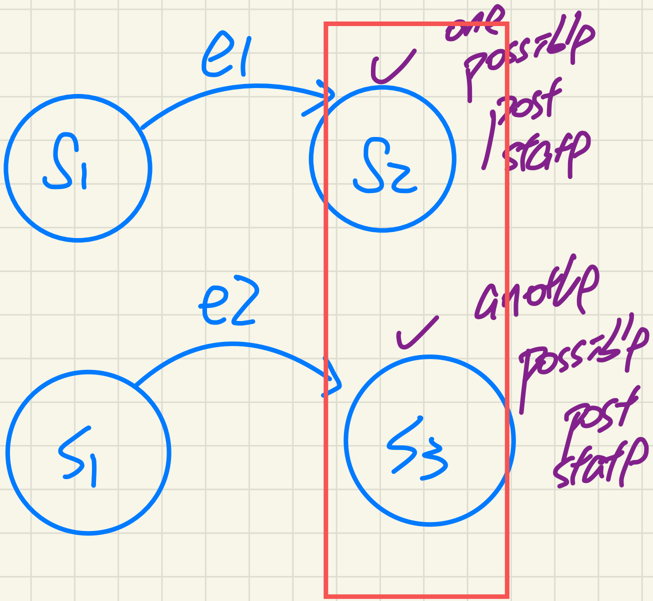
$$\forall \overset{x}{\cancel{a}} \cdot \overset{x}{\cancel{a}} \in dom(\cancel{b}) \Rightarrow \cancel{b}(\overset{x}{\cancel{a}}) \geq -C$$

$b \mathbin{\lhd\!\!\!+} \{ \ldots \}$



say: e1, e2 are enabled

~ only one event can happen at a time.



all possible poststates must maintain the invariant.

PO/VC Rule of Invariant Preservation: Components

is defined as

$c \triangleq \langle d \rangle$

constants: d

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

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

a function on constants

$v = \langle n \rangle$ pre-state

$v' = \langle n' \rangle$ post-state

variables: n

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

m_0

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

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

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

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

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

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

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

$G(\langle d \rangle, \langle n \rangle) = true.$

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

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

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

$E(\langle \rangle, \langle x, y \rangle) = \langle y, x \rangle$

- c: list of constants
- A(c): list of axioms
- v and v′: variables in pre- and post-state
- I(c, v): list of invariants

$\langle d \rangle \quad \langle n \rangle$

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

$\langle d \rangle \ \langle n \rangle$

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

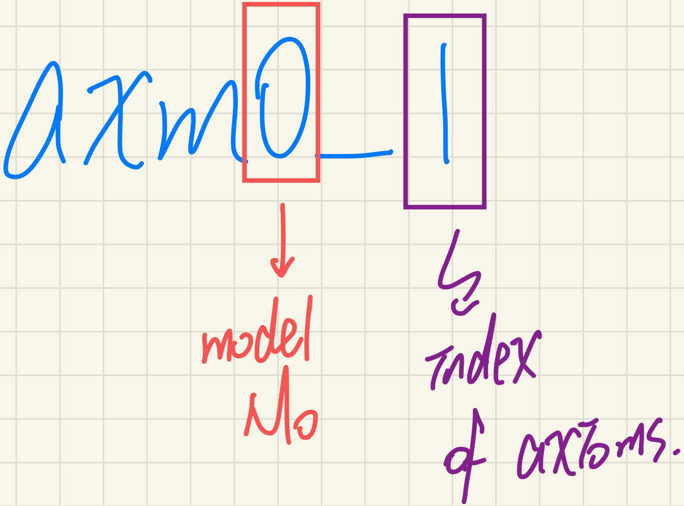
$\langle d \rangle \ \langle n \rangle \rightarrow \langle n \rangle$

effect (RHS of BAP) → effect of event action.

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

Before-after pred.

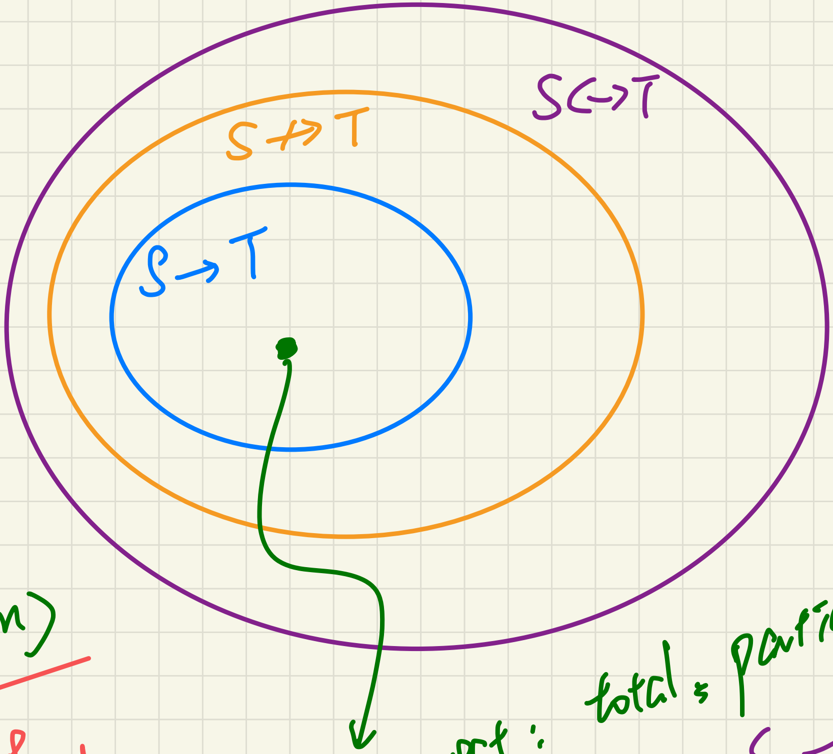
post-state value


axm0_1
model No
index of axioms.

Correct

least accurate

most accurate



Event-B summary

↳ placed desktop.

↳ print copy ~~(bring your own)~~

not to be sketched

correct : total, partial, rel

most accurate : $S \to T$

least accurate : $S \leftrightarrow T$

$$a : \mathbb{N} \nrightarrow \enclose{circle}{\mathbb{Z}}$$

set of all integers

$$a : \mathbb{N} \nrightarrow \underbrace{Int}_{\downarrow}$$

the set of 32-bit ints (in Java).

$$a : \mathbb{N} \rightarrow ?$$

($\underline{\underline{\text{not}}}$ feasible)

$$a : \mathbb{N} \mathrel{+\!\!\!\!\twoheadrightarrow} \mathbb{Z}$$

($\underline{\underline{\text{not}}}$ feasible)

$$a : \mathbb{N} \mathrel{+\!\!\!\!\twoheadrightarrow} Int$$

(feasible).

Well-defined expressions

$$\boxed{\boxed{\boxed{\neg p} \land q} \lor r} \Rightarrow \boxed{s \lor \boxed{\boxed{\neg t} \land u}}$$

$\neg$

$\land$

$\lor$

$\Rightarrow$

$$\frac{x}{y}$$ WD if $y \neq 0$

$r : S \leftrightarrow T$

$$\frac{S = \{a, b, c\}}{T = \{1, 2\}}$$

↳ a valid relation contains members from S and T only.

$r1 = \{(a, 1), (b, 2), (a, 2)\}$

$r1[\{c\}] = \emptyset$.

$r1[\{\overset{\in S}{a}, d\}]$ $d \notin S$ not well-defined.

Lecture 13 - Oct 23

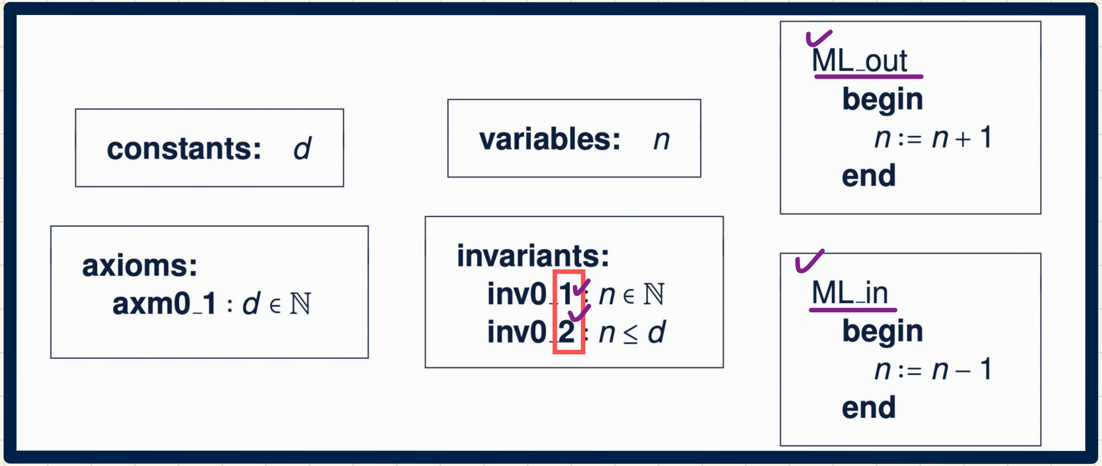
Bridge Controller

Proof Obligation Rule: Inv. Preservation
Inference Rule: Syntax and Semantics
Sequent Proofs via Inference Rules

Announcements/Reminders

- Today's class: notes template posted
- **ProgTest** results to be released by next Tuesday's class
- **WrittenTest1** results to be released by early Monday

PO/VC Rule of Invariant Preservation: Sequents



constants: d

variables: n

✓ ML_out
begin
$n := n + 1$
end

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

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

✓ ML_in
begin
$n := n - 1$
end

assume: all invariants hold in pre-state

$A(c)$ axioms

$I(c, \mathbf{v})$ pre-state version of invariant

* $G(c, \mathbf{v})$ guards of some event

$\vdash$

** $I_i(c, \mathbf{E(c, v)})$

effect of the event's action

post-state version of invariant

as a goal, prove invariant at index i 1, 2.

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

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

** Some invariant condition: inv0_1 or inv0_2 (2)

Total # of POs (sequents): 2 * 2 = 4

constants: d

variables: n

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

invariants:
inv0_1 : $n \in \mathbb{N}$
inv0_2 : $n \leq d$
inv0_3 : ...

ML_out
begin
$n := n + 1$
end

ML_in
begin
$n := n - 1$
end

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

PO1: ML_out/inv0_1/INV invariant preservation

PO2: ML_out/inv0_2/INV

PO3: ML_in/inv0_1/INV

PO4: ML_in/inv0_2/INV

PO5: ML_out/inv0_3/INV

PO6: ML_in/inv0_3/INV

ML_out / inv0_1 / INV

↓

P.O. is related to whether or not taking a state transition via some event action can preserve/maintain inv0_1.

PO/VC Rule of Invariant Preservation: Sequents

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 < true
begin
$n := n + 1$
end

BAP: $n' = n+1$

ML_in < true
begin
$n := n - 1$
end

BAP: $n' = n-1$

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

PO1 : ML_out/inv0_1/INV

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \boxed{\text{true}} \\ \vdash \\ \overset{n+1}{\cancel{n}} \in \mathbb{N} \end{array}$$

PO2 : ML_in/inv0_2/INV

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ \boxed{\text{true}} \\ \vdash \\ n-1 \; \cancel{n} \leq d \end{array}$$

Exercises

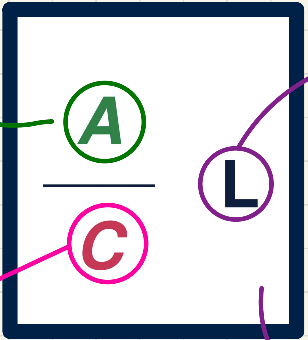
PO3 : ML_out/inv0_2/INV

PO4 : ML_in/inv0_1/INV

1. copy the INV.
2. adapt it to post-state version
3. substitute each primed var. using BAP

Inference Rule: Syntax and Semantics

Syntax



antecedent (a set of sequents)

consequent (a single sequent)

name of the inference rule

Semantics

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

$$\boxed{\frac{}{C}} \; L \quad \equiv \quad \boxed{\frac{\text{True}}{C}}$$

means C is an axiom without any proof.

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

To prove the consequent C, it's sufficient to prove the antecedent A instead

rewriting

Examples

$$\frac{H_1 \vdash G}{H_1, H_2 \vdash G} \; \text{MON}$$

singleton set of sequents

a single sequent.

MON by monotonicity

To prove $H_1, H_2 \vdash G$ it's sufficient to prove $\boxed{H_1 \vdash G}$

we can drop hypothesis.

Sequent

$$\boxed{\begin{array}{c} H \\ \vdash \\ G \end{array}} \qquad H \Rightarrow G$$

(subject to prove)

e.g. $\boxed{\begin{array}{c} H_1 \\ H_2 \\ \vdash G \end{array}}$ MON $\boxed{\begin{array}{c} H_1 \\ \vdash G \end{array}}$

Inference Rule

$$\frac{A}{C}$$

may be used as a proof step.

↳ to make progress in proofs; rewrite C as A

$A \Rightarrow C \equiv True$

↳ (assumed to be true)

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

✓

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

rewriting

relevant, useful

can be dropped

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

base case

$d \in \mathbb{N}$ not relevant
$n \in \mathbb{N}$ ✓
$n \leq d$ not useful for arguing $n+1 \in \mathbb{N}$
$\vdash$
$n + 1 \in \mathbb{N}$ goal

MON

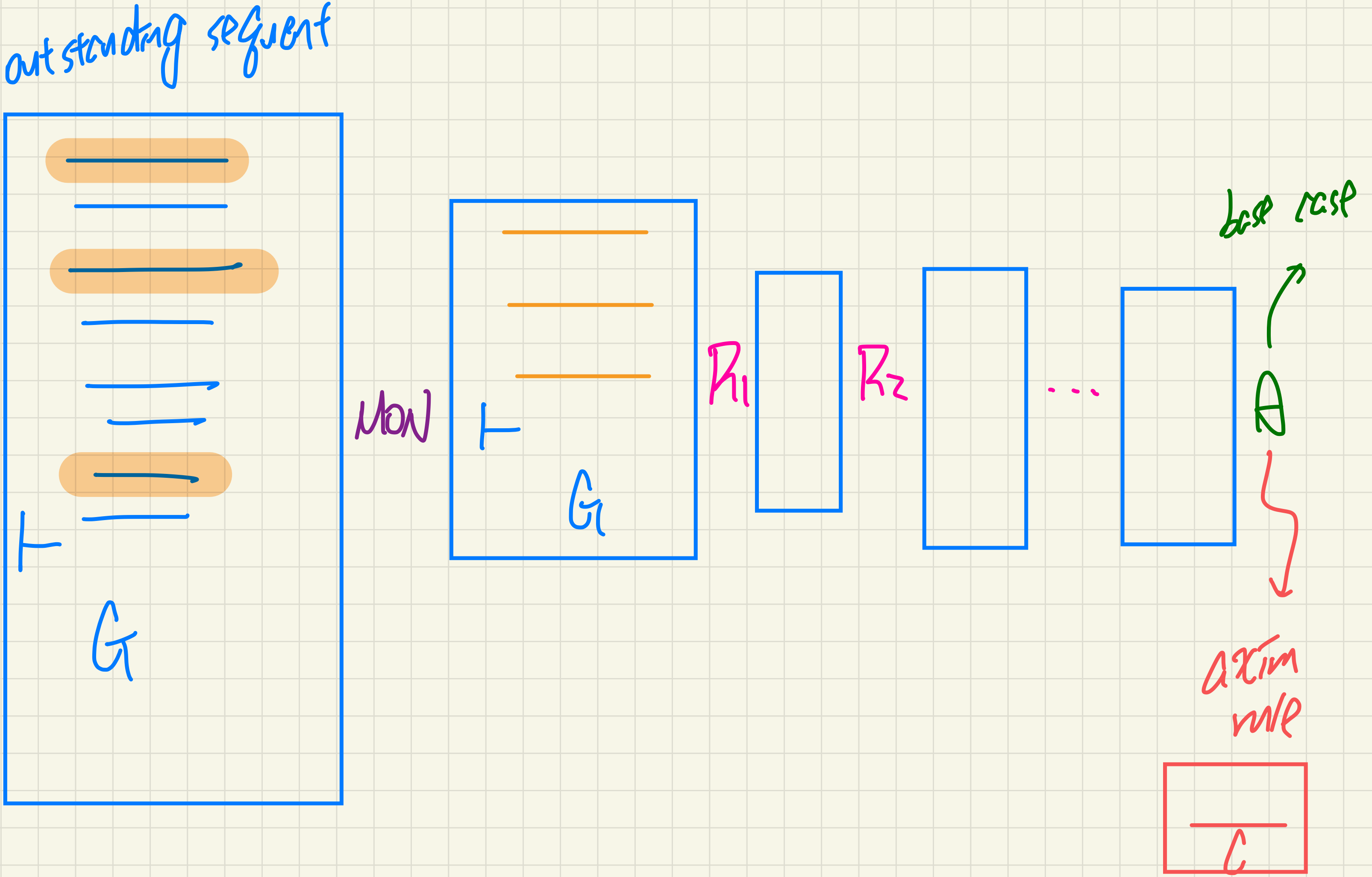
$n \in \mathbb{N}$
$\vdash$
$n+1 \in \mathbb{N}$

P2

* Analyze the goal predicate to prove:

Q1. what are the relevant vars?

Q2. what hypotheses are useful?


outstanding segment
⊢
G
Mon
⊢
G₁
R₁
R₂
. . .
best case
θ
axiom rule
C

Understanding Inference Rule: OR_L

OR_L
OR_R
AND_L

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

disjunction

OR appears to the L of ⊢

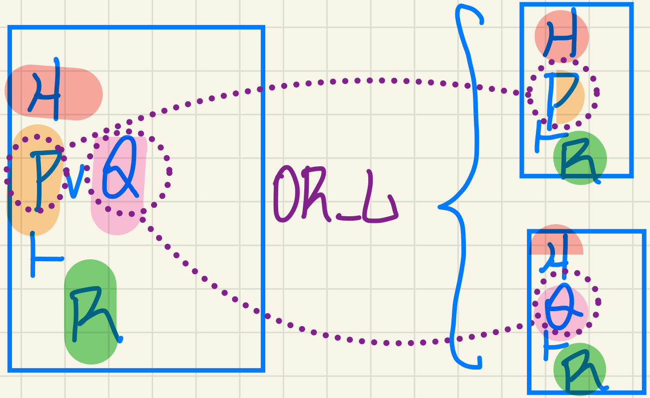
$$\frac{A}{C}$$

To prove C, sufficient to prove A

Q. Does OR_L help us: AND_R

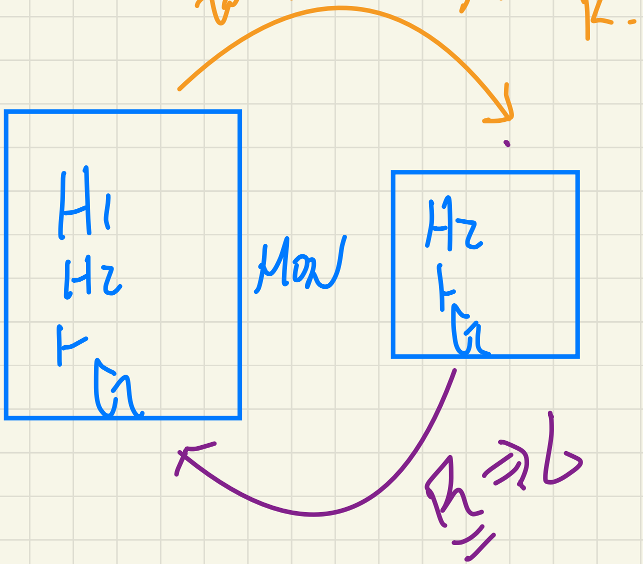
(A) split one sequent to prove into two sequents to prove

(B) Combine two sequents to prove into one sequent to prove.



$$\dfrac{\checkmark\ H_2 \vdash G}{\boxed{H_1, H_2 \vdash G}}\ \text{MON}$$

to prove the L, sufficient to prove R.



