EECS3342 System Specification and Refinement

Lecture Notes

Winter 2023

Jackie Wang

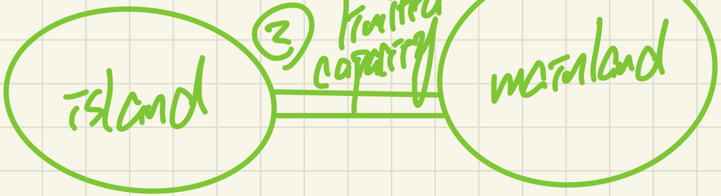
Lecture 1 - January 10

Syllabus & Introduction

Safety-Critical Systems
Code of Ethics of a Professional Engineer
Developing Safety-Critical Systems

Safety - Critical Systems

1. nuclear power plants (e.g. Darlington).
2. auto driving.
3. pacemaker (pacemaker challenge McMaster)
4. bridge controller ① one-way ② limited capacity



EECS 4312

Precise.

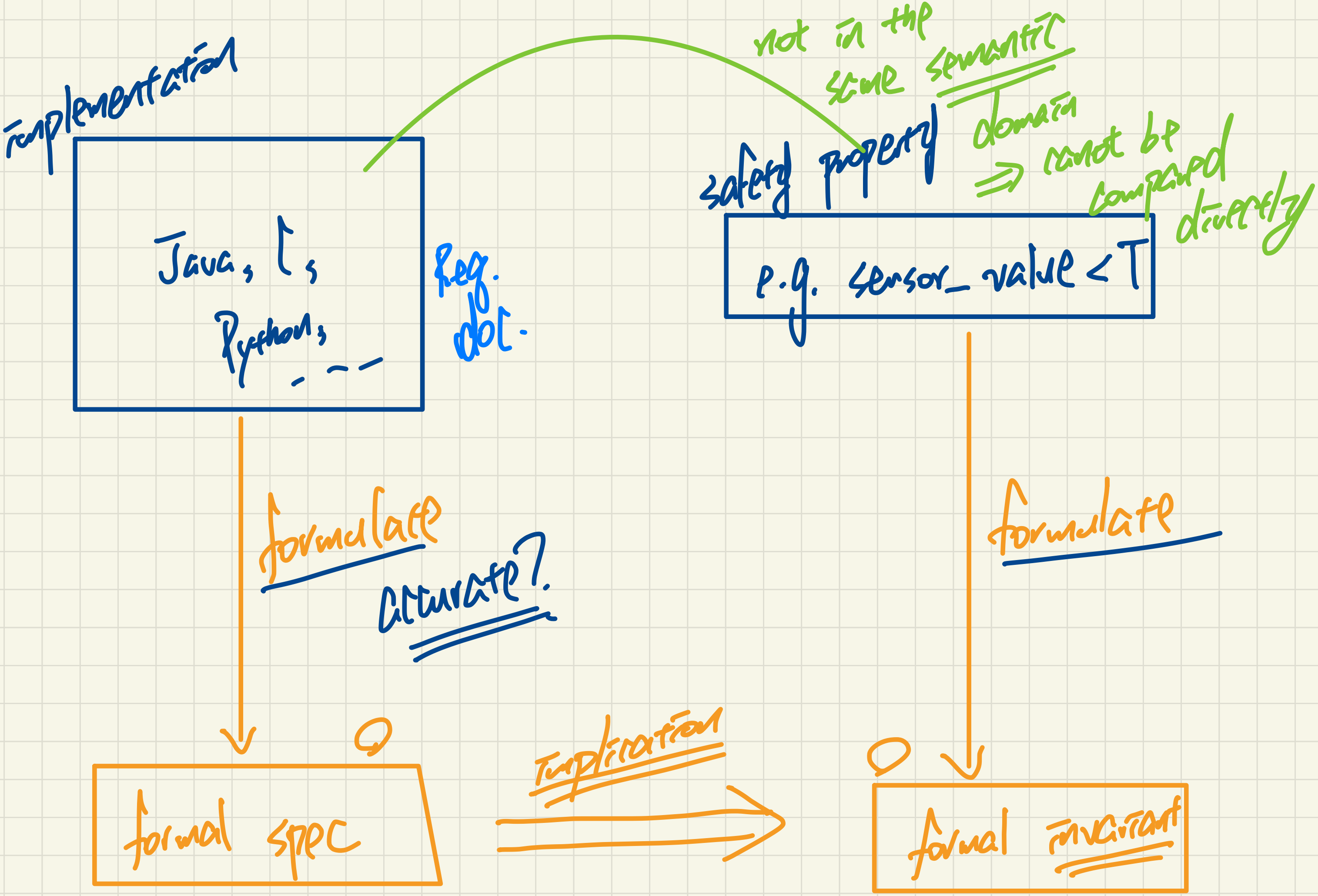
↳ no scope of multiple interpretation

↳ math!

Complete

↳ no missing scenarios.

↳ all possible "executions" of the system exhibit the safety property.


implementation
Java, C,
Python, ...
Req.
doc.
formulate
accurate?
formal spec
implication
safety property
e.g. sensor_value < T
not in the same semantic domain
⇒ cannot be compared directly
formulate
formal invariant

Lecture 2 - January 12

Introduction

Safety-Critical vs. Mission-Critical
Formal Methods, Industrial Standards
Verification vs. Validation
Model-Based Development

Consequence : SCS MCS

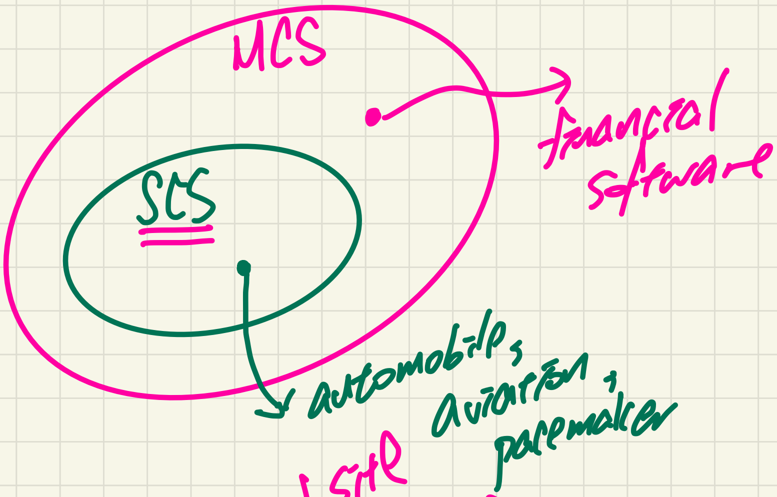
~~(1)~~ SCS $\overset{\Leftrightarrow}{\equiv}$ MCS

(2) SCS $\Rightarrow$ MCS

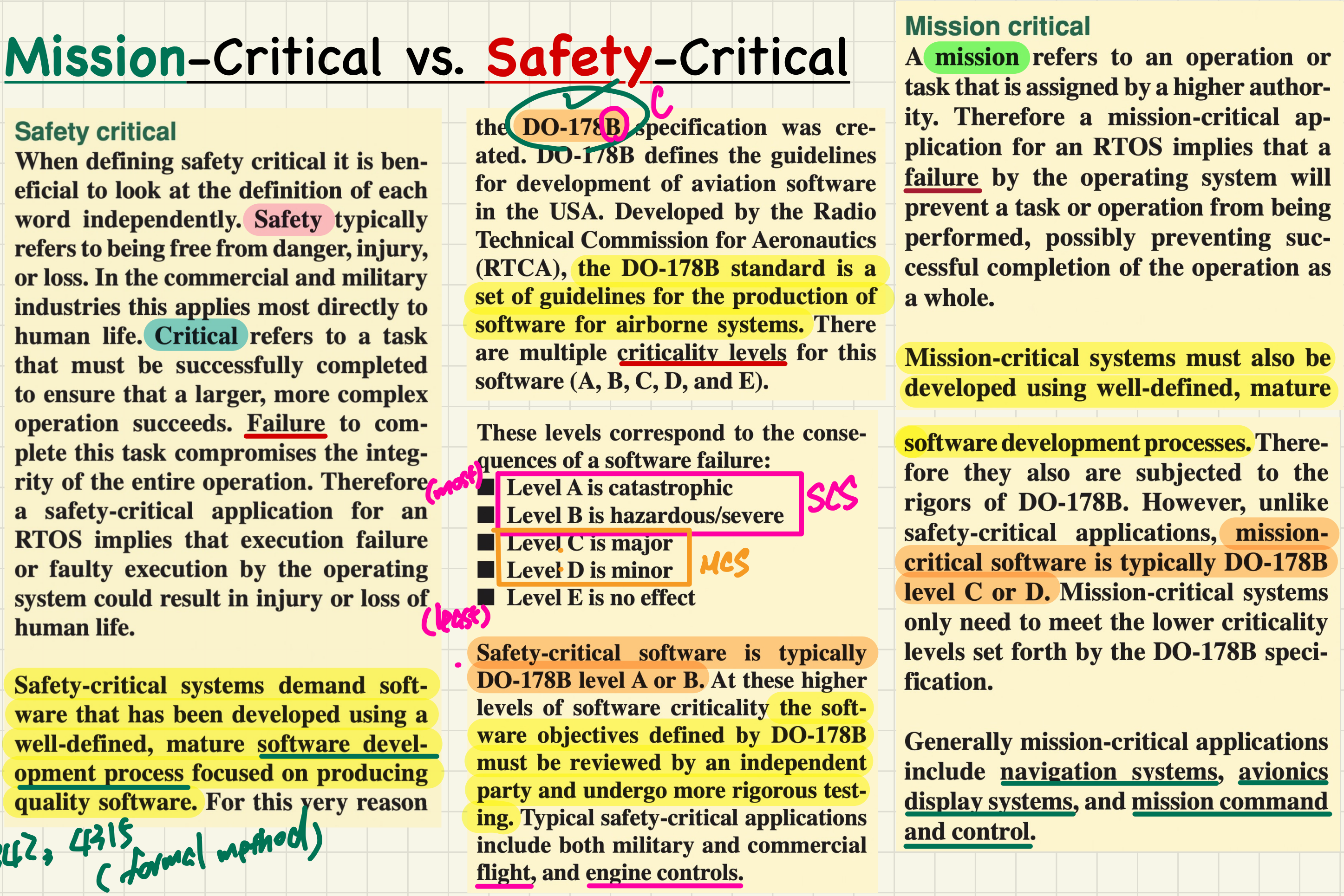
(3) MCS $\overset{\times}{\Rightarrow}$ SCS → shopping website (financial software

MCS → mission

SCS → safety



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).

C

2342, 4315 (formal method)

These levels correspond to the consequences of a software failure:

(most)

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

SCS

MCS

(least)

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.

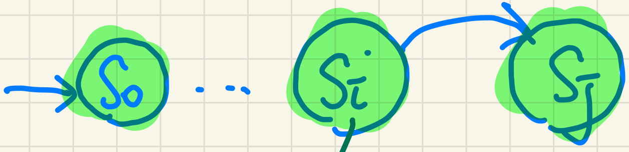
Source: http://pdf.cloud.opensystemsmedia.com/advancedtca-systems.com/SBS.Jan04.pdf

Safety Property → predicates

Invariant property.

└→ Every possible state of the system should satisfy it.

reactive system is running all the time (infinite # of state)



- assume S_i safe
- prove S_j safe. (S_{i+1})

process

- Are we buiding the poduct right?

not right e.g. without testing

property?