R ⇒ L

Lecture 14 - Oct 28

Bridge Controller

Justifying the OR_L Inference Rule

Interpreting Unprovable Sequents

Announcements/Reminders

- Today's class: notes template posted
- ProgTest and WrittenTest1 results released
- Tomorrow's lab sessions (1:30 to 3:30):
  Shangru to go over parts of your ProgTest
- Lab3 released

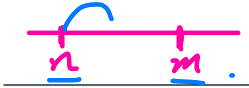
E: Peano Numbers theorems

Example Inference Rules

axiom rules (base cases of a sequent proof)

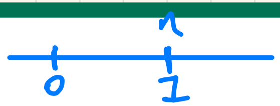
$$\frac{}{\vdash 0 \in \mathbb{N}}\ \textbf{P1}$$ ✓

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



$$\frac{}{\boxed{n < m \vdash \underline{n+1} \le \underline{m}}}\ \textbf{INC}$$

e.g. n=2 m=3 ⊢ n+1 ≤ m



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

① n=2 m=③ n-1: ① ② n=m=3

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

① n<m
② n=m

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

(recursive cases of sequent proofs)

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

to the R of ⊢

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

1

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

2

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

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

R1

H
$\vdash$
$P \vee Q$

OR_R2

H
$\vdash$
Q

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

R2

H
$\vdash$ ✓
$P \vee Q$

OR_R1

H
$\vdash$
P

Justifying Inference Rule: OR_L

$$\dfrac{\overset{\checkmark}{\cancel{H}, P \vdash R} \quad \overset{\wedge}{} \quad \cancel{H}, Q \vdash R}{\cancel{H}, P \vee Q \vdash R}\ \textbf{OR\_L}$$

$$\big((P \Rightarrow R) \wedge (Q \Rightarrow R) \Rightarrow P \vee Q \Rightarrow R\big) \equiv \text{True.}$$

$$(P \Rightarrow R) \wedge (Q \Rightarrow R)$$

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

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

$\equiv$ < distributivity of $\vee$ over $\wedge$: $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$ }

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

$\equiv$ { de morgan: $\neg(x \vee y) \equiv \neg x \wedge \neg y$ }

$$\overset{y}{\boxed{R}} \vee \neg\overset{x}{\boxed{(P \vee Q)}} \quad \equiv \{\text{def. of imp.}\} \quad P \vee Q \Rightarrow R$$

Discharging POs of original m0: Invariant Preservation

ML_out/inv0_1/INV

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

P2^{x} not valid ∵ the consequent of P2 has exactly one hypothesis

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

ML_in/inv0_1/INV

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

$n > 0$

MON $n \in \mathbb{N} \vdash n-1 \in \mathbb{N}$ unprovable ∵ $n=0$ then $n-1 \notin \mathbb{N}$

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

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

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

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

ML_out/inv0_2/INV

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

MON $n \leq d \vdash n < d$ / $n+1 \leq d$

not provable $n = d$

extra hypothesis needed to prove.

ML_in/inv0_2/INV

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

→ arithmetic

ARI $(a \leq b \equiv a < b \lor a = b)$

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$\boxed{n \leq d}$
$\vdash n-1 < d \lor n-1 = d$

MON

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

OR_R1

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

DEC

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

$\vdash$ True

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

→ ML_in/inv0_1/INV. ✓

$n' = n - 1$

given that this sequent is unprovable, we may want to add some additional hypothesis to help.

$n > 0$.

↳ where should this hypothesis go in the model?

① constants $n > 0$

② $A(c)$

$I(c, \mathbf{v})$ variables $n > 0$

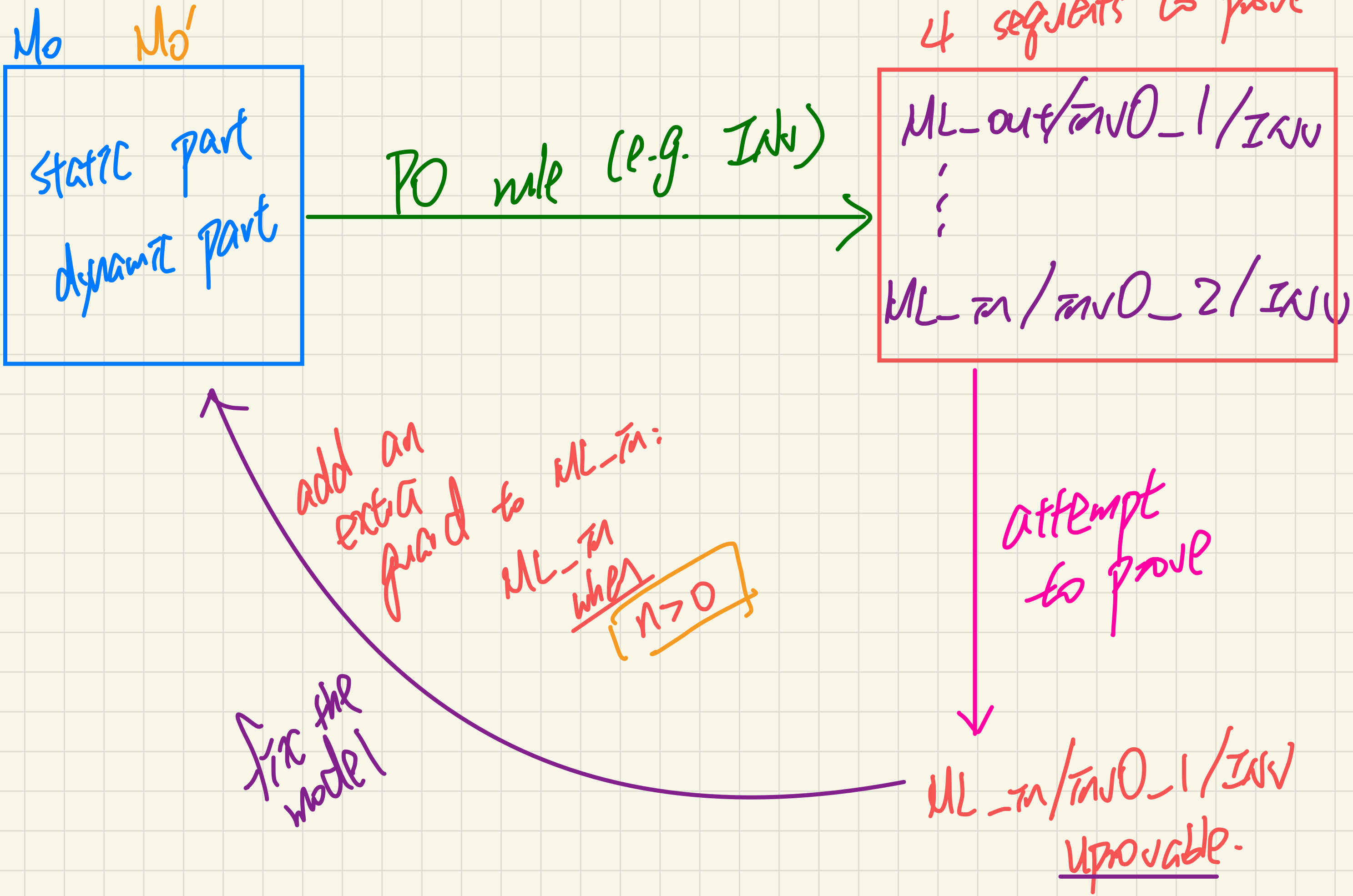
$G(c, \mathbf{v})$ ML_in

$\vdash$

$I_i(c, \mathbf{E(c, v)})$

this is the only suitable place to add the extra hypothesis.

ML_in
when
$n > 0$ → extra guard.
begin
⋮
end

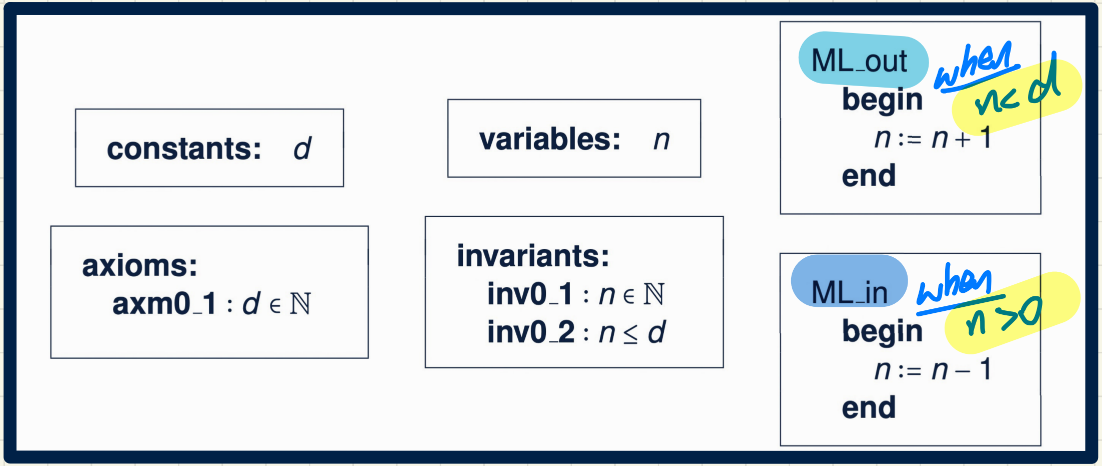

M0
M0'
static part
dynamic part
PO rule (e.g. INV)
4 sequents to prove
ML_out/inv0_1/INV
⋮
ML_in/inv0_2/INV
attempt to prove
ML_in/inv0_1/INV
unprovable.
fix the model
add an extra guard to ML_in:
ML_in
when
n>0

Lecture 15 - Oct 30

Bridge Controller

Revising M0: Adding Event Guards
Initializing System, Establishing Inv.
Deadlock Free: Intro, PO, 1st Attempt

PO/VC Rule of **Invariant** Preservation: **Revised** M0



constants: d

variables: n

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

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

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

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

$$A(c)$$
$$I(c, \mathbf{v})$$
$$\boxed{G(c, \mathbf{v})}$$
$$\vdash$$
$$\boxed{I_i}(c, \mathbf{E(c, v)})$$

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

4

Discharging POs of revised m0: Invariant Preservation

ML_out/inv0_1/INV

Ex.

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

ML_in/inv0_1/INV

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

new guard for ML_in

MON

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

P2'

ML_out/inv0_2/INV

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

New guard for ML_out

MON

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

INC

ML_in/inv0_2/INV

Ex.

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

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

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

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

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

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

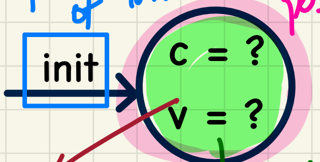
$$\frac{}{0 < n \vdash n-1 \in \mathbb{N}} \quad \text{P2'}$$

Initializing the System

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

Analogy to Induction:

Base Cases ≈ **Establishing** Invariants



no pre-state of init

post-state of init

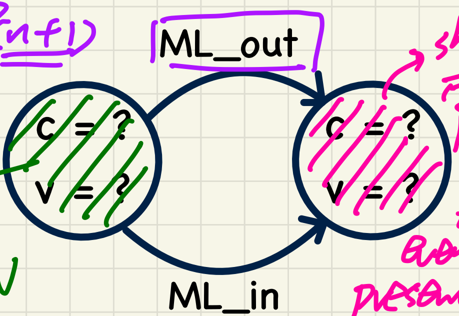
post-state of event (primed version of variables)

establishing INV for the first time.

Analogy to Induction:

Inductive Cases ≈ **Preserving** Invariants

$P(n) \Rightarrow P(n+1)$



pre-state assumed to satisfy INV

show that it's provable for the event to preserve INV

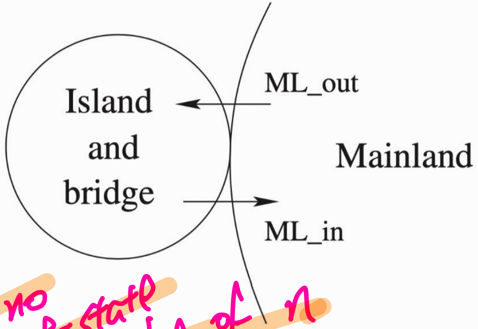
The Initialization Event

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

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

constants: d

variables: n

init
  begin
    $n := 0$
  end

BAP: $n' = 0$

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

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

$E(c, v)$ → 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

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

axioms

no invariants assumed ∵ not established yet.

index of INV

Exercise:

Generate Sequents from the INV rule.

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 X

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

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

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

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

Bridge Controller : REACTIVE SYSTEM

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

unacceptable : deadlock

no event enabled to occur

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

$\equiv$

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

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

event enabledness 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

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

DLF: deadlock freedom

at least one of the m events is enabled.

- c: list of ***constants*** $\langle d \rangle$
- $A(c)$: list of ***axioms*** $\langle \mathbf{axm0\_1} \rangle$
- $\mathbf{v}$ and $\mathbf{v'}$: list of ***variables*** in ***pre***- and ***post***-states $\mathbf{v} \mathrel{\widehat{=}} \langle n \rangle$, $\mathbf{v'} \mathrel{\widehat{=}} \langle n' \rangle$
- $I(c, v)$: list of ***invariants*** $\langle \mathbf{inv0\_1}, \mathbf{inv0\_2} \rangle$
- $G(c, v)$: 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.

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

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

Example Inference Rules

⊤ true
⊥ false

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

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

false ⇒ P ≡ True

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

↳ P ⇒ True ≡ True

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

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

✓

replace every free occurrence of E by F

$$\frac{H(E), E = F \vdash P(E)}{H(F), \underset{L}{E} = \underset{R}{F} \vdash P(F)}\ \textbf{EQ\_RL}$$

Discharging PO of DLF: First Attempt

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

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

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

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

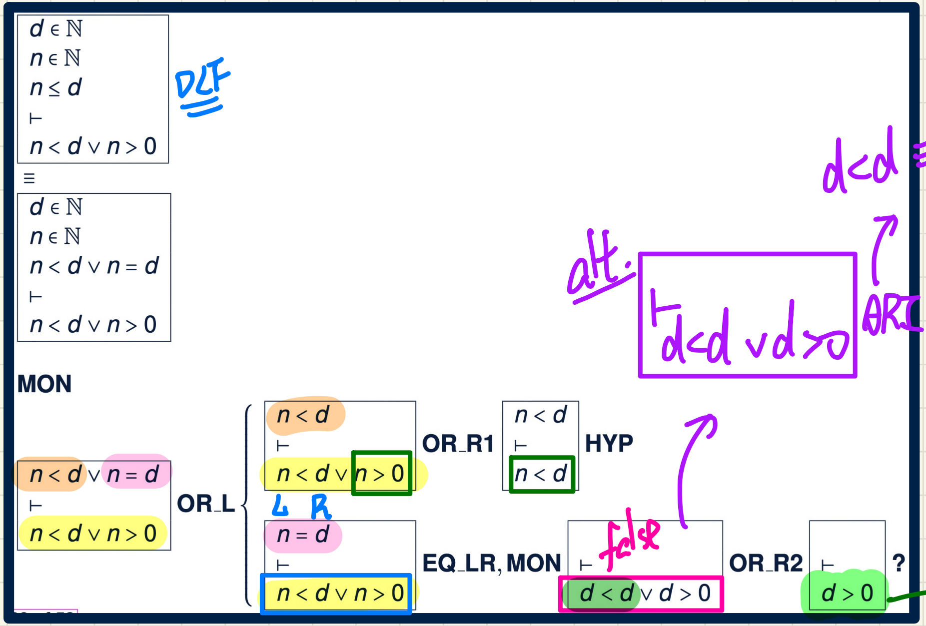
$$\frac{H \vdash Q}{H \vdash P \vee Q}\ \textbf{OR\_R2}$$

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

$\text{false} \vee P \equiv P$

$\cancel{n} < d \vee \cancel{n} > 0$

$d \qquad\qquad d$



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

DLF

$\equiv$

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

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**
- $n = d \vdash n < d \vee n > 0$ **EQ_LR, MON** $\vdash d < d \vee d > 0$ **OR_R2** $\vdash d > 0$ **?**

L R

false

alt: $\vdash d<d \vee d>0$ ARI $\vdash \text{false} \vee d>0$ ARI $\vdash d>0$

$d<d \equiv \text{false}$

unprovable given no additional hypotheses

Lecture 16 - Nov 4

Bridge Controller

DLF: Alternative Unprovable Sequent
1st Refinement: Abstraction
1st Refinement: State Space

Announcements/Reminders

- Today's class: notes template posted
- WrittenTest2 next Wednesday (November 12):
  - + Guide released
  - + Practice Questions released
  - + Lab3 solution to be release soon (for WrittenTest2)

Discharging PO of DLF: Revisiting First Attempt

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

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

replace in the goal every free occurrence of the $\boxed{R}$ by $\boxed{L}$
d n

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

$\equiv$

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

MON

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

- $n < d \vdash n < d \lor n > 0$ **OR_R1** $n < d \vdash n < d$ **HYP**
- $n = d \vdash n < d \lor n > 0$ * **EQ_LR**, **MON** $\vdash d < d \lor d > 0$ **OR_R2** $\vdash d > 0$?

what if EQ_RL was used?

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

✓✓ EQ_RL

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

MON

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

ARI

(1) $n < n \equiv F$
(2) $F \lor - \equiv -$

$\vdash n > 0$

may not be a realistic hypothesis to add back to the model

∵ IB compound may not have any ctr'

Understanding the Failed Proof on DLF

constants: d

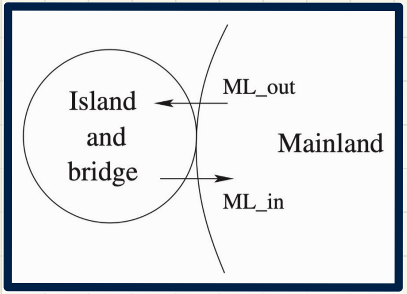
variables: n

axioms:
axm0_1 : $d \in \mathbb{N}$ ② $d \geq 0$
axm0_2 : $d > 0$

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: $\vdash d > 0$ goal.

$\underbrace{0 < 0}_{F} \lor \underbrace{0 > 0}_{F}$

↳ system deadlocks right after init when $d = 0$

Not being able to prove $d > 0$

↳ current model may violate it : $\neg(d > 0)$ is true

① $d \leq 0$ ⤳ violating the goal

② $d \geq 0$ ⤳ typing constraint

↳ $d = 0.$

Say $d = 0$.
After init. $n = 0$

G(ML_out) $\overset{0}{n} < \underset{0}{d}$ ✗   G(ML_in) $\overset{0}{n} > 0$

Fix: add extra axiom $d > 0$

Discharging PO of **DLF**: Second Attempt

$d \in \mathbb{N}$ axm0_1

$n \in \mathbb{N}$ $d>0$ /* axm0_2 */

$n \leq d$

$\vdash$

$n < d \vee n > 0$

$\equiv$

$d \in \mathbb{N}$

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

$n < d \vee n = d$

$\vdash$

$n < d \vee n > 0$

MON

$d>0$

$n < d \vee n = d$

$\vdash$

$n < d \vee n > 0$

OR_L

- $d>0$, $n < d \vdash n < d \vee n > 0$ **OR_R1** $d>0$, $n < d \vdash n < d$ **HYP**
- $d>0$, $n = d \vdash n < d \vee n > 0$ **EQ_LR, MON** $d>0 \vdash d < d \vee d > 0$ **OR_R2** $d>0 \vdash d > 0$ **?** HYP

Discharging PO of DLF: Second Attempt

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

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

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

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

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

$d \in \mathbb{N}$

$d > 0$

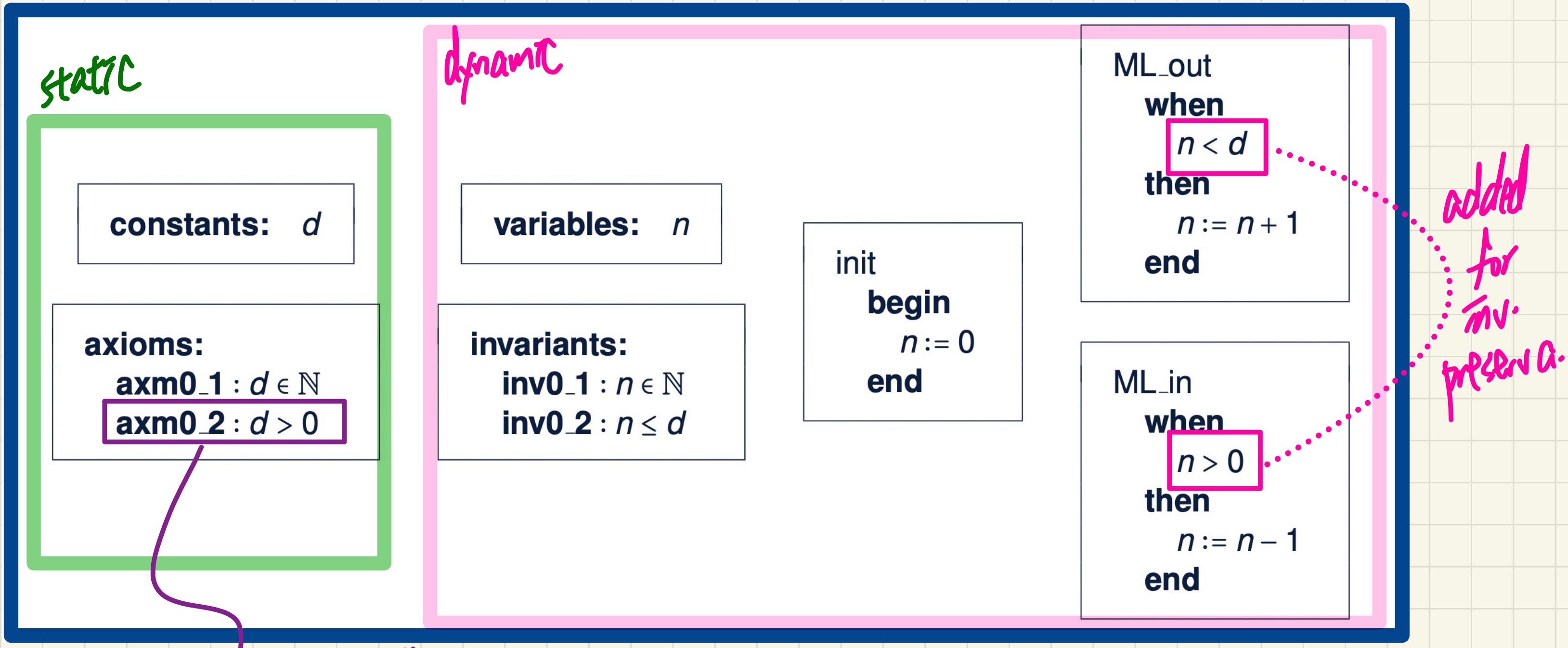
$n \in \mathbb{N}$

$n \leq d$

$\vdash$

$n < d \lor n > 0$

Summary of the Initial Model: Provably Correct



static

constants: d

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

fix for proving deadlock freedom

dynamic

variables: n

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

init
begin
$n := 0$
end

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

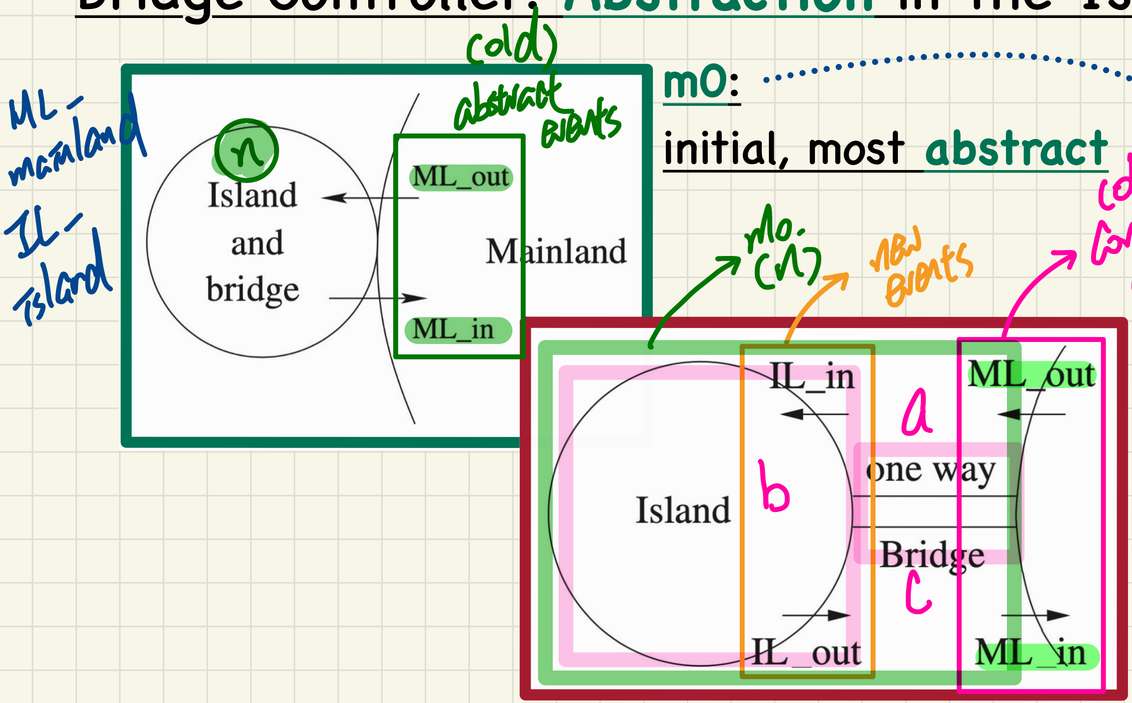
ML_in
when
$n > 0$
then
$n := n - 1$
end

added for inv. preserva.

Correctness Criteria:

+ Invariant Establishment
+ Invariant Preservation
+ Deadlock Freedom

Bridge Controller: Abstraction in the 1st Refinement



m0:

initial, most abstract

m1:

second, more concrete

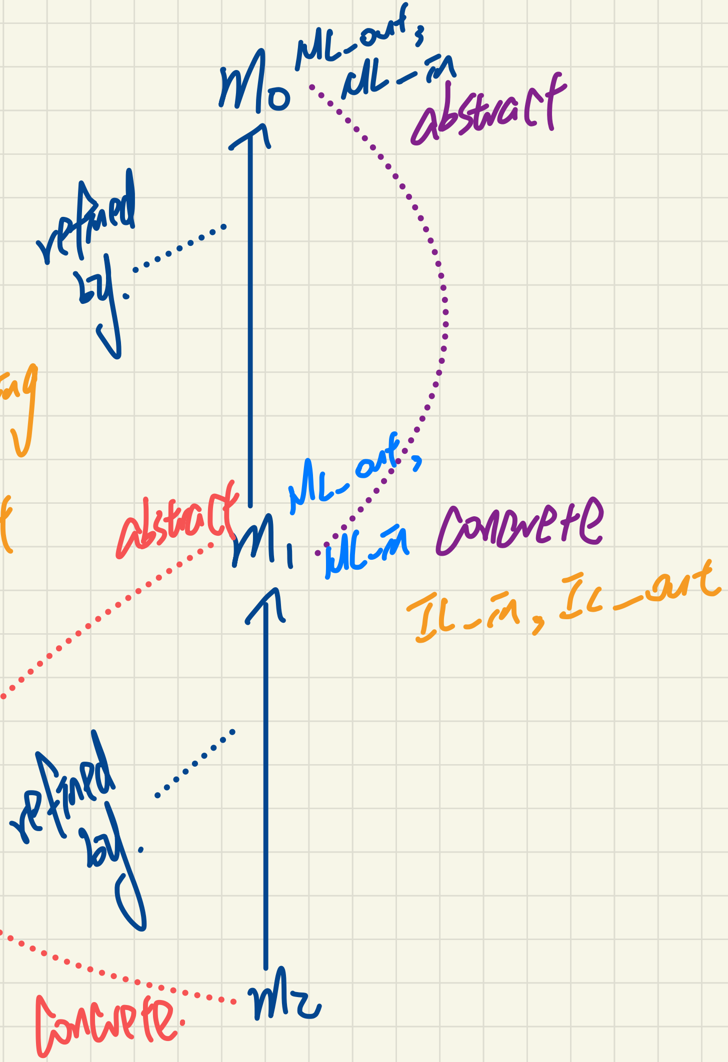
added "concreteness" (i.e., one-way bridge)

a, b, c

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.

existence \ version	M_0 abstract	M_1 concrete
M_0 old	ML_out, ML_in	ML_out, ML_in.
M_1 new	n.a.	IL_in, IL_out

events only existing in the refinement



** $\boxed{a * c = 0}$ valid but hard to use as Hyp. * $(a>0 \wedge c=0) \vee (c>0 \wedge a=0)$

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.

** $c=0$

$\boxed{\cancel{a}+c=\cancel{a}}$

$\vee$

$\cancel{a+c}=\cancel{c}$

$a=0$

Dynamic Part of Model

variables: a, b, c

linking / gluing invariant.

invariants:

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

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

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

inv1_4 : ??

inv1_5 :* ??

typing constraints

M_1

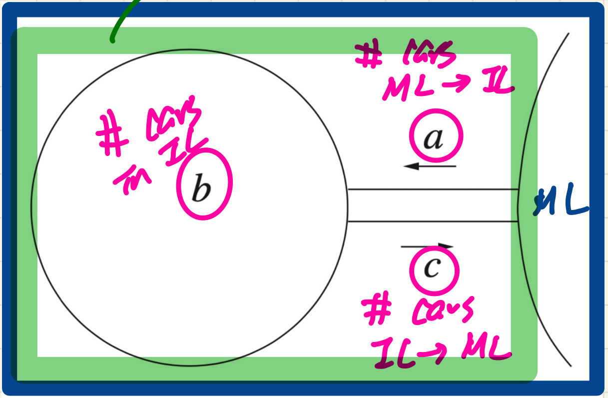
$n = a+b+c$

M_0

$\rightarrow n$ (M_0)

$a=0 \vee c=0$

safety inv.



Static Part of Model

constants: d

axioms:

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

axm0_2 : $d > 0$

Exercises

Q. Can inv1_5 be $a \neq c$? 3 2

inv1_4: linking abstract & concrete states

inv1_5: bridge is one-way

Lecture 17 - Nov 6

Bridge Controller

Concrete Guards: ML_out, ML_in
Guard Strengthening: Intuition, PO

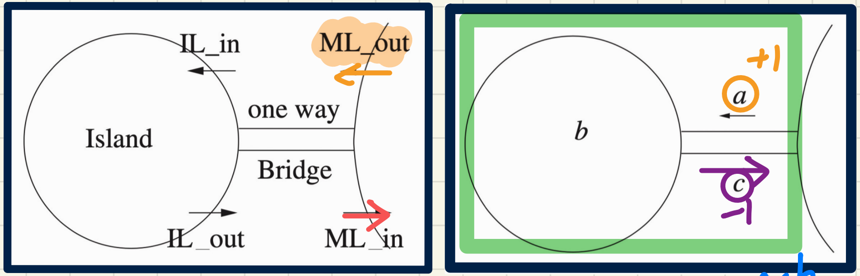
Announcements/Reminders

- Today's class: notes template posted
- WrittenTest2 next Wednesday (November 12):
  - \+ Guide released
  - \+ Practice Questions released
  - \+ Lab3 solution to be release soon (for WrittenTest2)

* if this sequent is provable → $a=0$ not necessary. (hint. OR_L)

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

$n<d$



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

```
ML_out
  when
    ??      (1) c=0
  then      (2) a+b<d
    a := a + 1
  end
```

(a) From M_0, abstract guard of ML_out: $n < d$

(b) linking INV: $a + b + c = n$

(c) concrete grd: $c = 0$

(a) $\overset{a+b}{n} < d$

(b) $a+b+\overset{0}{c}=n$

(c) $c=0$

$a+b<d$

constants: d

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

variables: a, b, c

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

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 guard: $a=0$?

*

$c > 0$ /* guard */

$a=0 \lor c=0$ /* inv1_5 */

$\vdash a=0$

m0 m1 Trace: <init, ML_out, ML_in

?

Bridge Controller: Abstract vs. Concrete State Transitions

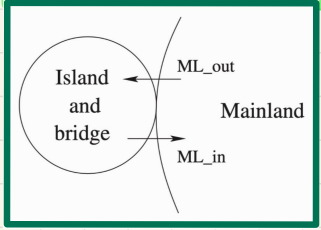
Abstract m0

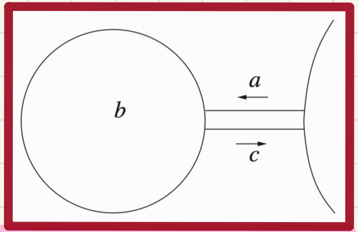
variables: n

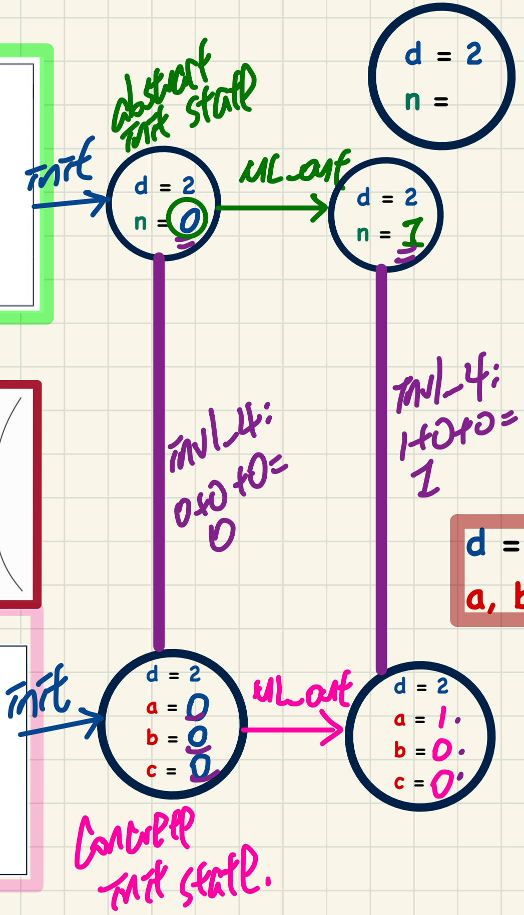
invariants:
inv0_1 : $n \in \mathbb{N}$
inv0_2 : $n \leq d$

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

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



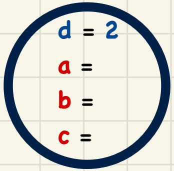




d = 2
n initialized to 0

Scenario
- car leaving ML
- car entering ML

d = 2
a, b, c initialized to 0



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

Before-After Predicates of Event Actions: 1st Refinement

Events	ML_in **when** $0 < c$ **then** $c := c - 1$ **end**	ML_out **when** $a + b < d$ $c = 0$ **then** $a := a + 1$ **end**
Before–after predicates	$a' = a \ \wedge \ b' = b \ \wedge$ $c' = c - 1$	$a' = a + 1 \ \wedge \ b' = b \ \wedge$ $c' = c$

Concrete events (a, b, c)

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

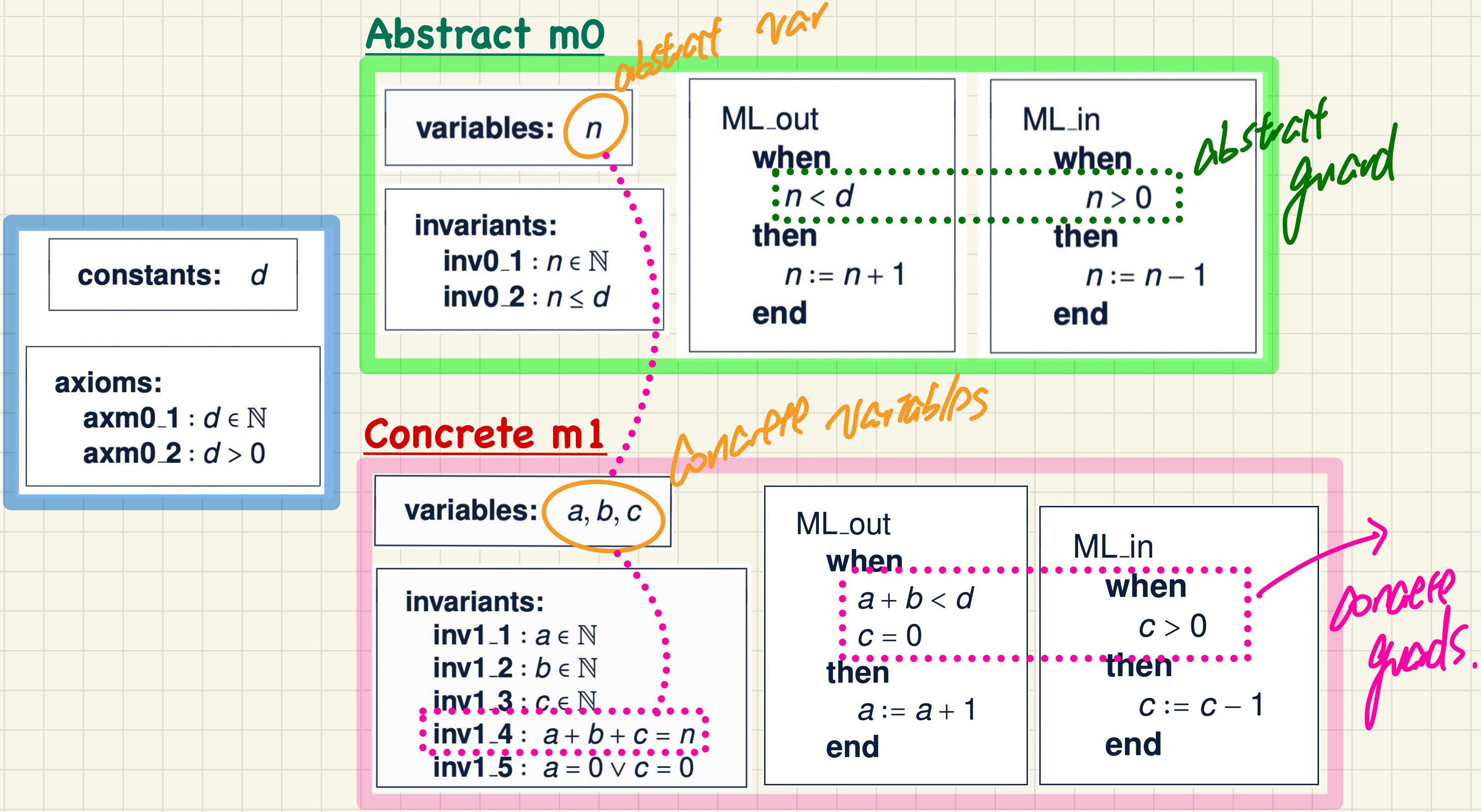
$a' = a + 1$
$\wedge b' = b \wedge c' = c$

- Pre-State
- Post-State
- Sate Transition

No: n is an abst. var.