Implicit assumption: given what to build

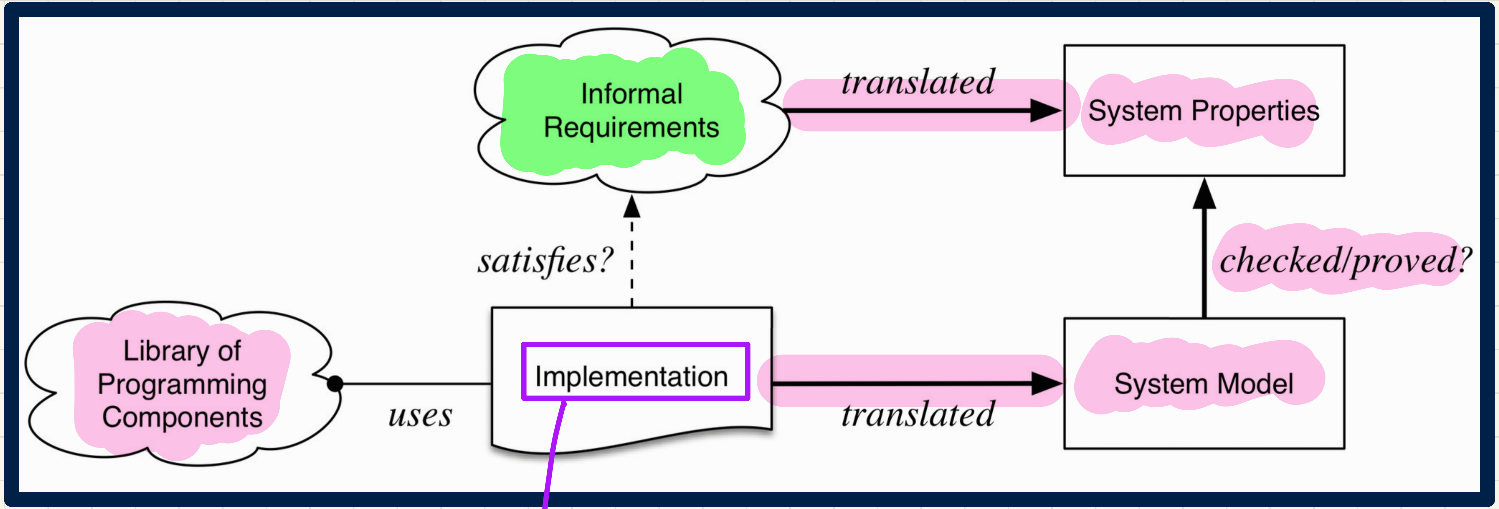
goal

- Are we building the right podct?

does it deviate fom the customer's intended req.

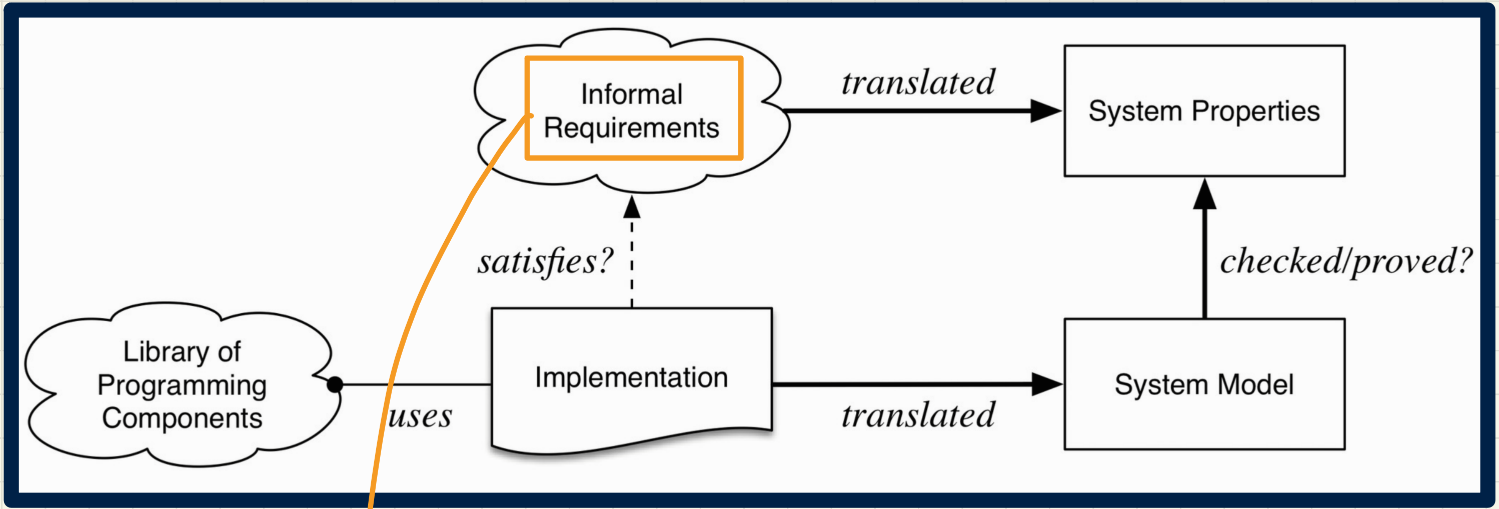
EECS4312.

Building the product right?



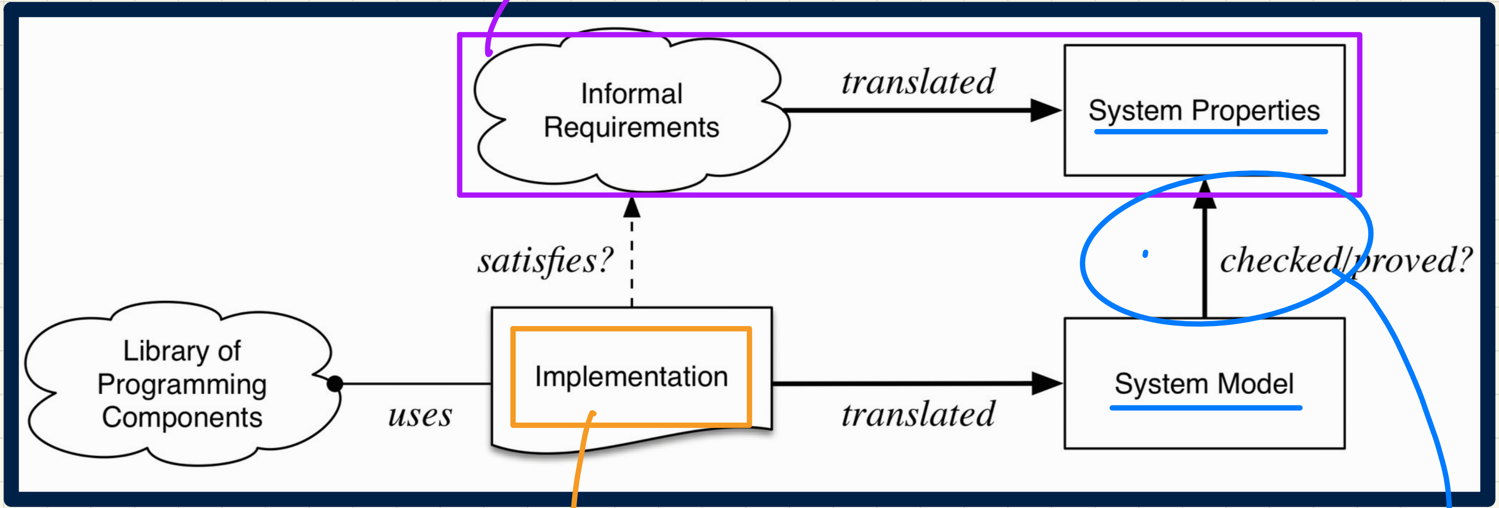
depends on library classes

Building the right product?