$\wedge \boxed{n' = n}$

States, Invariants, Events: Abstract vs. Concrete



constants: d

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

Abstract m0

abstract var

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
```

abstract guard

Concrete m1

Concrete variables

variables: a, b, c

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

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

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

concrete guards.

PO Rule of Invariant Preservation in Refinement: Components

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

$G(\langle d \rangle, \langle n \rangle)$ of ML_out: $\langle n < d \rangle$

$E(\langle d \rangle, \langle n \rangle)$ of ML_out

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

$H(\langle d \rangle, \langle a, b, c \rangle)$ of ML_out: $\langle a + b < d, c = 0 \rangle$

$E(\langle d \rangle, \langle a, b, c \rangle)$ of ML_out

$v = \langle n \rangle$ $\quad v' = \langle n' \rangle$

$w = \langle a, b, c \rangle$ $\quad w' = \langle a', b', c' \rangle$

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

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

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

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

I(c, v): list of abstract invariants

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

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

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

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

$J(\langle d \rangle, \langle n \rangle, \langle a, b, c \rangle) = \langle inv1\_1, \ldots, inv1\_5 \rangle$

Predicates: Weaker vs. Stronger

$p \Rightarrow q$

↳ p is "stronger" than q

↳ q is "weaker" than p

Weakest predicate: True

$\{x \mid \text{True}\}$

Strongest predicate: False

$\{x \mid \text{False}\} = \emptyset$

$p(x) \triangleq x > 0$

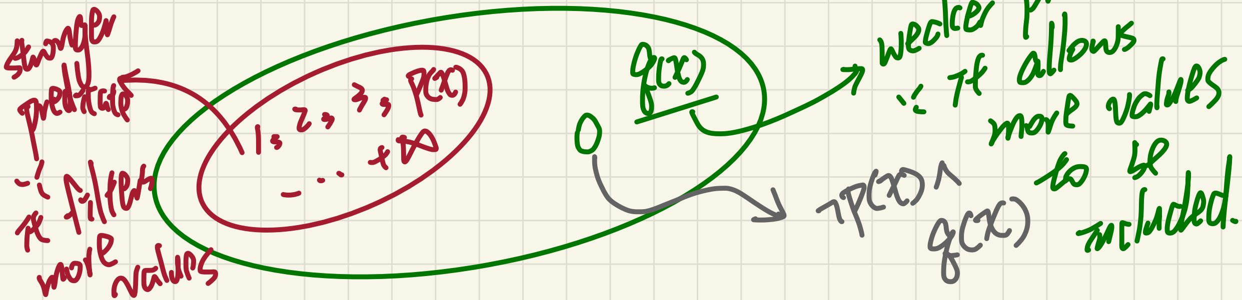
$q(x) \triangleq x \geqslant 0$

$p(x) \Rightarrow q(x)$ ✓

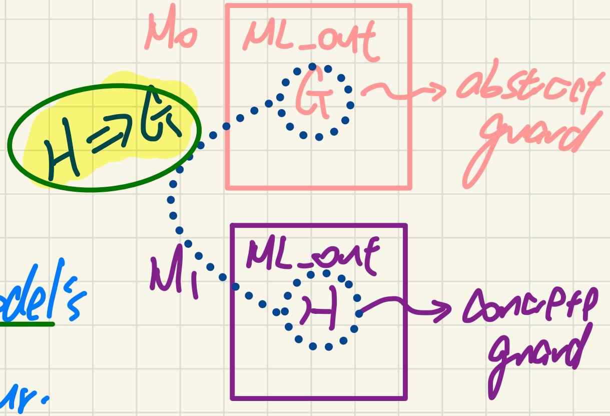
$q(x) \Rightarrow p(x)$ ✗

✓

$$\boxed{\{x \mid p(x)\}} \subseteq \boxed{\{x \mid q(x)\}}$$



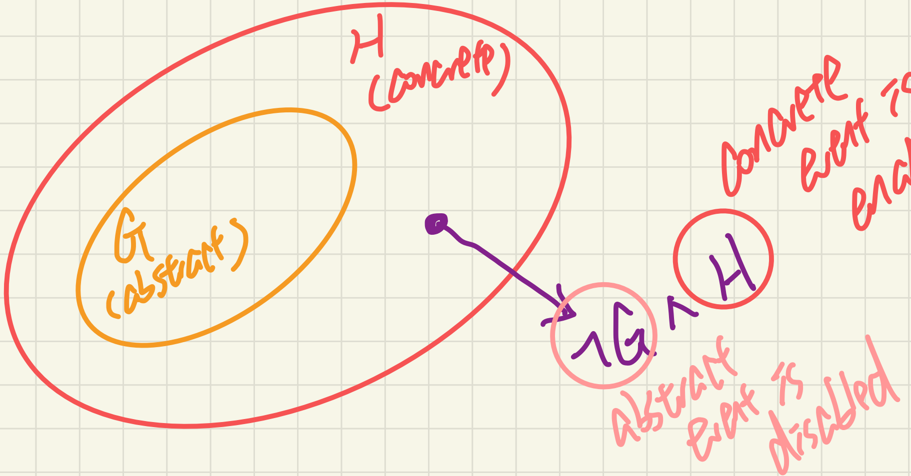
Refinement: Why Guard Strengthening



Principle: a $\overset{M_1}{\text{refinement's}}$ behaviour should be consistent with the $\overset{M_0}{\text{abstract model's}}$ behaviour.

Why is it wrong: $\overset{\text{abstract}}{G} \Rightarrow \overset{\text{concrete}}{H}$

($\Rightarrow$ no new behaviour introduced by the concrete event).



new behaviour
(not acceptable)

e.g. for some value, ML_out is disabled in M0 but enabled in M1

PO/VC Rule of Guard Strengthening: Sequents

Abstract m0

variables: n

invariants:

inv0_1 : $n \in \mathbb{N}$

inv0_2 : $n \le d$

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

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

Concrete m1

variables: a, b, c

invariants:

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

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

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

inv1_4 : $a + b + c = n$

inv1_5 : $a = 0 \vee c = 0$

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

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

each abstract guard

$A(c)$ axioms

$I(c, v)$

$J(c, v, w)$

$H(c, w)$

$\vdash$

$G_i(c, v)$

concrete grd ⇒ abstract grd

ML_out/GRD

$d \in \mathbb{N}$	axm0_1
$d > 0$	axm0_2
$n \in \mathbb{N}$	inv0_1
$n \le d$	inv0_2
$a \in \mathbb{N}$	$a+b+c=n$
$b \in \mathbb{N}$	$a=0 \vee c=0$
$c \in \mathbb{N}$	
$a+b<d$	$c=0$

$\vdash n < d$

exercise: ML_in/GRD

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

Written Test 2

slide 58 .

~

ML-out

$\boxed{n < d}$

ML-out

$c = 0$

$a + b \leq d$

Lecture 18 - Nov 11

Bridge Controller

Guard Strengthening: Review
INV Preservation: POs
INV Preservation: Commuting Diagram

Announcements/Reminders

- Today's class: notes template posted
- **WrittenTest2** Wednesday (November 12)

Refinement: Guard Strengthening

$\neg G \Rightarrow \neg H$

$H \Rightarrow G$

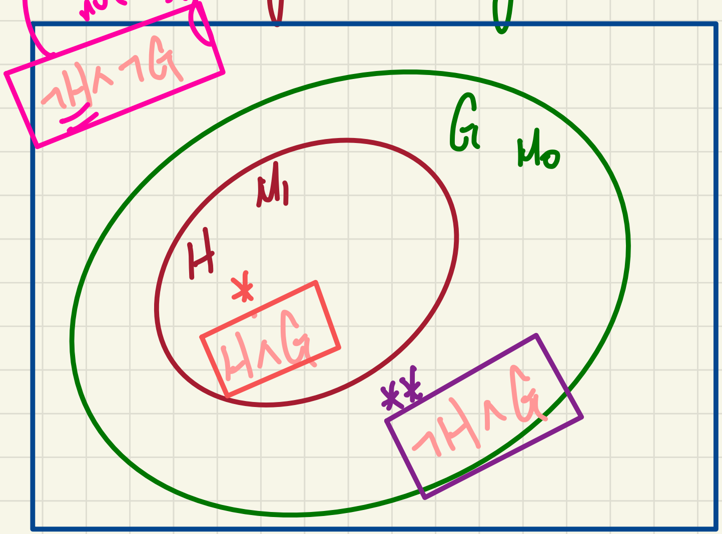
concrete guard (H)

abstract guard (G)

what's not allowed by M0 is also not allowed by M1.

* what's enabled in M1 is also enabled in M0.

** some scenarios allowed by M0 is not allowed by M1. (ok ∵ no new scenarios created by M1).



M0

ML_out

$n < d$

(G)

M1

ML_out

$a+b<d$

(H)

$a+b+c=n$

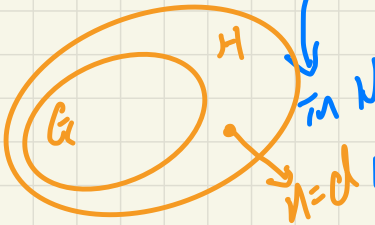
$c=0$

change this to $\le$

Q. guard strengthening?

A. $\underset{n}{\cancel{a+b}} \le d \Rightarrow n<d$

in M1, ML_out is enabled when $n=d$, but it is disabled in M0.



Discharging POs of m1: Guard Strengthening in Refinement

ML_out/GRD

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

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

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

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

Discharging POs of m1: Guard Strengthening in Refinement

ML_in/GRD

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

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

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

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

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

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

PO/VC Rule of Invariant Preservation: Sequents

Abstract m0

$* \; \underset{a+1}{\cancel{a}} + \underset{b}{\cancel{b}} + \underset{c}{\cancel{c}} = \cancel{n}\; n+1$

variables: n

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

ML_out
when
$n < d$
then
$\boxed{n := n + 1}$
end

$n' = n+1$

ML_in
when
$n > 0$
then
$\boxed{n := n - 1}$
end

$n' = n-1$

pre-state

$\boxed{A(c)}$ axioms
$\boxed{I(c, v)}$ abst. Inv.
$\boxed{J(c, v, w)}$ conc. Inv.
$\boxed{H(c, w)}$ concrete guard
$\vdash$
$J_i(c, E(c, v), F(c, w))$

concrete inv. sat in post state

Concrete m1

$** \; \underset{a}{\cancel{a}} = 0 \vee \underset{c-1}{\cancel{c}} = 0$

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
$\boxed{a := a + 1}$
end

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

ML_in
when ✓
$c > 0$
then
$\boxed{c := c - 1}$
end

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

2 con. evts * 5 con. inv. = 10 POs.

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

ML_out/inv1_4/INV

$d \in \mathbb{N}$
$d > 0$
$n \in \mathbb{N}$
$n \le d$
$a \in \mathbb{N} \;\; b \in \mathbb{N} \;\; c \in \mathbb{N}$
$a+b+c = n$
$a = 0 \vee c = 0$
$a+b < d \quad c = 0$
$\vdash *$
$\boxed{(a+1)+b+c = n+1}$

ML_in/inv1_5/INV

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

linking invariant:

Visualizing Invariant Preservation in Refinement

(1) defined in the concrete model

(2) involves both abst. & concrete variables

commuting diagram

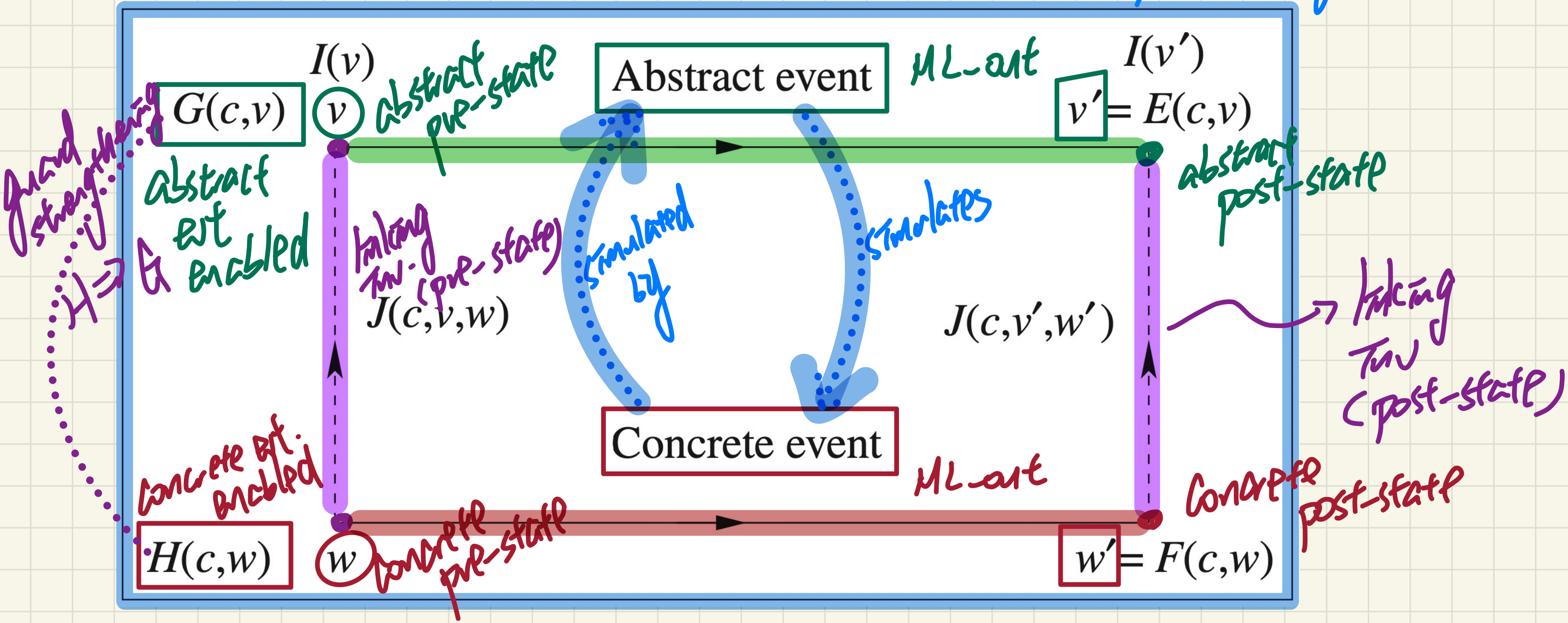
ML_out

Each **concrete state transition** (from w to w′)

should be **simulated by** → found correspondence to.

ML_out

an **abstract state transition** (from v to v′)



Discharging POs of m1: Invariant Preservation in Refinement

ML_out/inv1_4/INV

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

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

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

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

Discharging POs of m1: Invariant Preservation in Refinement

ML_in/inv1_5/INV

$d \in \mathbb{N}$

$d > 0$

$n \in \mathbb{N}$

$n \leq d$

$a \in \mathbb{N}$

$b \in \mathbb{N}$

$c \in \mathbb{N}$

$a + b + c = n$

$a = 0 \lor c = 0$

$c > 0$

$\vdash$

$a = 0 \lor (c - 1) = 0$

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

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

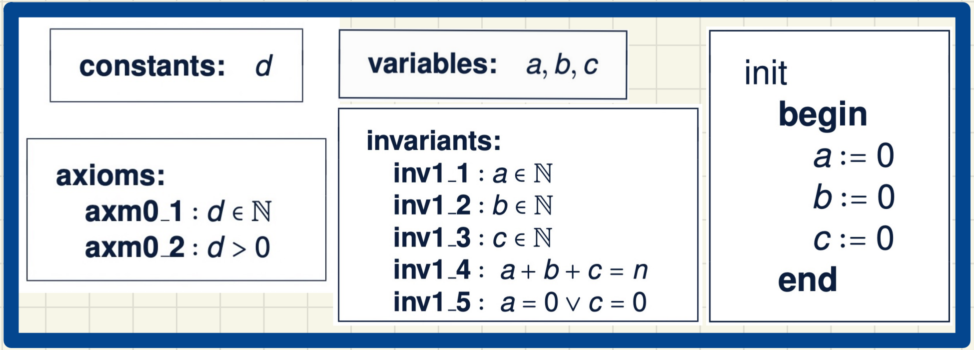
$$\frac{H \vdash P}{H \vdash P \lor Q} \quad \textbf{OR\_R1}$$

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

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

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

PO of Invariant **Establishment** in **Refinement**



constants: d

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

variables: a, b, c

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

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

Components

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

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

Rule of **Invariant Establishment**

$$A(c) \vdash J_i(c, \mathbf{K(c)}, \mathbf{L(c)})$$

Exercise:

Generate Sequents from the INV rule.

⑤

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

Discharging PO of Invariant Establishment in Refinement

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

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

$$\begin{array}{l} d \in \mathbb{N} \\ d > 0 \\ \vdash \\ 0 + 0 + 0 = 0 \end{array}$$

init/**inv1_4**/INV

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

init/**inv1_5**/INV

Lecture 19 - Nov 13

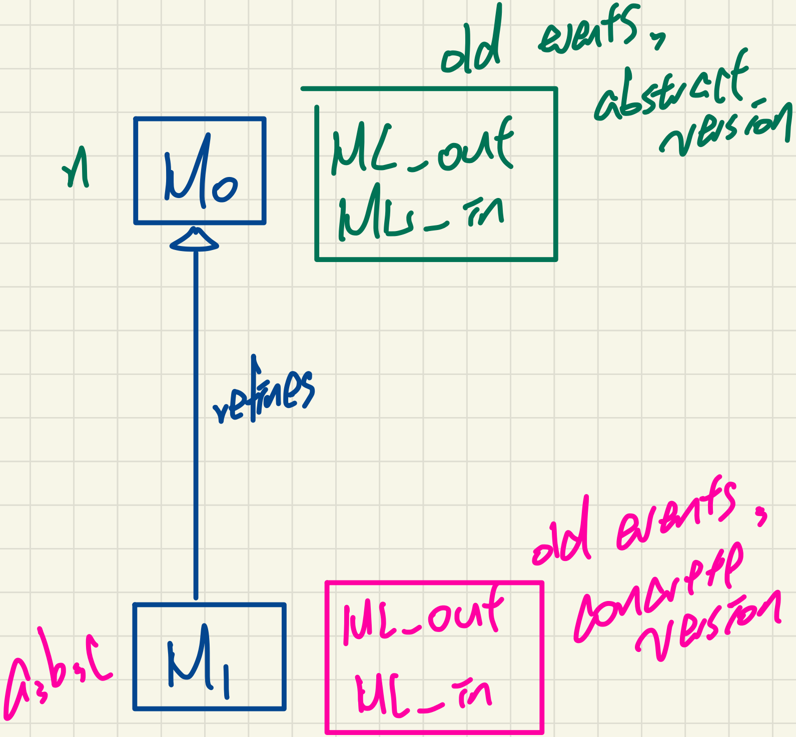
Bridge Controller

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

Announcements/Reminders

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

Events



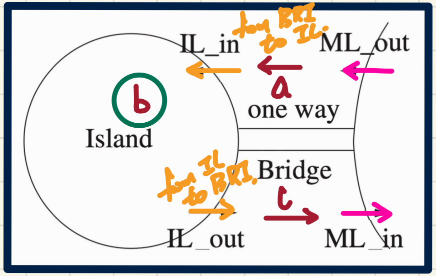
New events: IL_in, IL_out

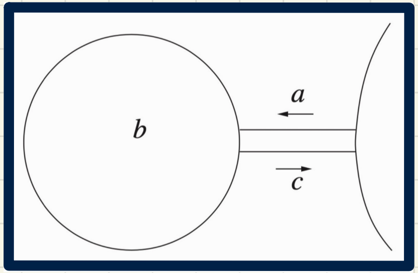
Island.

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

skip.

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





constants: d

axioms:

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

axm0_2 : $d > 0$

variables: a, b, c

invariants:

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

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

inv1_3 : $c \in \mathbb{N}$ ✓

inv1_4 : $a + b + c = n$

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

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

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

① $a > 0$ ✓

② $c = 0$

necessary ?

$a > 0$

$a = 0 \lor c = 0$

$\vdash c = 0$

$a := a - 1$

$b := b + 1$

pre : $a + b + c = n$

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

IL_out: A car exits **island** (getting on the **bridge**). as if nothing happened in M0.

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

① $b > 0$

② $a = 0$

all Inv

$b > 0$

$\vdash a = 0$

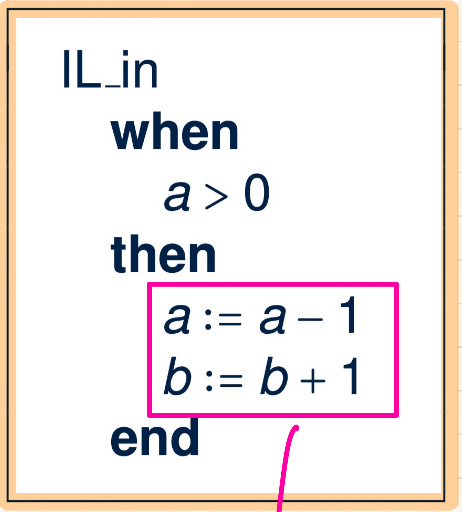
$b := b - 1$

$c := c + 1$

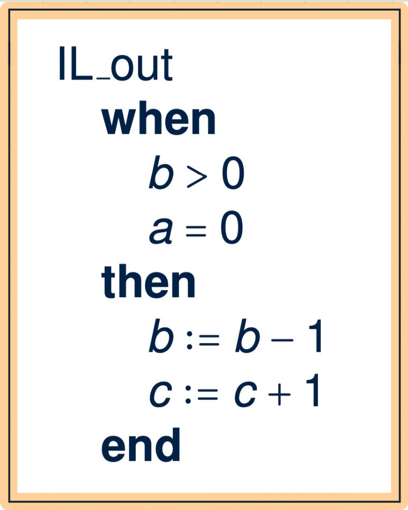
pre : $a + b + c = n$

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

Before-After Predicates of Event Actions: 1st Refinement



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

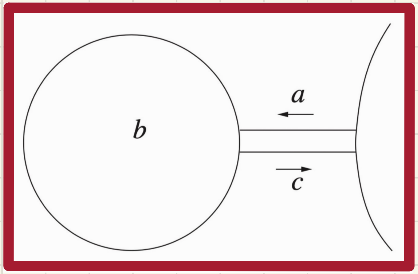