does the set of requirements really accurately represent the customer's needs.

given req written in NAT, formulate it in predicates



given the NAT descriptions of a strategy, formulate it as abstract state machines

1. constants
2. variables
3. axioms

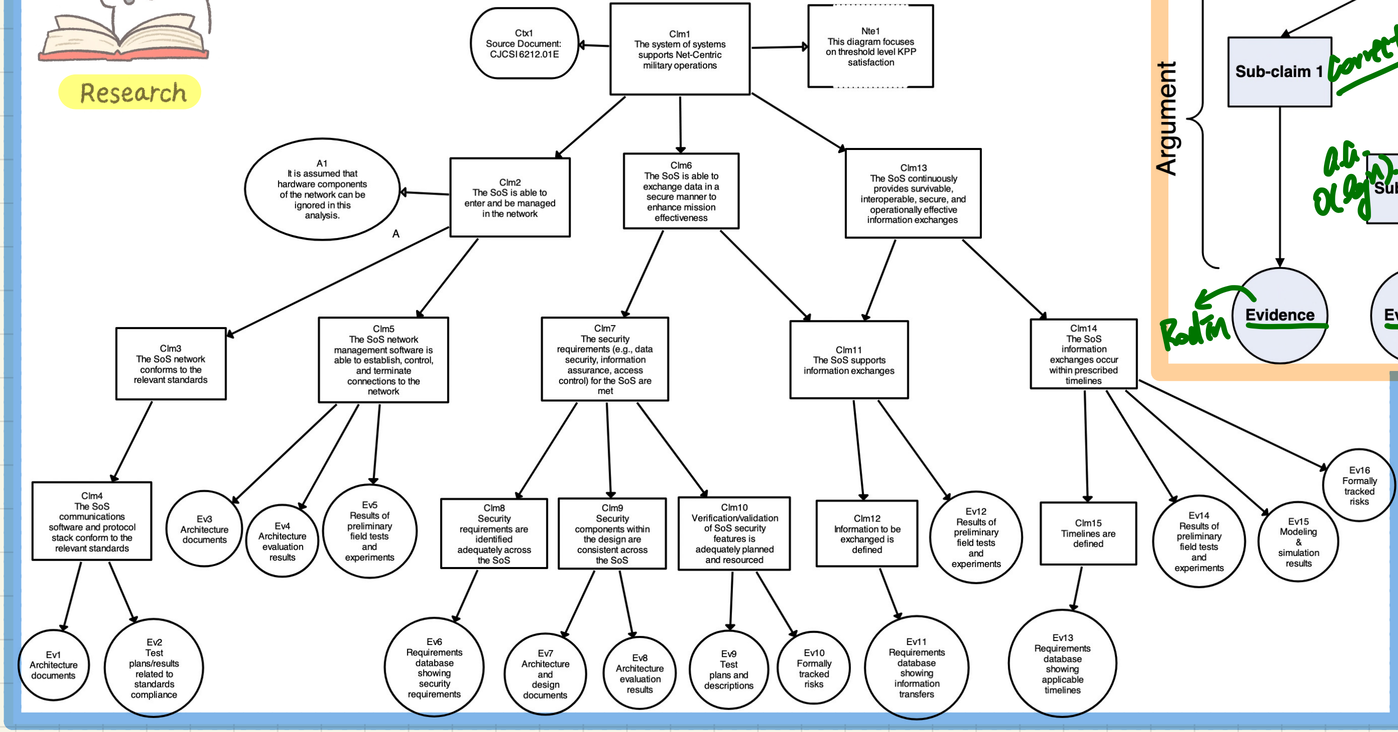
proof obligations

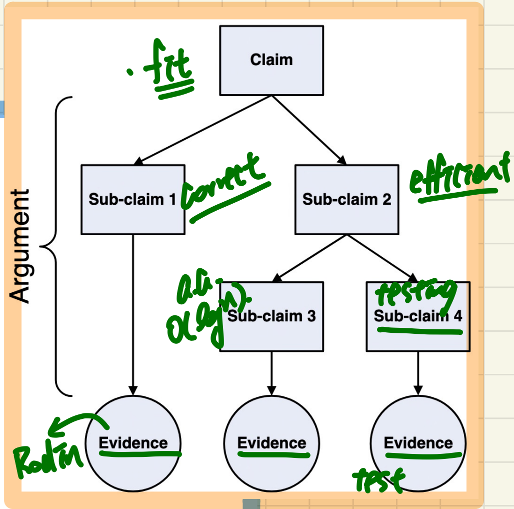
1. prove -
2. genera

Certifying Systems: Assurance Cases

Research

Research on "Assurance Cases" if interested!


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
Ev16
Formally tracked risks
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
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

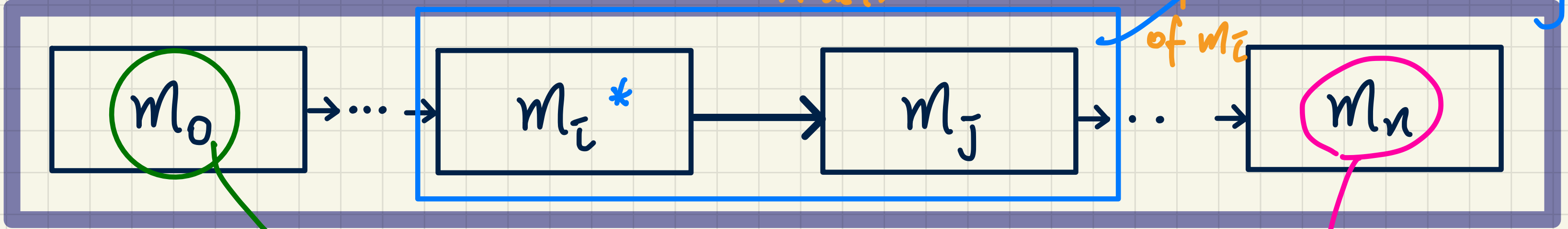
Correct by Construction

2. Instead, distribute different properties to different models

3. Prove m_j is a formal refinement of m_i

n+1 models

m_i is more abstract than m_j



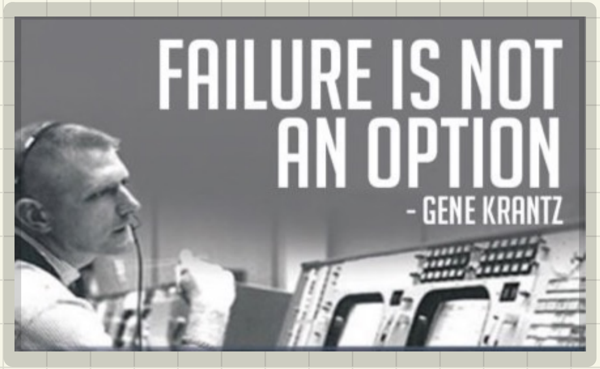
all models describe the same system

Initial, simplest, most abstract model.

1. having a single model and proving all properties on it is infeasible

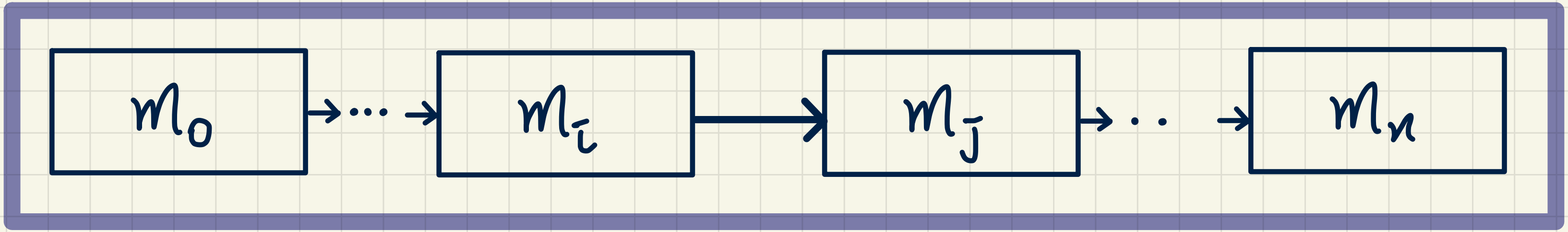
final, most sophisticated, most concrete model (

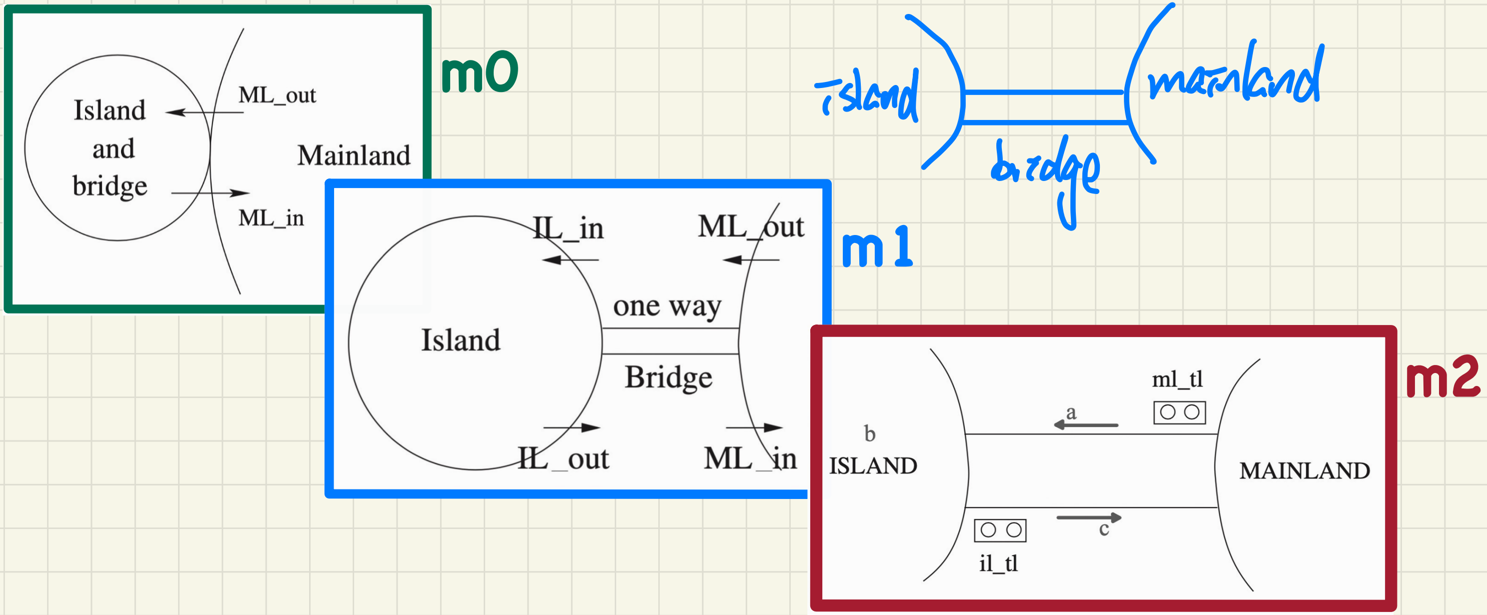
closest for translating into code.