```
IL_out
  when
    $b > 0$
    $a = 0$
  then
    $b := b - 1$
    $c := c + 1$
  end
```

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

- Pre-State
- Post-State
- Sate Transition

Concrete State Space



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

Visualizing Invariant Preservation in Refinement

IL_in, IL_out

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

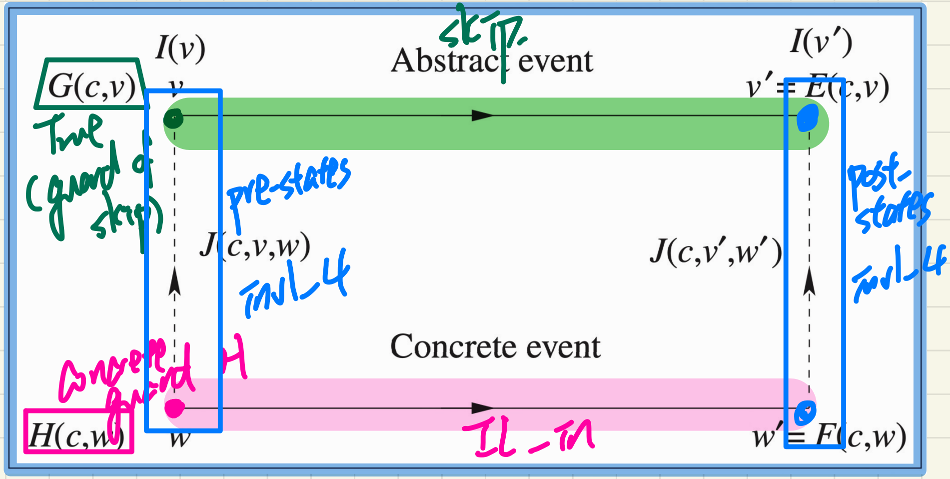
skip $v' = v$

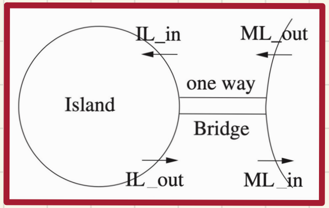
```
skip
begin

end
```

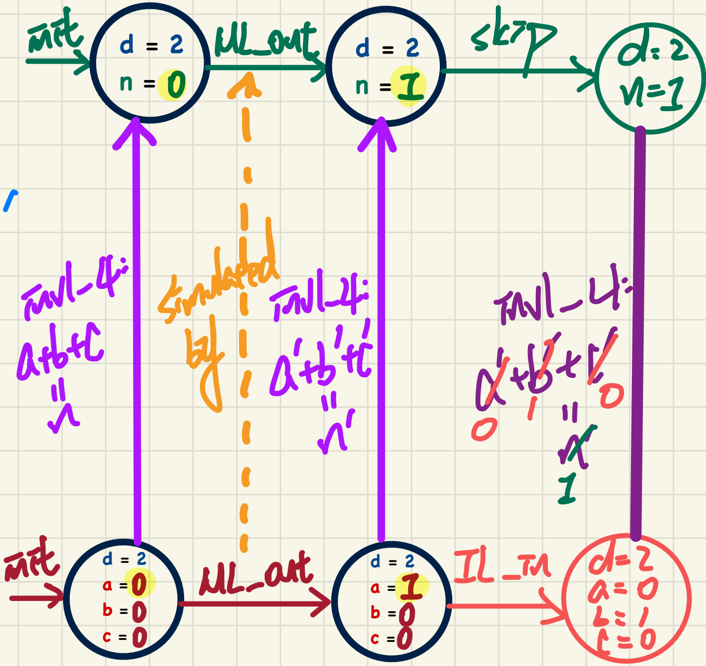
True

No. $n' = n$





trace: <init, ML_out, IL_in, ...>



old and new events

PO/VC Rule of Invariant Preservation: Sequents

Abstract m0

constants: d

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

variables: n

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

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

effect on abstract model

each concrete inv. in post-state

effect on concrete model

Concrete m1

variables: a, b, c

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

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

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

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

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

$2 * 5 = 10$

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

IL_in/INV1_4/INV

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

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

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

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

IL_in/INV1_5/INV

(exercise)

Discharging POs of m1: Invariant Preservation in Refinement

IL_in/inv1_4/INV

$d \in \mathbb{N}$

$d > 0$

$n \in \mathbb{N}$

$n \leq d$

$a \in \mathbb{N}$

$b \in \mathbb{N}$

$c \in \mathbb{N}$

$a + b + c = n$

$a = 0 \lor c = 0$

$a > 0$

$\vdash$

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

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

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

Discharging POs of m1: Invariant Preservation in Refinement

ML_in/inv1_5/INV

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

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

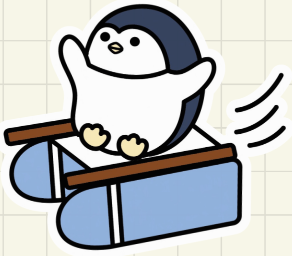
$$\frac{H \vdash Q}{H \vdash P \lor Q}\ \textbf{OR\_R2}$$

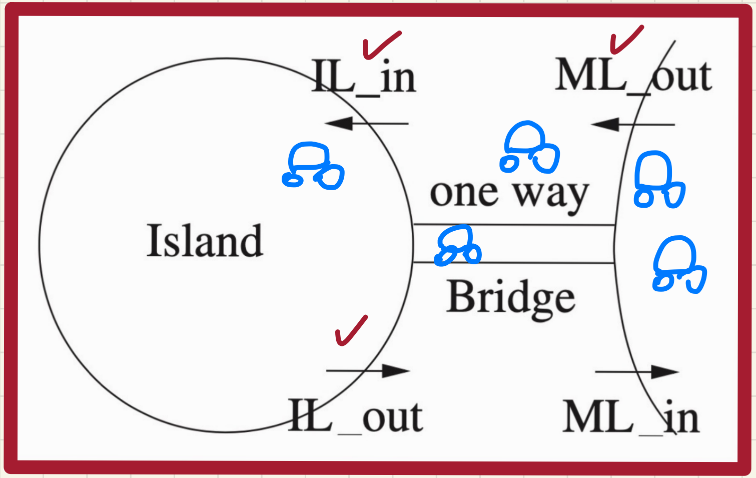
$$\frac{}{H, P \vdash P}\ \textbf{HYP}$$

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

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

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





Exercise Show the abstract & concrete transitions of:

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

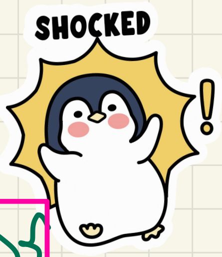
concrete transitions: <init, ML_out, IL_in, IL_out, ML_in>

Livelock Caused by New Events Diverging

- livelock
- divergence

sys. diverges

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



An alternative m1 (for demonstration)

constants: d

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

variables: a, b, c

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

old events

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

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

new events (always enabled)

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

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

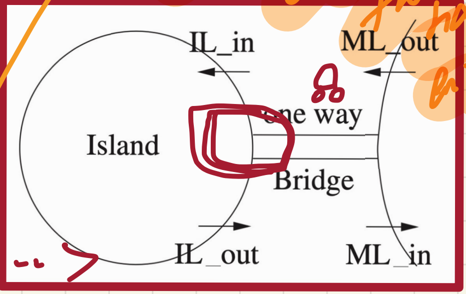
A possible scenario that's problematic:

Abstract Transitions: < init, ML_out, skip* >

Concrete Transitions: <init, ML_out, IL_in, IL_out, IL_in, IL_out, ... >

infinite interleaving of new events

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



Livelock / Divergence

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

→ (system) variant ($\in \mathbb{N}$)

~ not a solution to livelock

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

mak sure it's not ∞

two POs to discharge

→ if unprovable

↳ fix model

Lecture 20 - Nov 18

Bridge Controller

Invariant vs. Variant
Observing Patterns of Variant Values
POs of Variants: NAT vs. VAR

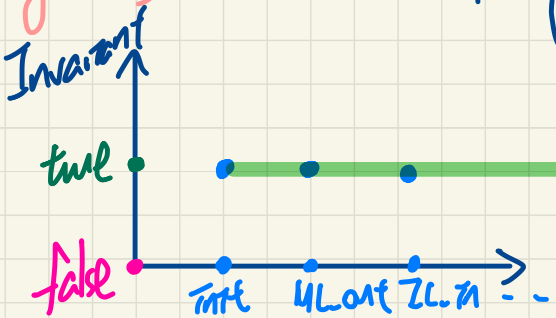
Announcements/Reminders

- Today's class: notes template posted
- **Lab4** released
- A reference paper for the **tabular method** (**Lab4**)
- Online course evaluation

Invariant vs. Variant

Invariant: Boolean expression that should always hold (after init and all event occurrences)

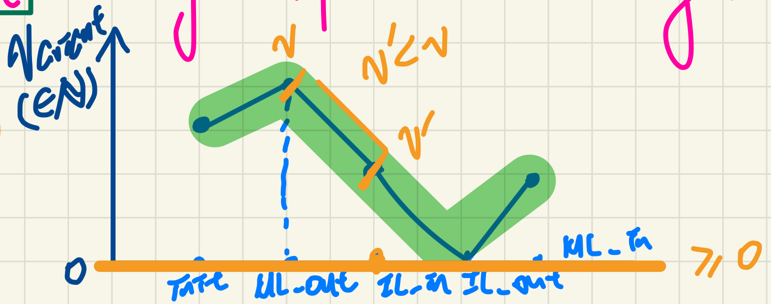
(system)



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

Variant : Integer expression that may change after event occurrences.

meant to measure # of interleavings of new events



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 : $a = 0 \lor c = 0$

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

```
ML_in
  when
    $c > 0$
  then
    $c := \underline{c - 1}$
  end
```

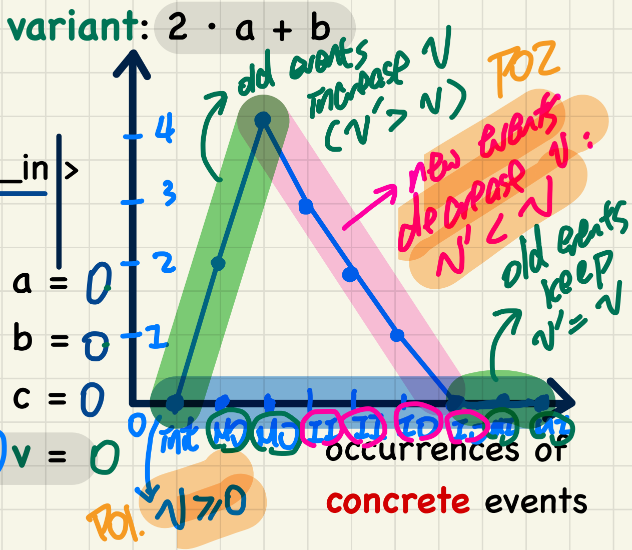
```
IL_in
  when
    $\boxed{a > 0}$
  then
    $a := \underline{a - 1}$
    $b := \underline{b + 1}$
  end
```

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

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

<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



PO of Convergence/Non-Divergence/Livelock Freedom

Variant Stays Non-Negative

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

actions

abs. Inv.

con. Inv.

guard of new event.

const.

con. vari.

variant

IL_in/NAT

new event

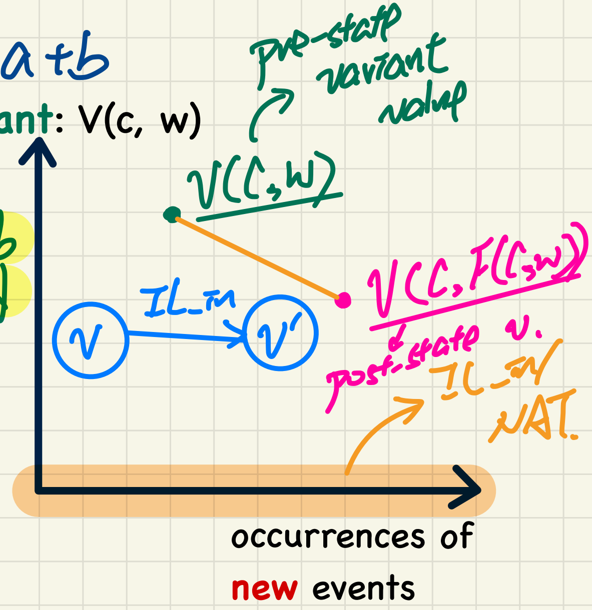
$v \geq 0$

$d \in \mathbb{N}$
$d > 0$
$n \in \mathbb{N}$ $a \in \mathbb{N}$ $b \in \mathbb{N}$ $c \in \mathbb{N}$ $a > 0$ $\vdash$ $2 \cdot a + b \in \mathbb{N}$
$n \leq d$ $a = 0 \vee c = 0$ $a + b + c = n$

Variants for New Events: 2 · a + b

* $v' < v$

$$2 \cdot \underset{a-1}{\cancel{a}} + \underset{b+1}{\cancel{b}} < 2 \cdot a + b$$



A New Event Occurrence Decreases Variant

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

effect of new event.

v'

v

IL_in/VAR

new event

v's value strictly dec.

$d \in \mathbb{N}$
$d > 0$
$n \in \mathbb{N}$ $a \in \mathbb{N}$ $b \in \mathbb{N}$ $c \in \mathbb{N}$ $a > 0$ $\vdash$ * $\boxed{2 \cdot (a-1) + (b+1) < 2 \cdot a + b}$
$n \leq d$ $a = 0 \vee c = 0$ $a + b + c = n$

Exercise Given variant: $a + b$

(1) Trace the value of v using the same trace.

Can the same patterns be observed?

(2) Formulate the VAR and NAT POs.

$(\underset{IL\_in,\ IL\_out}{\underline{2}} * \underset{NAT,\ VAR.}{\underline{2}}$ sequents to prove)

(3) Are they provable?

Example Inference Rules

$(H \wedge \neg P \Rightarrow Q) \overset{\equiv}{\Rightarrow} (H \Rightarrow P \vee Q)$

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

disjunction

right of ⊢

splitting a single hypothesis to two.

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

left of ⊢

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

right of ⊢

(compare: OR_L)

$$\boxed{H \Rightarrow P \vee Q}$$

$\equiv \{ P \equiv \neg\neg P \}$

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

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

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

$\equiv \{ \text{shunting} : x \Rightarrow (y \Rightarrow z) \equiv x \wedge y \Rightarrow z \}$

$$\boxed{H \wedge \neg P \Rightarrow Q}$$

$$\boxed{H \vdash P \wedge Q} \quad \text{AND\_R} \left\{ \begin{array}{l} \boxed{H \vdash P} \\ \boxed{H \vdash Q} \end{array} \right.$$

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

$$\begin{array}{l} \quad H \\ \vdash \\ \underbrace{(x \vee y)}_{P} \vee \underbrace{z}_{Q} \end{array} \quad \text{OR\_R} \quad \begin{array}{l} \quad H \\ \quad \neg(x \vee y) \\ \vdash \\ \quad z \end{array}$$

Lecture 21 - Nov 20

Bridge Controller

Relative Deadlock Freedom
m2: abstraction, superposition, invariant

Announcements/Reminders

- Today's class: notes template posted
- **Lab4** released
- A reference paper for the **tabular method** (**Lab4**)
- Online course evaluation

Idea of Relative Deadlock Freedom

proved for a refinement model.

Principles

* 1. DL is bad! (for reactive systems)

2. a refinement should not introduce a DL scenario not existing in the abstract M_0 model.

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

DL freedom cond. for M_0

DL freedom cond. for M_1.

disjunction of all abstract guards.

disjunction of all concrete guards

Guard Strengthening

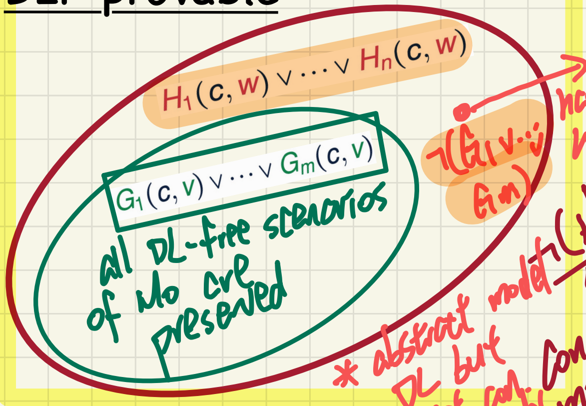
$$\boxed{\begin{array}{l} \vdots \\ H_c \\ \vdash \\ G_c \end{array}}$$

H_c: concrete guard of some event

G_c: abstract guard of that event.

* it's unacceptable if there's a state where the abstract model does not DL but the concrete model does.

DLF provable

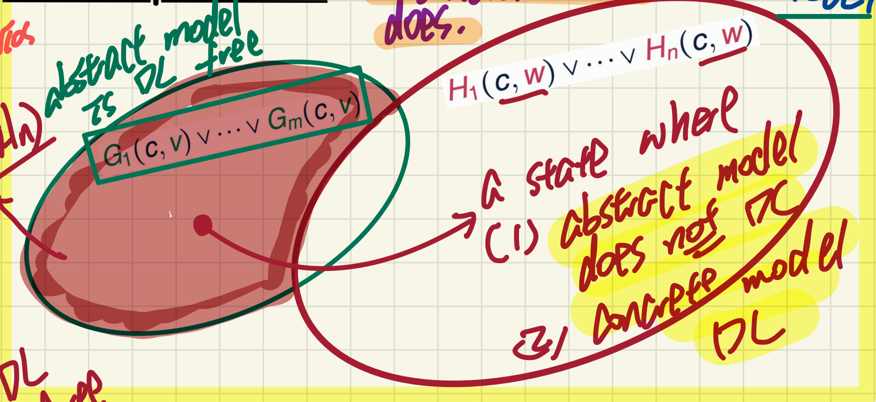


ok to have scenarios where *

* abstract model DL but not con. model

abstract model $\neg(H_1 \vee \cdots \vee H_n)$ ⇒ concrete model is not DL free

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
```

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

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

DL freedom of M0

$n < d$
$n > 0$

$\vdash$

DL freedom of M1

$(a + b < d$
$\land$
$c = 0)$
$\lor$
$(c > 0)$
$\lor$
$(a > 0)$
$\lor$
$(b > 0 \land a = 0)$

Discharging POs of m1: Relative Deadlock Freedom

Part 1

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

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

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

$$\begin{array}{ll}
\multicolumn{2}{l}{d \in \mathbb{N}} \\
\multicolumn{2}{l}{d > 0} \\
\multicolumn{2}{l}{n \in \mathbb{N}} \\
\multicolumn{2}{l}{n \leq d} \\
\multicolumn{2}{l}{a \in \mathbb{N}} \\
\multicolumn{2}{l}{b \in \mathbb{N}} \\
\multicolumn{2}{l}{c \in \mathbb{N}} \\
\multicolumn{2}{l}{a + b + c = n} \\
\multicolumn{2}{l}{a = 0 \lor c = 0} \\
\multicolumn{2}{l}{n < d \lor n > 0} \\
\multicolumn{2}{l}{\vdash} \\
 & a + b < d \land c = 0 \\
\lor & c > 0 \\
\lor & a > 0 \\
\lor & b > 0 \land a = 0
\end{array}$$

Discharging POs of m1: Relative Deadlock Freedom

Part 2

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

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

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

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

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

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

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

Initial Model and 1st Refinement: Provably Correct

Abstract m0

constants: d

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

variables: n

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

init
begin
$n := 0$
end

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

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

Concrete m1

constants: d

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

variables: a, b, c

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

variants:
$2 \cdot a + b$

2 POs:
VAR & NAT.

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

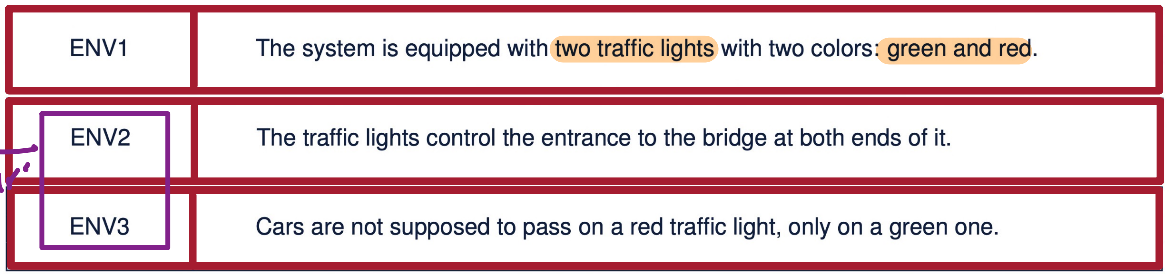
IL_in
when
$a > 0$
then
$a := a - 1$
$b := b + 1$
end

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

Correctness Criteria:

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

Bridge Controller: Abstraction in the 2nd Refinement

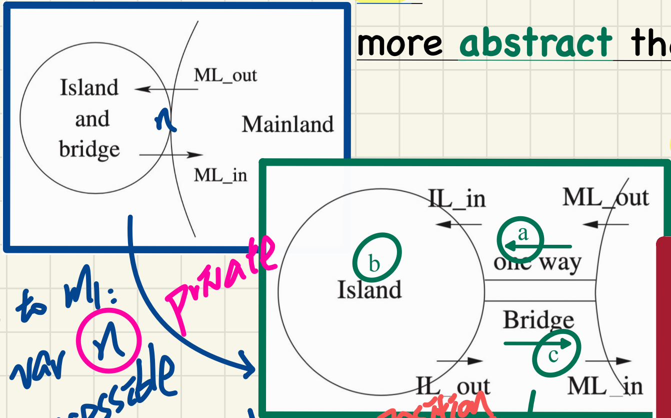


ENV1	The system is equipped with two traffic lights with two colors: green and red.
ENV2	The traffic lights control the entrance to the bridge at both ends of it.
ENV3	Cars are not supposed to pass on a red traffic light, only on a green one.

without these assumptions, very hard to predict the model behaviour.

m0:

more abstract than m1

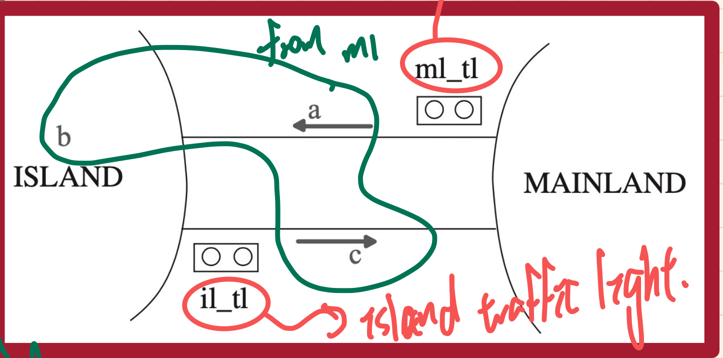


Correctness of refinement

1. guard strengthening
2. relative DLF
3. livelock / divergence
4. Inv. est. & preserved

m1:

more concrete than m0, more abstract than m2



mainland traffic light

island traffic light.

m2:

more concrete than m1

m0 to m1:

abs. var n private

not accessible by m1 (event guards & actions).

superposition

m1 to m2:

protected abs. var a, b, c accessible by con. model m2.