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

Correct by Construction: Bridge Controller System

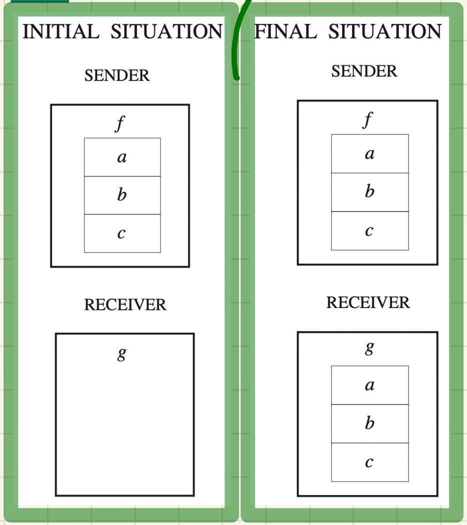




Correct by Construction: File Transfer Protocol

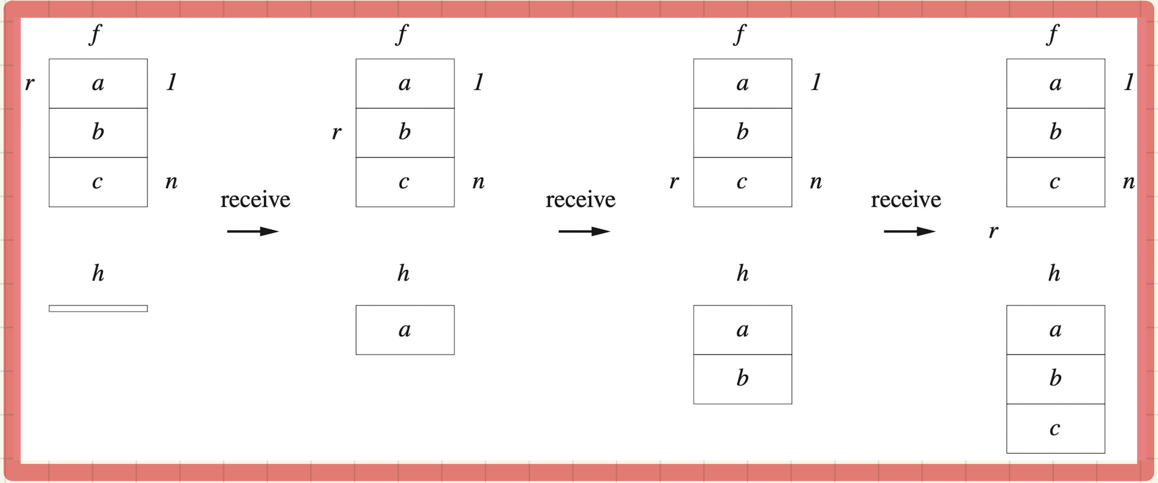
$$\begin{tikzcd} \boxed{m_0} \arrow[r] & \cdots \arrow[r] & \boxed{m_i} \arrow[r] & \boxed{m_j} \arrow[r] & \cdots \arrow[r] & \boxed{m_n} \end{tikzcd}$$

m0



abstracted away delay in sending
(instantaneous transfer)

m1: more concrete than m0



Lecture 3 - January 17

Math Review

Propositional Logic & Predicate Logic

Announcement

- Lab1 released ✓ 2.5 hours

  + tutorial videos

  + problems to solve

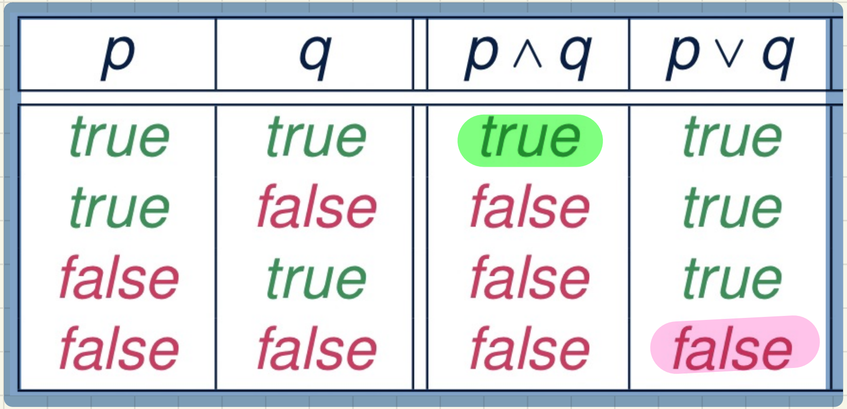
  + Study along with the Math Review lecture notes.

Book

Book.zip

Look at the links of Root installation

Logical Operator vs. Programming Operator



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

short-circuit
↳ evaluation:
L to R

e1 && e2
↳ if LHS evaluates to F, skip the evaluation of RHS

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

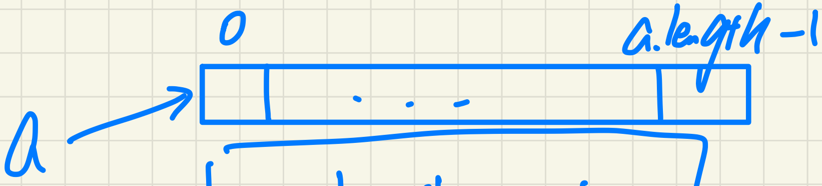
∧ → logical operator

&& ✓ → programming operator
↳ runtime evaluation

short-circuit

e1 || e2
T
↳ if LHS evaluates to T, skip the evaluation of RHS

Accessing Array



a.length == 10

$\underset{①}{i >= 0}$ && $\underset{②}{i < a.length}$ && $\underset{③}{a[i]}$

guarding conditions

expression to protect

① $\boxed{i == -2}$

① evaluates to F ②, ③ skipped

overall : F.

② $\boxed{i == 12}$

① evaluates to T ② evaluates to F

③ skipped overall : F.

int[] a = ...

Exercise Assume $\boxed{a.length == 10}$

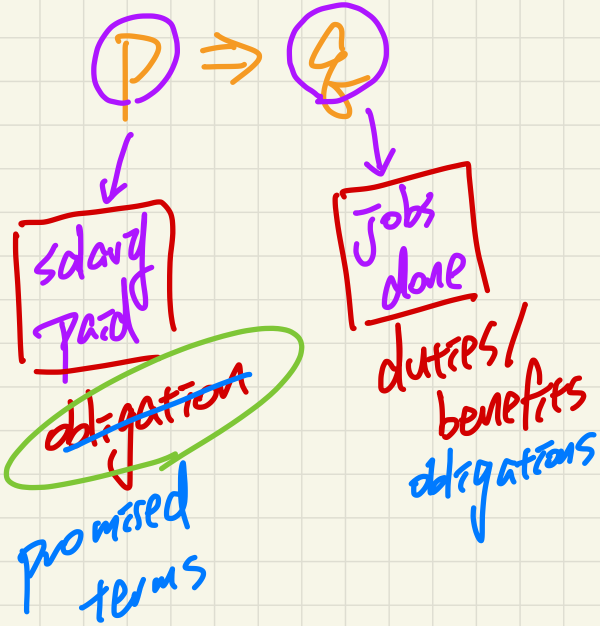
$i < a.length \ \&\& \ a[i] > 10 \ \&\& \ \boxed{i >= 0}$ too late to evaluate for guarding.