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.

Dynamic Part of Model

variables:

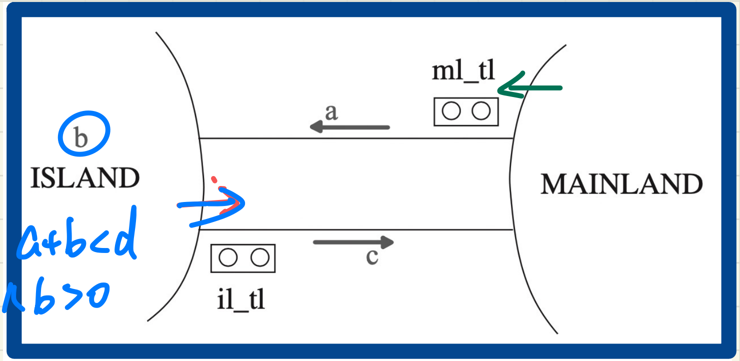
M_1 a, b, c

M_2 ml_tl
il_tl

invariants:

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

m_2



Static Part of Model

sets: $COLOR$

constants: $red, green$

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

Exercises

inv2_3: being allowed to exit ML $\Rightarrow$ means limited cars & no crash $c=0$

inv2_4: being allowed to exit IL $\Rightarrow$ means some car in IL & no crash $a=0$

Lecture 22 - Nov 25

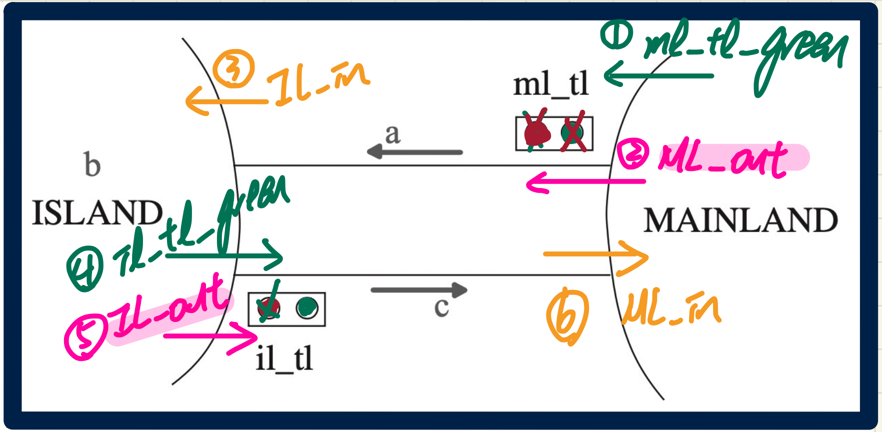
Bridge Controller

2nd Refinement: Splitting Guards
2nd Ref.: Unprovable Sequent for INV
Adding an Invariant

Announcements/Reminders

- Today's class: notes template posted
- Lab4 released
- A reference paper for the tabular method (Lab4)

Bridge Controller: "Old" and "New" Events



split ML_out in M1
to: ① ml_tl_green
② ML_out.

Single Car Travel:

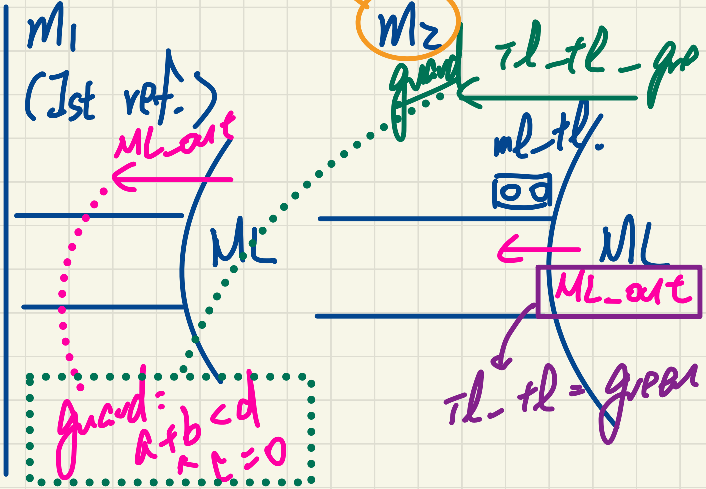
<init,
①ml_tl_green, ②ML_out,
③IL_in,
④il_tl_green, ⑤IL_out, ⑥ML_in>

driver's behaviour
↳ lack of tl.

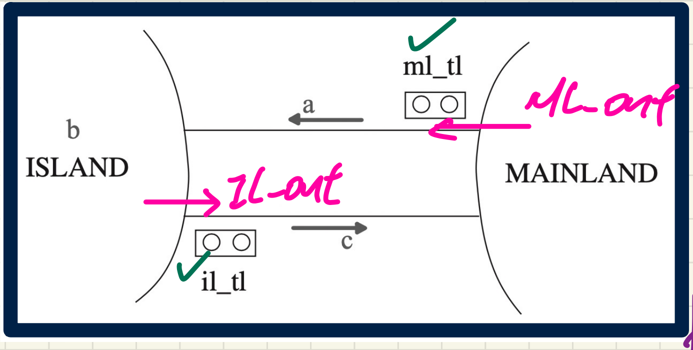
a, b, c are computer variables

↳ drivers have no access to their values

↳ ML_out
IL_out) drivers should only be concerned about traffic light colours.



Bridge Controller: **Guards** of "old" Events 2nd Refinement



sets: *COLOR*

constants: *red*, *green*

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

variables:
a, b, c
ml_tl
il_tl

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

In order for these events not to be enabled at the same time, il_tl and ml_tl cannot be green at the same time.

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

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

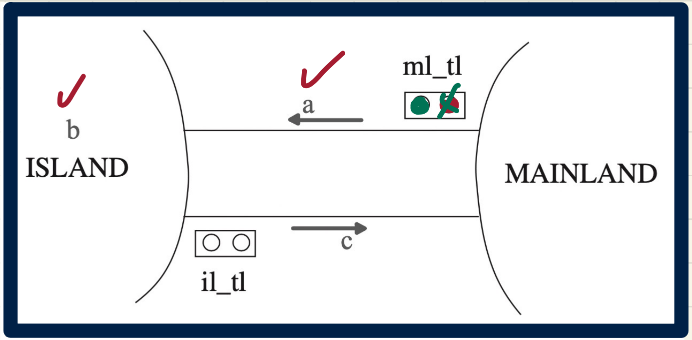
ml_tl = green.

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 : $green \neq red$

variables:
a, b, c
ml_tl
il_tl

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

ML_tl_green:

turn the traffic light **ml_tl** to green

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

(1) $ml\_tl = red$

(2) $a + b < d \land c = 0$

abstract guard of ML_out in M1

IL_tl_green:

turn the traffic light **il_tl** to green

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

(1) $il\_tl = red$

(2) $b > 0 \land a = 0$

abstract guard of IL_out in M1

PO/VC Rule of Invariant Preservation: Sequents

Abstract m1

variables: a, b, c

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

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

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

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

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
    $\boxed{a := a + 1}$
  end
```

$a' = a + 1$

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

ML_out/inv2_4/INV

axm0_1	$\{\ d \in \mathbb{N}$
axm0_2	$\{\ d > 0$
axm2_1	$\{\ COLOUR = \{green, red\}$
axm2_2	$\{\ green \neq red$
inv0_1	$\{\ n \in \mathbb{N}$
inv0_2	$\{\ n \leq d$
inv1_1	$\{\ a \in \mathbb{N}$
inv1_2	$\{\ b \in \mathbb{N}$
inv1_3	$\{\ c \in \mathbb{N}$
inv1_4	$\{\ a + b + c = n$
inv1_5	$\{\ a = 0 \lor c = 0$
inv2_1	$\{\ ml\_tl \in COLOUR$
inv2_2	$\{\ il\_tl \in COLOUR$
inv2_3	$\{\ ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4	$\{\ il\_tl = green \Rightarrow b > 0 \land a = 0$
Concrete guards of *ML_out*	$\{\ ml\_tl = green$
	$\vdash$
Concrete invariant **inv2_4** with *ML_out*'s effect in the post-state	$\{\ il\_tl = green \Rightarrow b > 0 \land \overset{a'}{\boxed{(a + 1)}} = 0$

Exercise: Specify IL_out/inv2_3/INV

Example Inference Rules

$$\frac{H, \boxed{P, Q} \vdash R}{H, \boxed{P, P \Rightarrow Q} \vdash R}$$ ✓ **IMP_L**

Modus ponens

$(P \wedge (P \Rightarrow q)) \Rightarrow q$

$$\frac{H, P \vdash Q}{H \vdash P \Rightarrow Q}$$ ✓ **IMP_R**

$H \wedge P \Rightarrow q$

$\equiv$

$H \Rightarrow (P \Rightarrow q)$

$$\frac{H, \boxed{\neg Q \vdash P}}{H, \boxed{\neg P \vdash Q}}$$ **NOT_L**

$\neg Q \Rightarrow P$

$\neg P \Rightarrow q$

Discharging POs of m2: Invariant Preservation

First Attempt

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

MON

ML_out/inv2_4/INV

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

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

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

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

Outstanding/Unprovable Sequent

green ≠ red ✓
ml_tl = green ✓
Il_tl = green
⊢
1 = 0

INV2_3:
↳ c = 0
INV2_4:
↳ a = 0

turns out that the current model allows a = 1 ∧ c = 1 when both lights are green.

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

IMP_R

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

IMP_L

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

AND_L

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

AND_R

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

HYP

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

EQ_LR, MON

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

ARI

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

??

Discharging POs of m2: Invariant Preservation

First Attempt

IL_out/inv2_3/INV

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

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

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

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

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

MON

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

IMP_R

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

IMP_L

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

AND_L

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

AND_R

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

MON

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

ARI

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

EQ_LR, MON

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

ARI

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

??

Fixed

Understanding the Failed Proof on INV

variables:
a, b, c
ml_tl
il_tl

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

IL_out/inv2_3/INV

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

ML_out/inv2_4/INV

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

there'll be multiple fixes, fix 1: add a new inv.

in this state
(1) il_tl is green
(2) inv2_4 entails a=0
(3) but a=1

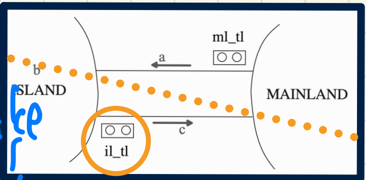
fake !!

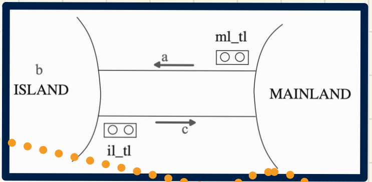
Unprovable Sequent:

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

should not be allowed.

$a = 0$





⟨ init	ML_tl_green	ML_out	IL_in	IL_tl_green	IL_out	ML_out ⟩
$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$
$a' = 0$	$a' = 0$	**a’ = 1**	**a’ = 0**	$a' = 0$	$a' = 0$	**a’ = 1**
$b' = 0$	$b' = 0$	$b' = 0$	**b’ = 1**	$b' = 1$	**b’ = 0**	$b' = 0$
$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	**c’ = 1**	$c' = 1$
$ml\_tl' = red$	**ml_tl’ = green**	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = green$	$ml\_tl' = green$
$il\_tl' = red$	$il\_tl' = red$	$il\_tl' = red$	$il\_tl' = red$	**il_tl’ = green**	$il\_tl' = green$	$il\_tl' = green$

Possible for both lights green.

a car heading to ML

∧ a=0

a car heading to IL

$$\neg(ml\_tl = green \wedge il\_tl = green)$$

$$\equiv \{ red \neq green \text{ (axiom)} \}$$

$$\underline{\neg}(ml\_tl\ \underline{\overset{\checkmark}{\neq}}\ red \wedge il\_tl \neq red)$$

$$\equiv \{ \text{de morgan} \}$$

$$\boxed{ml\_tl \overset{\checkmark}{=} red \vee il\_tl = red}$$

new invariant.

Fixing m2: Adding an Invariant

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

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

Abstract m1

variables: a, b, c

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

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

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

Concrete m2

variables:
a, b, c
ml_tl
il_tl

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

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

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

ML_out/inv2_4/INV

axm0_1	$d \in \mathbb{N}$
axm0_2	$d > 0$
axm2_1	$COLOUR = \{green, red\}$
axm2_2	$green \neq red$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
inv1_1	$a \in \mathbb{N}$
inv1_2	$b \in \mathbb{N}$
inv1_3	$c \in \mathbb{N}$
inv1_4	$a + b + c = n$
inv1_5	$a = 0 \lor c = 0$
inv2_1	$ml\_tl \in COLOUR$
inv2_2	$il\_tl \in COLOUR$
inv2_3	$ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4	$il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5	$ml\_tl = red \lor il\_tl = red$
Concrete guards of *ML_out*	$ml\_tl = green$

$\vdash$

Concrete invariant **inv2_4** with *ML_out*'s effect in the post-state	$il\_tl = green \Rightarrow b > 0 \land (a + 1) = 0$

new INV. has become a new hypothesis

Exercise: Specify IL_out/inv2_3/INV

Discharging POs of m2: Invariant Preservation

Second Attempt

ML_out/inv2_4/INV

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

MON

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

IMP_R

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

IMP_L

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

AND_L

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

AND_R

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

HYP

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

EQ_LR, MON

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

ARI

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

new inv. added

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

OR_L

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

Exercise.

EQ_LR, MON

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

Approach 1: NOT_L

Approach 2: $green = red$.

false hypothesis.

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

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

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

Discharging POs of m2: Invariant Preservation

Second Attempt

IL_out/inv2_3/INV

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

MON

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

IMP_R

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

IMP_L

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

AND_L

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

AND_R

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

MON

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

ARI

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

EQ_LR, MON

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

ARI

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

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

Assignment

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

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

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

Lecture 23 - Nov 27

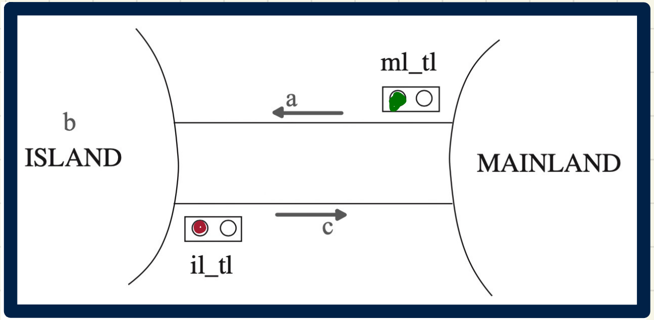
Bridge Controller

Adding Actions, Splitting Events
Preventing Livelock/Divergence
Proving Livelock/Divergence Freedom

Fixing m2: Adding Actions

Added inv2_5 : 1 * b

new inv.



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

ML_tl_green/inv2_5/INV

axm0_1	$\{ \; d \in \mathbb{N}$
axm0_2	$\{ \; d > 0$
axm2_1	$\{ \; COLOUR = \{green, red\}$
axm2_2	$\{ \; green \neq red$
inv0_1	$\{ \; n \in \mathbb{N}$
inv0_2	$\{ \; n \leq d$
inv1_1	$\{ \; a \in \mathbb{N}$
inv1_2	$\{ \; b \in \mathbb{N}$
inv1_3	$\{ \; c \in \mathbb{N}$
inv1_4	$\{ \; a + b + c = n$
inv1_5	$\{ \; a = 0 \lor c = 0$
inv2_1	$\{ \; ml\_tl \in COLOUR$
inv2_2	$\{ \; il\_tl \in COLOUR$
inv2_3	$\{ \; ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4	$\{ \; il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5	$\{ \; ml\_tl = red \lor il\_tl = red$

m0
m1
m2

ML_out
ML_in
IL_out
IL_in
ML_tl_green
IL_tl_green.

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

$ml\_tl' = g \land$
$il\_tl' = r$

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

Concrete grd. of ML_tl_gn. $\left\{ \begin{array}{l} ml\_tl = red \\ a + b < d \\ c = 0 \end{array} \right.$

$\vdash$

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

$\overset{green}{\cancel{ml\_tl'}} = red \lor \overset{red}{\cancel{il\_tl'}} = red$

Exercise: Specify IL_tl_green/inv2_5/INV

$inv2\_3 : ml\_tl = g \Rightarrow a+b<d \wedge \underline{c=0}$

$inv2\_4 : il\_tl = g \Rightarrow b>0 \wedge \underline{a=0}$

Discussed (Thursday)

$il\_tl = g \Rightarrow \boxed{a=0}$

ML_out / $\boxed{inv2\_4}$ / INV

IL_out / $\boxed{inv2\_3}$ / INV

$ml\_tl = g \Rightarrow \boxed{c=0}$

safety

To Discuss (Today)

$ml\_tl = g \Rightarrow \boxed{a+b<d}$

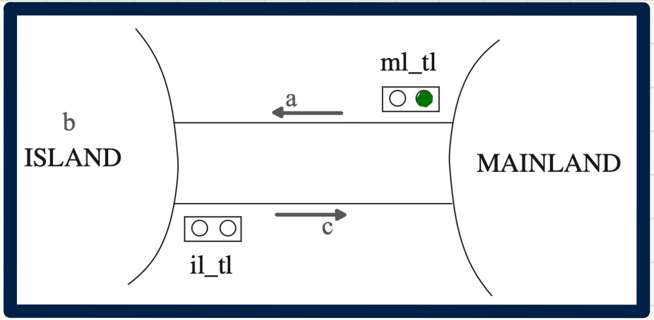
ML_out / $\boxed{inv2\_3}$ / INV

IL_out / $\boxed{inv2\_4}$ / INV

$il\_tl = g \Rightarrow \boxed{b>0}$

capacity.

Invariant Preservation: ML_out/inv2_3/INV



variables:
a, b, c
ml_tl
il_tl

ML_out
  when
    $ml\_tl = green$
  then
    $\boxed{a := a + 1}$
  end

$a' = a + 1$

IL_out
  when
    $il\_tl = green$ ✓
  then
    $\boxed{b := b - 1}$
    $c := c + 1$
  end

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

ML_out/inv2_3/INV

axm0_1 { $d \in \mathbb{N}$
axm0_2 { $d > 0$
axm2_1 { $COLOUR = \{green, red\}$
axm2_2 { $green \neq red$
inv0_1 { $n \in \mathbb{N}$
inv0_2 { $n \leq d$
inv1_1 { $a \in \mathbb{N}$
inv1_2 { $b \in \mathbb{N}$
inv1_3 { $c \in \mathbb{N}$
inv1_4 { $a + b + c = n$
inv1_5 { $a = 0 \lor c = 0$
inv2_1 { $ml\_tl \in COLOUR$
inv2_2 { $il\_tl \in COLOUR$
inv2_3 { $ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4 { $il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5 { $ml\_tl = red \lor il\_tl = red$
Concrete guards of *ML_out* { $ml\_tl = green$

$\vdash$

Concrete invariant **inv2_3** with *ML_out*'s effect in the post-state { $ml\_tl = green \Rightarrow \boxed{(a + 1) + b < d} \land c = 0$

post-state of ML_out ($a' = a + 1$)

$b' = b - 1$
$(b - 1) > 0$

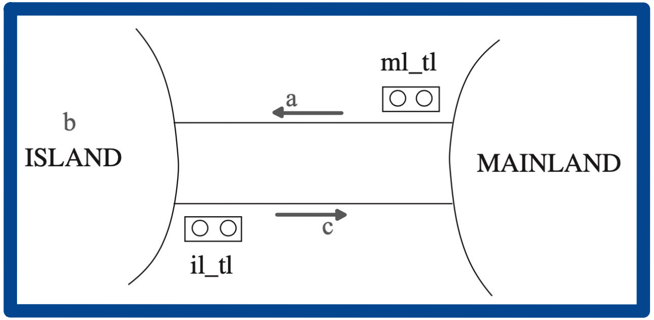
Exercise: Specify $\boxed{\text{IL\_out}}$/inv2_4/INV

Discharging POs of m2: Invariant Preservation

First Attempt

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

ML_out/inv2_3/INV



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

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

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

Unprovable

$a+b<d$
$c=0$
$ml\_tl=g.$
$\vdash (a+1)+b<d$

MON

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

IMP_R

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

IM~~P~~_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

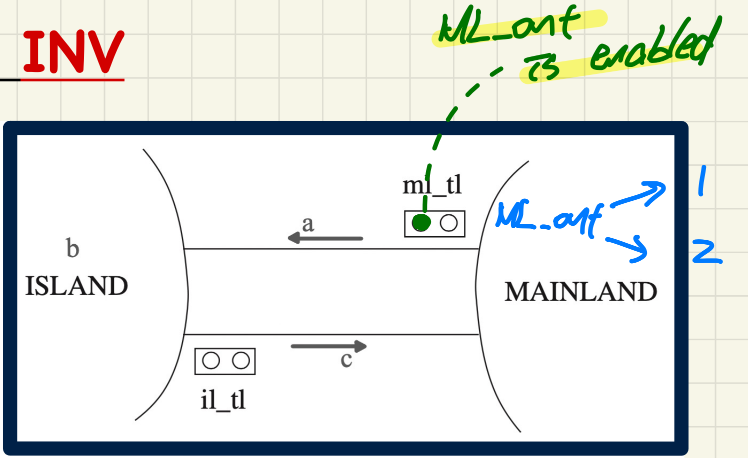
Understanding the Failed Proof on INV

variables:
a, b, c
ml_tl
il_tl

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



for $x+1<y$, x can't be equal to $y-1$!

$$\boxed{x < y} \overset{?}{\Rightarrow} x + 1 < y$$

not true in general

e.g. $3 < 4$

$3 + 1 \cancel{<} 4$

Unprovable Sequent:

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

ML_out allowing one new car on to the bridge

$d = 3, b = 0, a = 0$	[$(a + 1) + b < d$ 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.

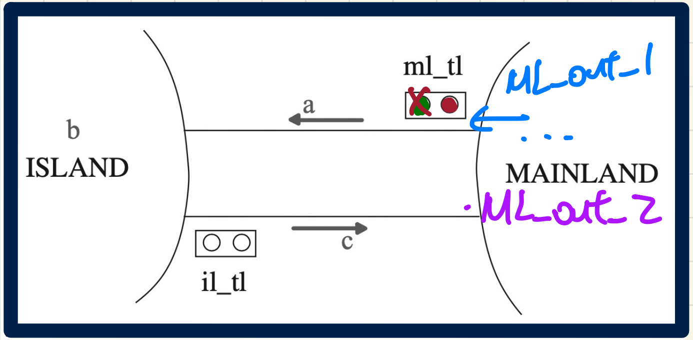
$(a+1)+b = d$

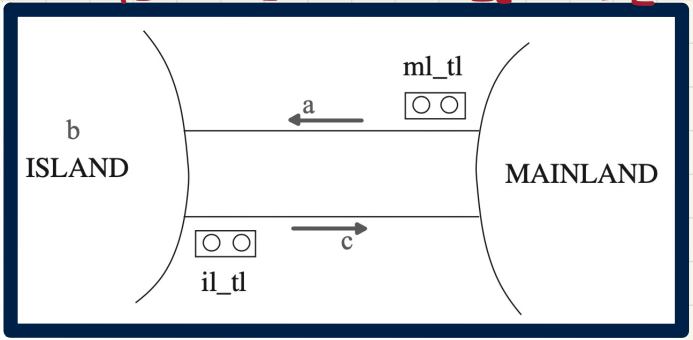
need to turn ml_tl to red right away

Fixing m2: Splitting Events

m1: ML_out → m2: ML_out_1, ML_out_2

IL_out → IL_out_1, IL_out_2



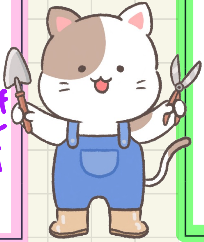


```
ML_out_1
  when
    $ml\_tl = green$
    $a + b + 1 \neq d$
  then
    $a := a + 1$
  end
```

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

disable ML_out_1

as soon as the capacity limit is reached, turn ml_tl to red



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

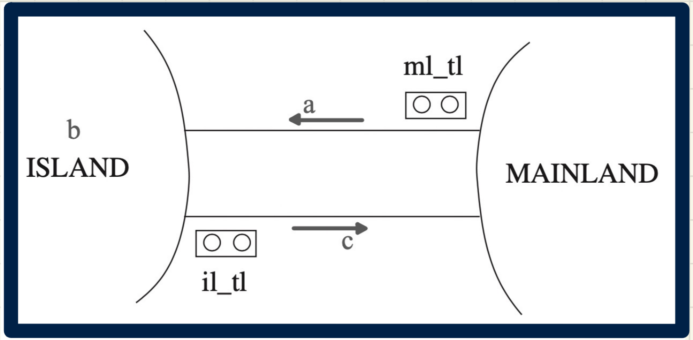
```
IL_out_2
  when
    $il\_tl = green$
    $b = 1$
  then
    $b := b - 1$
    $c := c + 1$
    $il\_tl := red$
  end
```

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

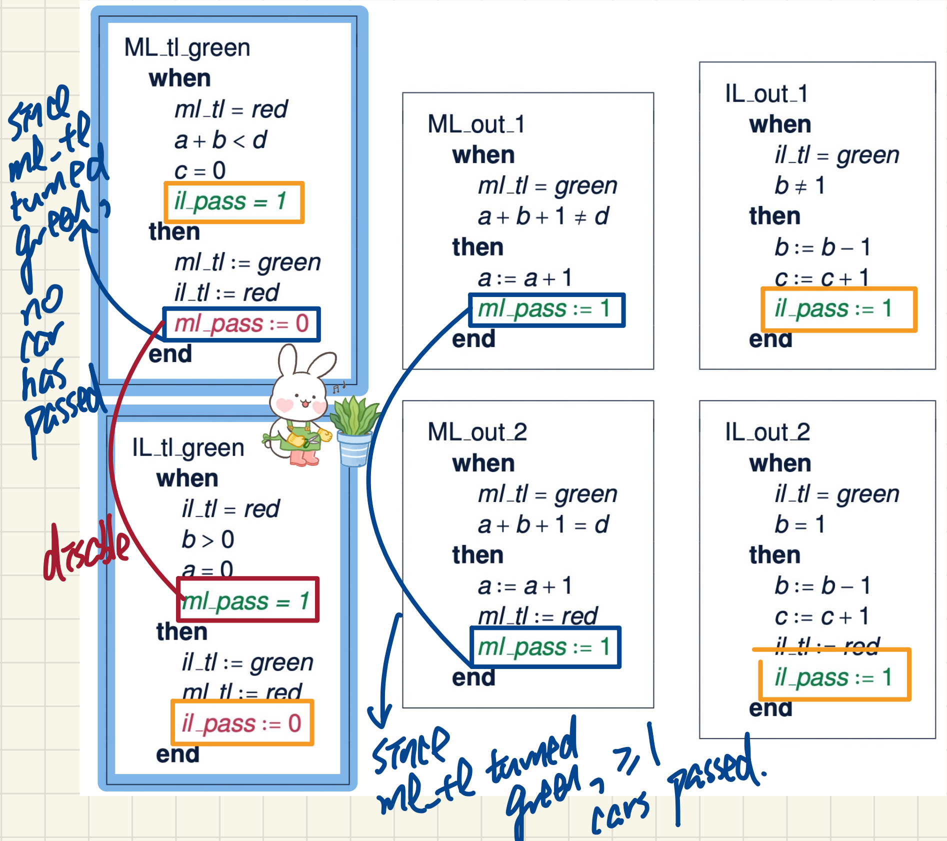
starting point of livelock

$\langle$ init, ML_tl_green, ML_out_1, IL_in, IL_tl_green, ML_tl_green, IL_tl_green, ... $\rangle$

init	ML_tl_green	ML_out_1	IL_in	IL_tl_green	ML_tl_green	IL_tl_green
$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$
$a' = 0$	$a' = 0$	$a' = 1$	$a' = 0$	$a' = 0$	$a' = 0$	$a' = 0$
$b' = 0$	$b' = 0$	$b' = 0$	$b' = 1$	$b' = 1$	$b' = 1$	$b' = 1$
$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$
ml_tl = red	ml_tl' = green	ml_tl' = green	ml_tl' = green	ml_tl' = red	ml_tl' = green	ml_tl' = red
il_tl = red	il_tl' = red	il_tl' = red	il_tl' = red	il_tl' = green	il_tl' = red	il_tl' = green

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, ...>



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

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

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

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

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

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

since ml-tl turned green, no car has passed

disable

since ml-tl turned green, ≥ 1 cars passed.

disabled IL-tl-green

both new events enabled.

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

Fixing **m2**: Measuring Traffic Light Changes

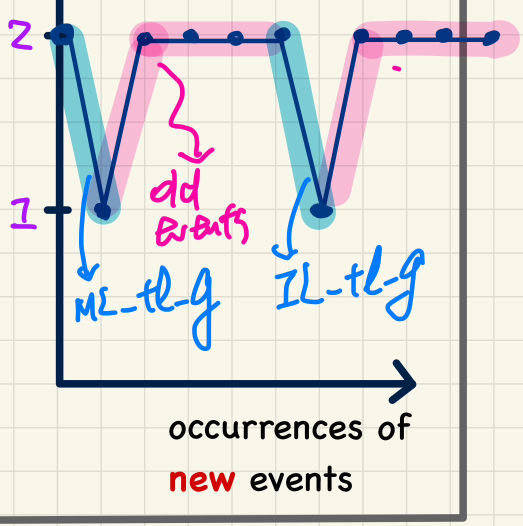
```
ML_tl_green
  when
    ml_tl = red
    a + b < d
    c = 0
    il_pass = 1
  then
    ml_tl := green
    il_tl := red
    ml_pass := 0
  end
```

```
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*

variant: V(c, w)



PO of Convergence/Non-Divergence/Livelock Freedom

A New Event Occurrence Decreases Variant

$A(c)$
$I(c, v)$
$J(c, v, w)$
$H(c, w)$
$\vdash$ ✓
$\boxed{V(c, F(c, w))} < \boxed{V(c, w)}$

post-state value of var.

pre-state value of var.