↳ does this property guard a[i] ?

↳ No! witness: $\boxed{i == -2}$

Exercises: Try other ordering of guarding conditions.

Implication ≈ Whether a Contract is Honoured



$p \Rightarrow q$

salary paid

~~obligation~~

promised terms

job done

duties/benefits

obligations

True ⇒ True ≡ True

True ⇒ False ≡ False

False ⇒ True ≡ True

False ⇒ False ≡ True.

~~obligations~~ promised terms not fulfilled

Contract not breached regardless of the job being done

Expressing Implications

p: snow storm

q: cancel class

q if p, p is sufficient for q

q is true if p is true

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

if p is not true, no guarantee what q is.

q unless ¬p

one condition for ⇒ to be T.

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

p only if q, q is necessary for p

if p already T, for ⇒ to be T, necessary for q to be T

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

p is not T, don't care.

Prove $p \Leftrightarrow q$

(1) $p \Rightarrow q$ (p only if q) T

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

$p \Rightarrow q$

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

(1) Inverse : $\underline{\neg} p \Rightarrow \underline{\neg} q$

Given

$x > 0 \wedge x \leq 10 \Rightarrow$

$y \geq 3 \vee y < 5$

(2) Converse : $\underline{q \Rightarrow p}$

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

(Inverse of Converse)

(1)

(2)

(3) (apply de morgan when applicable,

Identity

$true \Rightarrow p \equiv p$

$true \wedge p \equiv p$

$false \vee p \equiv p$

$0 + \tau = \tau$

$1 * \tau = \tau$

Zero

$false \Rightarrow p \equiv True$

$\underline{false} \wedge p \equiv \underline{false}$

$true \vee p \equiv true$

Predicate Logic: Quantifiers

- syntax
- base cases in programming

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

range

property

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

(implicitly, if no such i satisfies R, then ∀ is T)

$$\exists i \bullet \boxed{R(i)} \overset{\checkmark}{\wedge} \boxed{P(i)}$$

there's at least one i, s.t. i is in the range <u>and</u> i satisfies P.

(implicitly, if no such i satisfies R, then ∃ is F)

Lecture 4 - January 19

Math Review

Predicate Logic

Sets

Announcement

- Lab1 released
  - + tutorial videos
  - + problems to solve
  - + Study along with the Math Review lecture notes.

Predicate Logic: Quantifiers

- syntax
- base cases in programming

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

for empty array

R(i) false always

universal property

false ⇒ — ≡ T

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

for empty array, R(i) false

existential property

false ∧ — ≡ F

universe of disclosure

↳ what if it's empty

i.e.. R(i) false for any possible value of i

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

∵ no witness in empty array can prove otherwise

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

∵ no witness in empty array can prove so

$\subseteq$

$\subset$

<u>Exercises</u> $\mathbb{N} \subset \mathbb{Z}$, $\mathbb{Z} \subset \mathbb{N}$

$\mathbb{N} \subseteq \mathbb{Z}$, $\mathbb{Z} \subseteq \mathbb{N}$

$\mathbb{N}$ the set of **all** natural #s

$(0, 1, 2, \ldots, +\infty)$

$\mathbb{Z}$ the set of all integer #s

$(-\infty, \ldots, 0, \ldots +\infty)$

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

↳ should hold for <u>all</u> combinations of i, j.

↳ pay attention to how $\forall$ and $\exists$ should be written in Rodin.

Logical Quantifiers: Examples

$\forall i \bullet i \in \mathbb{N} \Rightarrow i \geq \underline{\underline{0}}$ T

0, 1, 2, ...

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

$-2 \in$

$\forall \underline{i, j} \bullet i \in \mathbb{Z} \wedge j \in \mathbb{Z} \Rightarrow \underline{i < j \vee i > j}$ F → $i = j$ $\underline{\underline{(3, 3)}}$

$3 \in \mathbb{Z} \wedge 3 \in \mathbb{Z}$ $3 < 3 \vee 3 < 3$ F

$\exists i \bullet i \in \mathbb{N} \wedge i \geq 0$ T

T e.g. witness: 0, 1, ...

✓ $\exists i \bullet i \in \mathbb{Z} \wedge i \geq 0$ T e.g. 0

$3 < 4$ $3 > 4$

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

$3 \in \mathbb{Z} \wedge 4 \in \mathbb{Z}$ T ↳ witness: $i = 3$, $j = 4$ T

Logical Quantifiers: Examples

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

(1) show $\neg R(\bar{i})$ (i.e. empty universe of disclosure)

zero of $\Rightarrow$: $F \Rightarrow P \equiv \boxed{T}$

harder

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

$T \Rightarrow T \equiv \boxed{T}$

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

similar

(1) show a witness $\bar{i}$ s.t. $R(\bar{i})$, $P(\bar{i})$

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

(1) give a counter-example/witness $\bar{i}$ s.t. $R(\bar{i})$, $\neg P(\bar{i})$

(i.e., an element in array but does not satisfy property).

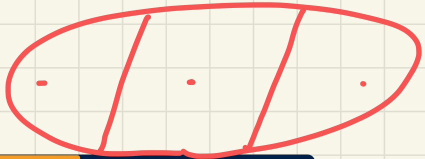
How to disprove $\exists i \bullet R(i) \wedge P(i)$?

(1) show $\neg R(\bar{i})$ (empty). $F \wedge P \equiv F$

harder

(2) show $R(\bar{i})$, $\neg P(\bar{i})$

Prove/Disprove Logical Quantifications



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

non-empty : 1, 2, 3, ..., 10 ⇒ all > 0.

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

↳ Counter-example/witness: $x = 1$

$T \Rightarrow F \equiv F$.

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

↳ witness: 2 $T \wedge T \equiv T$.

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

↳ non-empty: 1, 2, 3, ..., 10

↳ all make $x > 10$ F.