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
```

$ml\_pass' = 0$

$il\_pass' = il\_pass$

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

* $\overset{0}{\cancel{ml\_pass'}} + \overset{il\_pass}{\cancel{il\_pass'}} < ml\_pass + il\_pass$

PO of Relative Deadlock Freedom

axm0_1 $d \in \mathbb{N}$
axm0_2 $d > 0$
axm2_1 $COLOUR = \{green, red\}$
axm2_2 $green \neq red$
inv0_1 $n \in \mathbb{N}$
inv0_2 $n \leq d$
inv1_1 $a \in \mathbb{N}$
inv1_2 $b \in \mathbb{N}$
inv1_3 $c \in \mathbb{N}$
inv1_4 $a + b + c = n$
inv1_5 $a = 0 \lor c = 0$
inv2_1 $ml\_tl \in COLOUR$
inv2_2 $il\_tl \in COLOUR$
inv2_3 $ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4 $il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5 $ml\_tl = red \lor il\_tl = red$
inv2_6 $ml\_pass \in \{0, 1\}$
inv2_7 $il\_pass \in \{0, 1\}$
inv2_8 $ml\_tl = red \Rightarrow ml\_pass = 1$
inv2_9 $il\_tl = red \Rightarrow il\_pass = 1$

Disjunction of ***abstract*** guards

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

$\vdash$

Disjunction of ***concrete*** guards

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

Abstract m1

variables: a, b, c

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

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

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

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

Concrete m2

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

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

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

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

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

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

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

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

Discharging POs of m2: Relative Deadlock Freedom

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

⋮

IS#1

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

…

Study

IS#2

$$\begin{array}{ll}
 & d \in \mathbb{N} \\
 & d > 0 \\
 & b \in \mathbb{N} \\
 & ml\_tl = red \\
 & il\_tl = red \\
 & ml\_pass = 1 \\
 & il\_pass = 1 \\
\vdash & \\
 & \quad b < d \land ml\_pass = 1 \land il\_pass = 1 \\
 \lor & \quad b > 0 \land ml\_pass = 1 \land il\_pass = 1
\end{array}$$

…

IS#3

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

ARI

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

OR_L

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

OR_R2

$$\begin{array}{l}
d > 0 \\
b > 0 \\
\vdash \\
b > 0
\end{array}$$

HYP

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

EQ_LR, MON

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

OR_R1

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

HYP

1st Refinement and 2nd Refinement: Provably Correct

Abstract m1

variables: a, b, c

constants: d

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

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

variants:
$2 \cdot a + b$

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

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

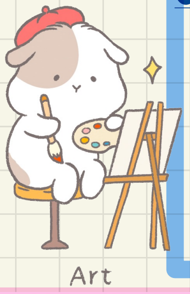
ML_in
when
$c > 0$
then
$c := c - 1$
end

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

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

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 : $ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4 : $il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5 : $ml\_tl = red \lor il\_tl = red$
inv2_6 : $ml\_pass \in \{0, 1\}$
inv2_7 : $il\_pass \in \{0, 1\}$
inv2_8 : $ml\_tl = red \Rightarrow ml\_pass = 1$
inv2_9 : $il\_tl = red \Rightarrow il\_pass = 1$

variants:
$ml\_pass + il\_pass$

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

Lecture 24 - Dec 2

Background

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

Announcements/Reminders

- Today's class: notes template posted
- **Lab4** released
- A reference paper for the **tabular method** (**Lab4**)
- **Review session** survey active now!
- **Exam guide**, example questions released

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

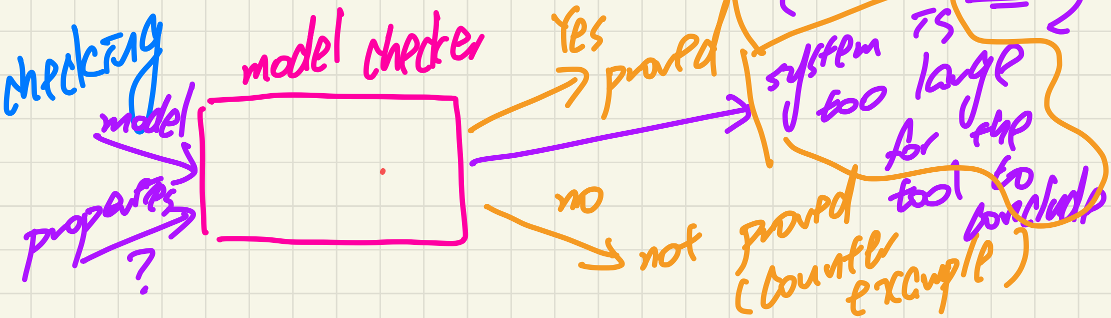
⟹ properties?

→ 4315

Mission-Critical system

model checking

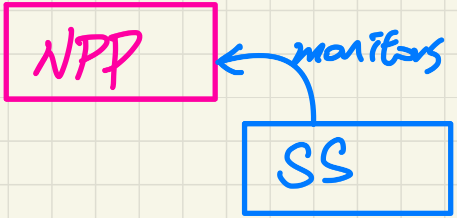
state explosion



Safety-Critical System

1. nuclear power plant

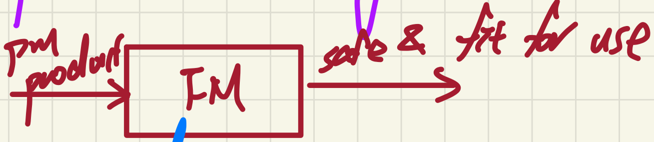
+ nuclear shutdown system



2. radiation

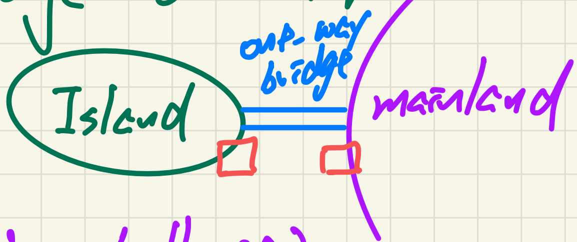
3. "glove"

4. pacemaker (pacemaker challenge)



5. auto-pilot & auto-driving.

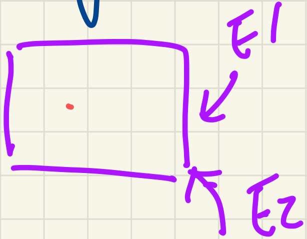
6. bridge controller



Acceptance Criteria

① Req. precise

↳ no ambiguities, no contradiction



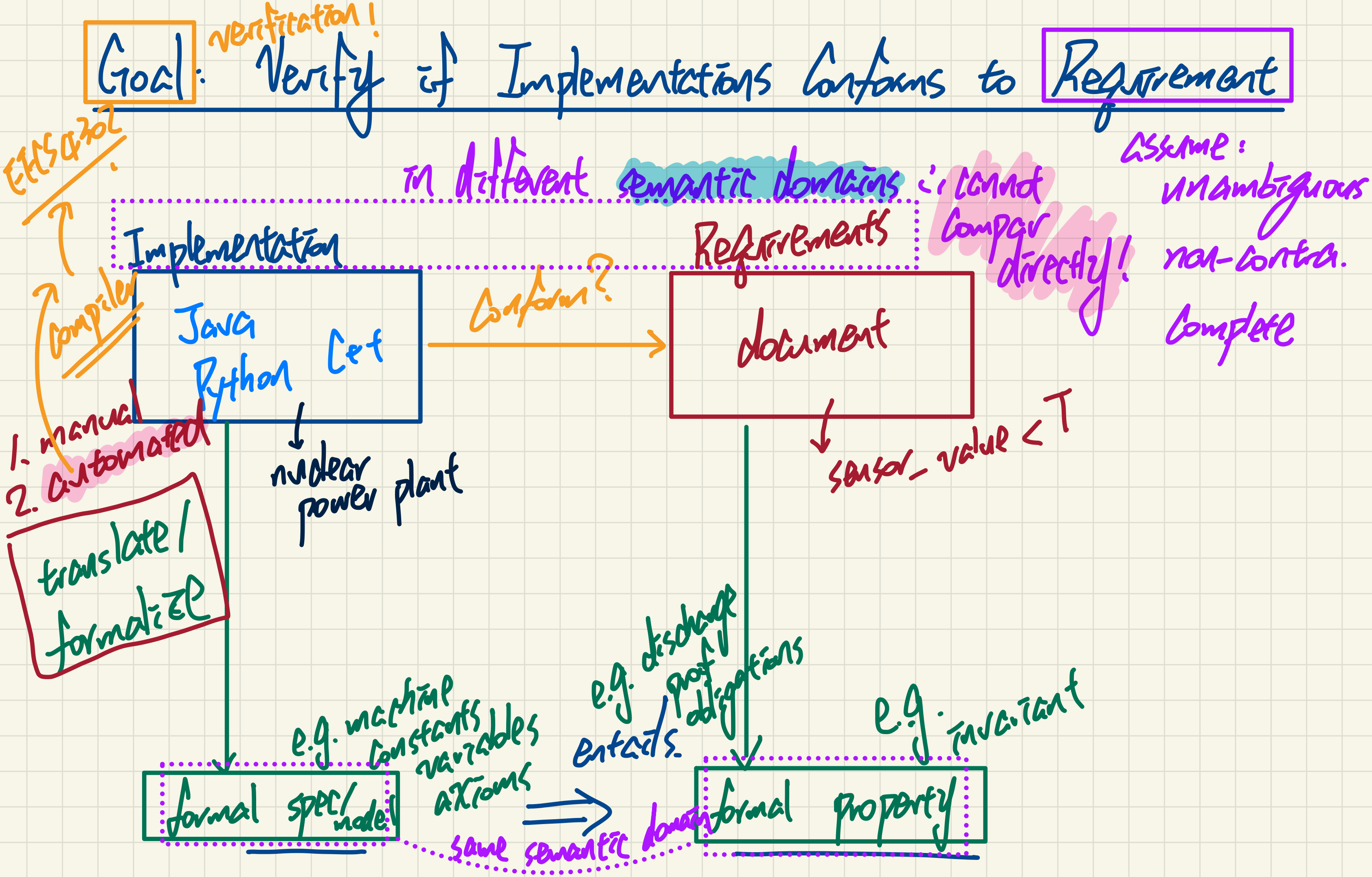
P
$\neg P$ ⇒ false

↳ complete

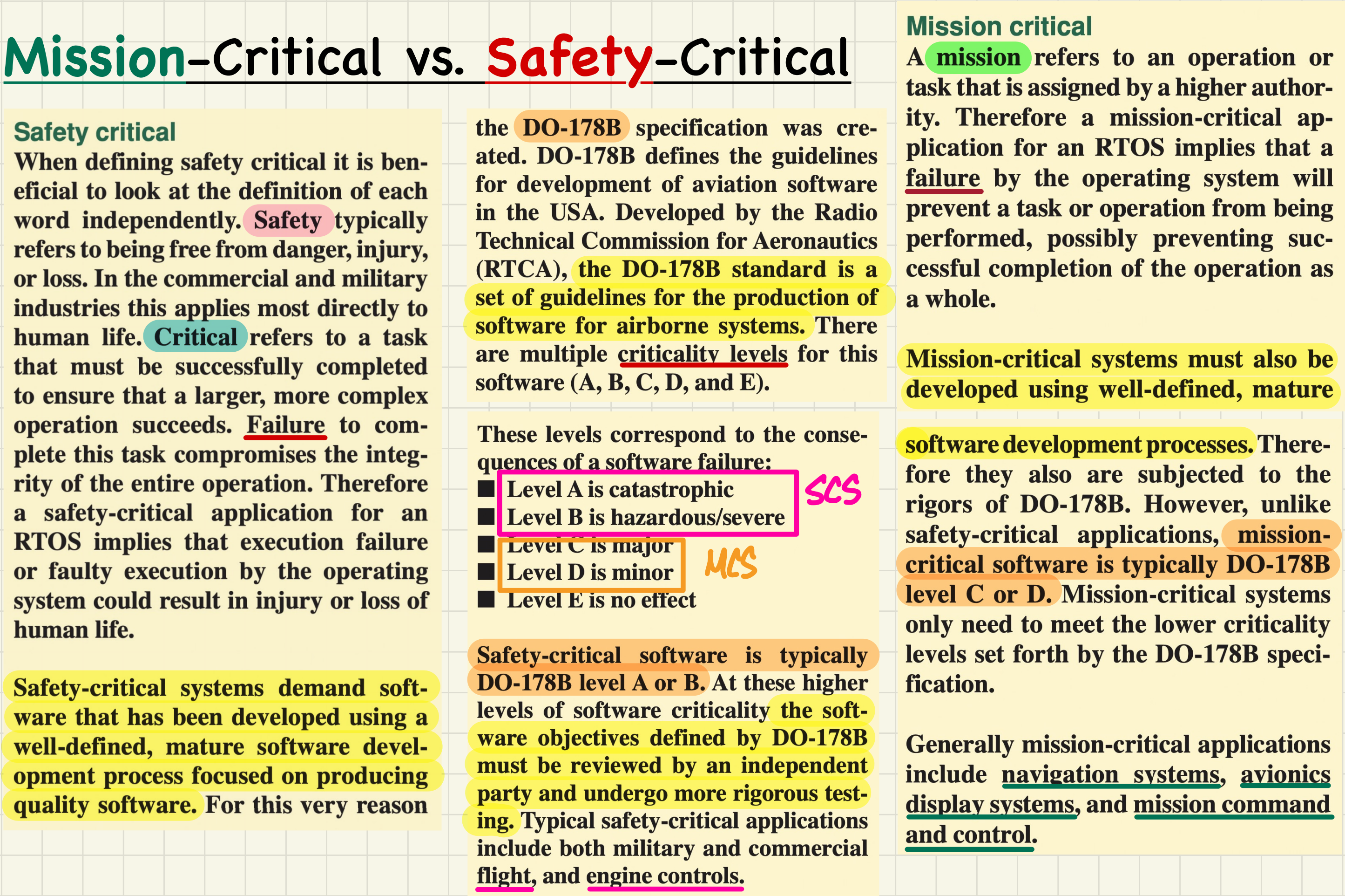
↳ no missing scenarios

Labs

Goal: Verify if Implementations Conforms to Requirement



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

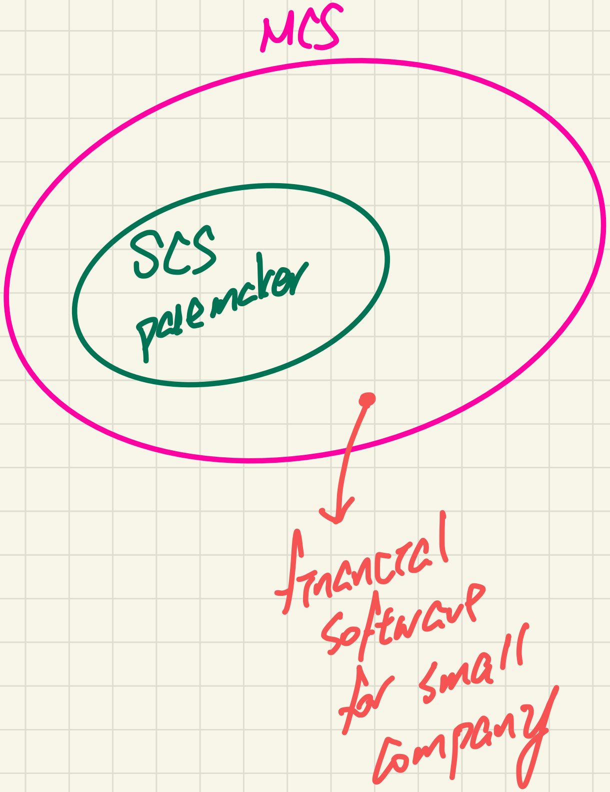
(1) ✗ SCS $\Leftrightarrow$ MCS

(2) ✓ SCS $\Rightarrow$ MCS

(3) ✗ MCS $\Rightarrow$ SCS

SCS $\subseteq$ MCS

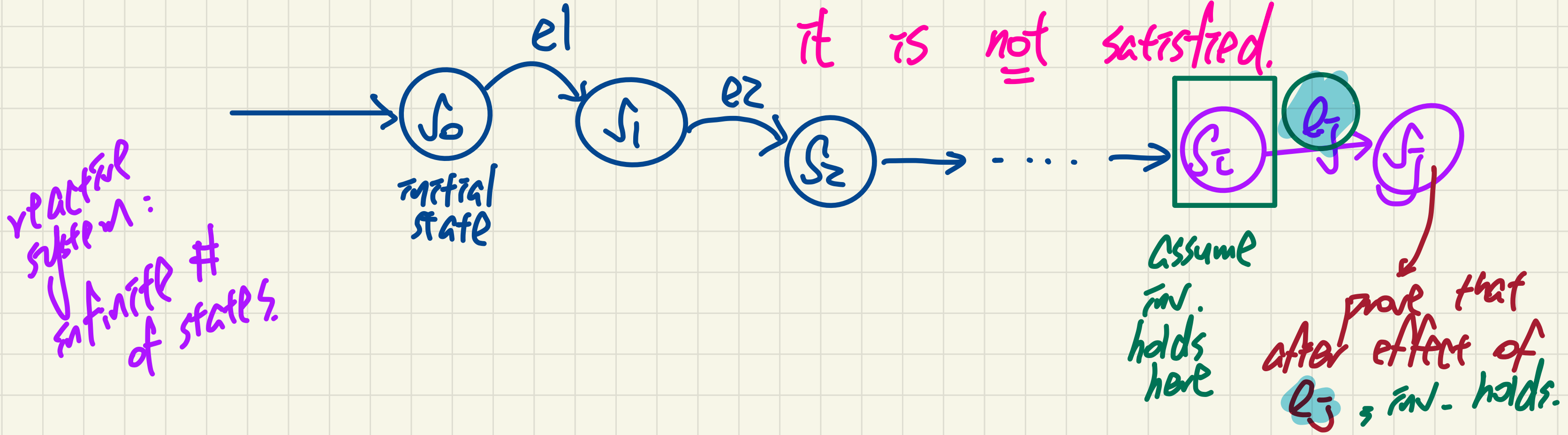
$\Rightarrow$



Safety Property / Invariant

↳ Every possible state of the system should satisfy it.

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



assumption: req. already given

Verification : Are we building the product right?

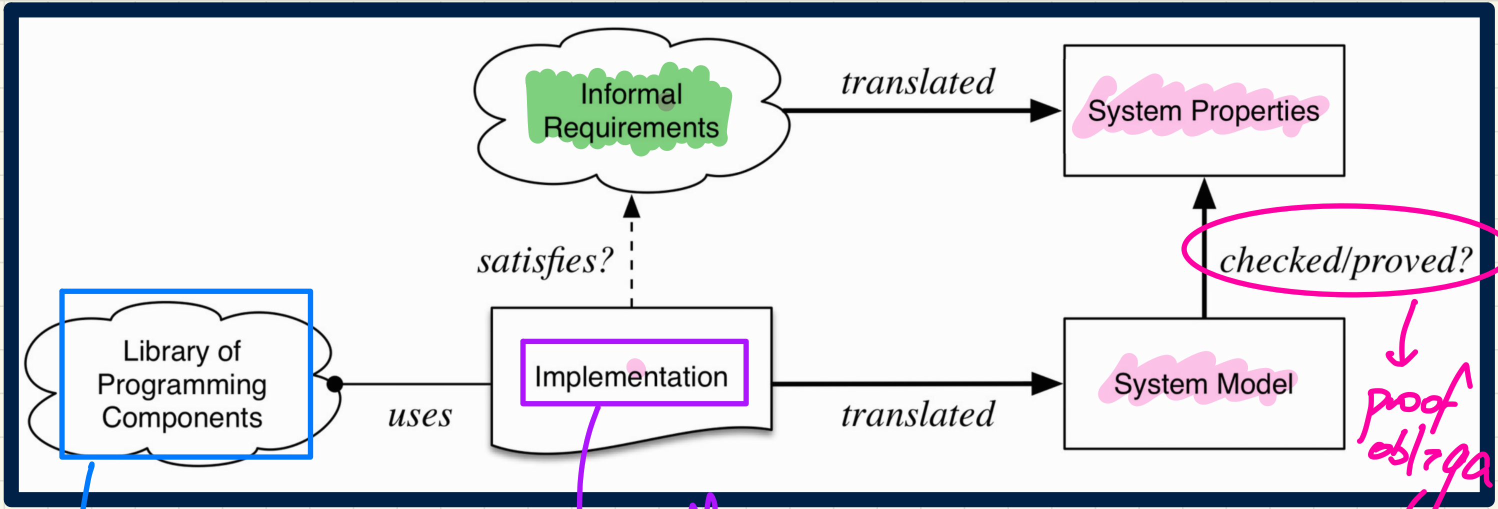
process of construction

4312.

Validation : Are we building the right product?

Are the req. given truly intended by customers?

Building the product right?

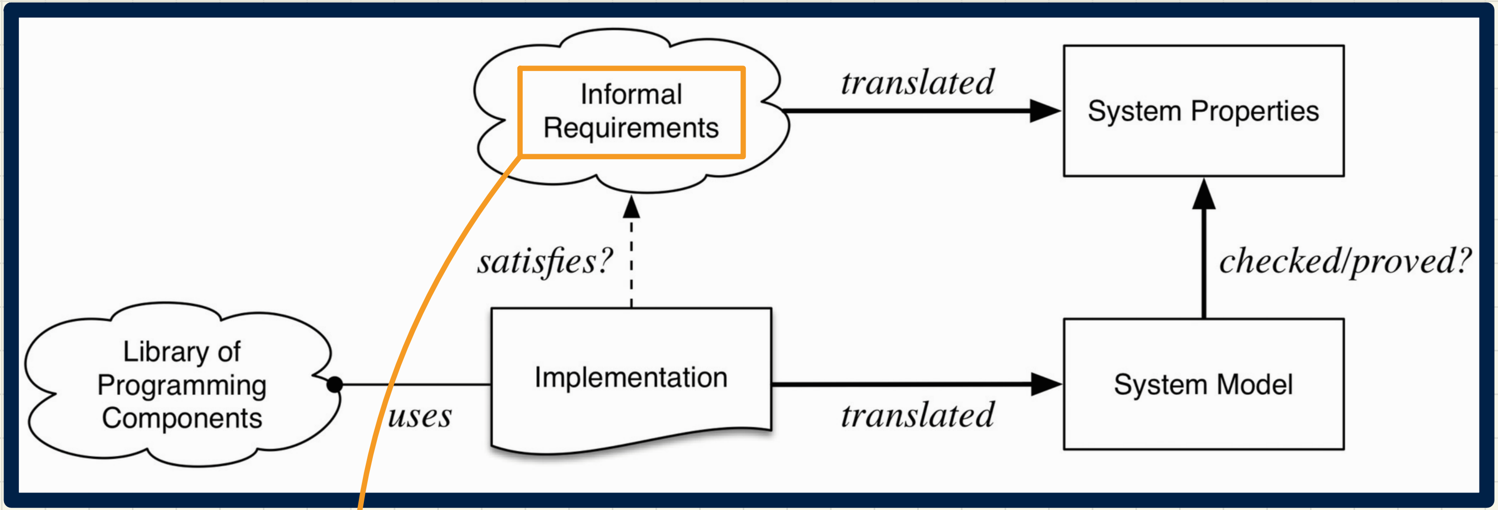


proof obligations.

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?

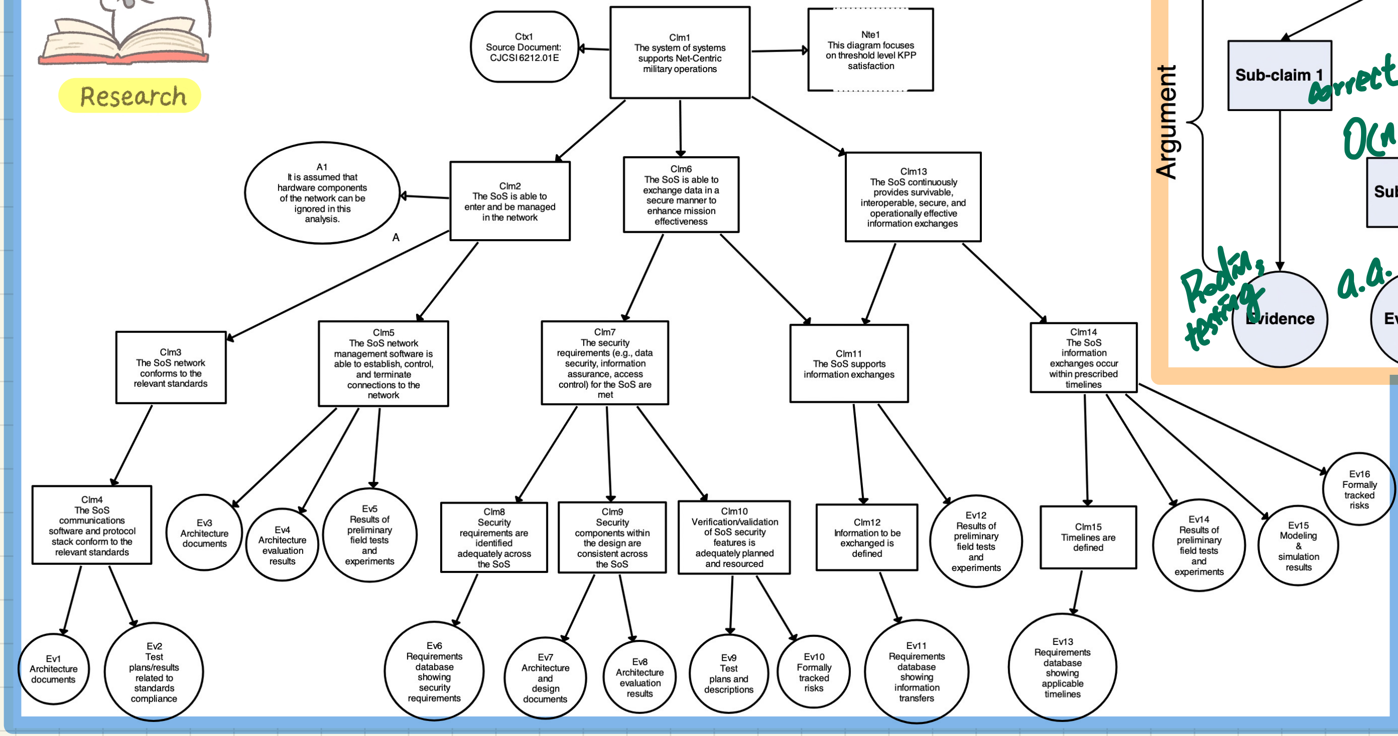


for validation,
critical on
what's given.

Certifying Systems: Assurance Cases

Research on "Assurance Cases" if interested!

Research



Ctx1
Source Document: CJCSI 6212.01E

Clm1
The system of systems supports Net-Centric military operations

Nte1
This diagram focuses on threshold level KPP satisfaction

A1
It is assumed that hardware components of the network can be ignored in this analysis.

A

Clm2
The SoS is able to enter and be managed in the network

Clm6
The SoS is able to exchange data in a secure manner to enhance mission effectiveness

Clm13
The SoS continuously provides survivable, interoperable, secure, and operationally effective information exchanges

Clm3
The SoS network conforms to the relevant standards

Clm5
The SoS network management software is able to establish, control, and terminate connections to the network

Clm7
The security requirements (e.g., data security, information assurance, access control) for the SoS are met

Clm11
The SoS supports information exchanges

Clm14
The SoS information exchanges occur within prescribed timelines

Clm4
The SoS communications software and protocol stack conform to the relevant standards

Ev3
Architecture documents

Ev4
Architecture evaluation results

Ev5
Results of preliminary field tests and experiments

Clm8
Security requirements are identified adequately across the SoS

Clm9
Security components within the design are consistent across the SoS

Clm10
Verification/validation of SoS security features is adequately planned and resourced

Clm12
Information to be exchanged is defined

Ev12
Results of preliminary field tests and experiments

Clm15
Timelines are defined

Ev14
Results of preliminary field tests and experiments

Ev15
Modeling & simulation results

Ev16
Formally tracked risks

Ev1
Architecture documents

Ev2
Test plans/results related to standards compliance

Ev6
Requirements database showing security requirements

Ev7
Architecture and design documents

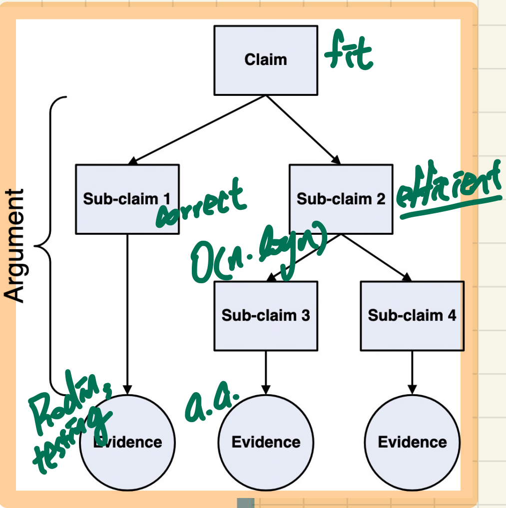
Ev8
Architecture evaluation results

Ev9
Test plans and descriptions

Ev10
Formally tracked risks

Ev11
Requirements database showing information transfers

Ev13
Requirements database showing applicable timelines

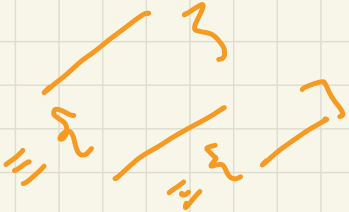


Source: https://resources.sei.cmu.edu/asset_files/whitepaper/2009_019_001_29066.pdf

Exam Info

- q. booklet (sketch)
- ans. booklet (no sketch)

- When: 9am to 12pm, Thursday, December 11 (ACW 206)
- Coverage: **Everything** (lecture materials & labs) ∀ !
  - + slides, iPad notes
- Even problems that look **challenging** at first are built on the **same** **foundational techniques** you've learned and practiced in **lectures** and **labs**. A solid, reflective grasp of the basics will take you **farther** than memorizing examples.
- Format: Mostly Written
  - + explanations/justifications + ASCII, math write math expressions + calculations, proofs
    - ↓
    - sequent proofs
    - predicate logic.
- Restrictions:
  - + **One-sided**, hand-written computer-typed, **min** 10pt data sheet ART
  - + No sketch paper (Exam booklet includes it) + **No** calculator
- What you should bring:
  - + Valid, Physical Photo ID (strict)
  - + Water/Snack



I hope you enjoyed learning with me

All the best to you!