Logical Quantifications: Conversions

$R(x)$: $x \in$ 3342_class

$P(x)$: x receives A+

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

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

$$\equiv \neg(\exists x \bullet \neg(R(x) \Rightarrow P(x)))$$

$$\equiv \neg(\exists x \bullet \neg(\neg R(x) \vee P(x))) \equiv \neg(\exists x \bullet \underline{R(x) \wedge \neg P(x)})$$

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

$\downarrow$

Exercise! . .

✓ De Morgan

Lecture 1b

Review on Math: Sets

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

Empty Set : $\emptyset$

$\{\}$

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

$|\emptyset| = 0$

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

Set Comprehension

$\{x \mid x \in \mathbb{N} \wedge x > 5\}$

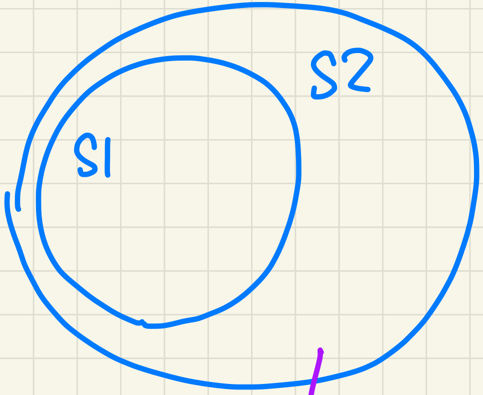
$= \{6, 7, 8, \ldots \infty\}$

$5 \in \mathbb{N}$

$-2 \notin \mathbb{N}$

$$\Big\{ \underbrace{\qquad\qquad}_{\text{list of variables}} \Big| \underbrace{\qquad\qquad\qquad\qquad}_{\text{constraint}} \Big\}$$

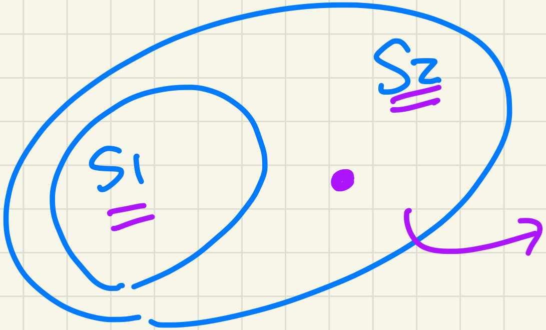
$S_1 \subseteq S_2$



$S_2 \setminus S_1$ may be ϕ

in S_2 but not in S_1

$S_1 \subset S_2$



$S_2 \setminus S_1$ must be non-empty

Power Set

$$\mathbb{P}(S) = \{x \mid x \subseteq S\}$$

each member in a power set is a subset.

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

$$\mathbb{P}(\{1,2,3\}) = \left\{ \begin{array}{ll} \emptyset, & \text{/* card is 0 */} \\ \{1\}, \{2\}, \{3\}, & \text{how many: } \binom{3}{1} = 3 \\ \{1,2\}, \{2,3\}, \{1,3\}, & \text{how many: } \binom{3}{2} = 3 \\ \{1,2,3\} & \text{/* card } |\{1,2,3\}| \end{array} \right\}$$

subsets of card. 1

subsets of card. 2

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

$$\underbrace{\binom{|S|}{0}}_{\emptyset} + \underbrace{\binom{|S|}{1}}_{\text{subsets of card 1}} + \cdots + \underbrace{\binom{|S|}{|S|}}_{S} = \sum_{c=0}^{|S|} \binom{|S|}{c}$$

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

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

Lecture 5 - January 24

Math Review

Relations

Announcement

- Lab1 submission due in a week
  - + tutorial videos
  - + problems to solve
  - + Study along with the Math Review lecture notes.

Sets: Exercises

$4 \leq 7$ T

$\land$

$4 \geq 7$ F

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

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

Find a common pattern for defining:

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

$\forall x, y \bullet x \in \mathbb{Z} \land y \in \mathbb{Z} \land x = y \Rightarrow x \geq y \land x \leq y.$

$S = T \Rightarrow S \subseteq T \land S \supseteq T.$

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

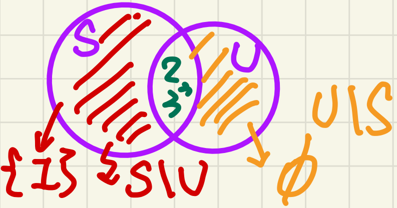
RHS

LHS

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

(exercise!).

$S \setminus U \quad U \setminus S$



Is set difference (\) commutative?

No.

$S \setminus T \overset{\text{In general}}{\neq} T \setminus S$

Bidirectional Subset Relations: Programming

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

Formulate the `allPositive` method using a set comprehension.

$$\text{input} = \{2, 3, -1, 4, -2\}$$

$$\text{allPositive}(\text{input}) = \underline{\{2, 3, 4\}}$$

$\{1, 25, 46\}$ ✗

formulate

$\wedge\ x \in \text{input}.$

✓

$$\text{allPositive}(\text{input}) = \left\{ x \;\middle|\; \underset{\text{not complete}}{\underline{x > 0}} \right\}$$

Bidirectional Subset Relations: Programming

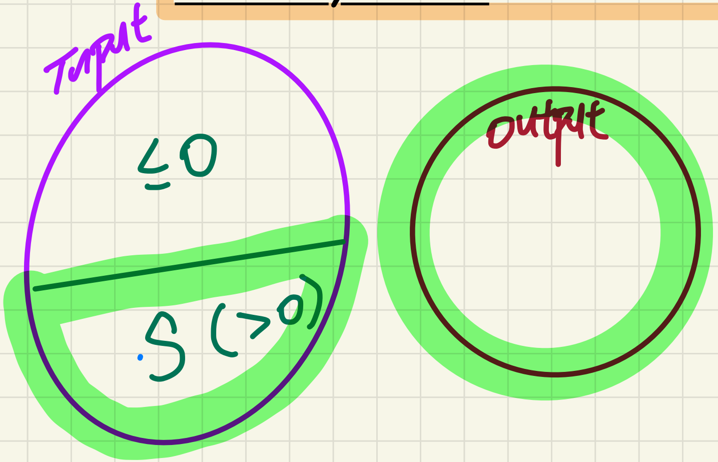
Post-Condition

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

Say:

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

Formally relate the two sets S and **output**.



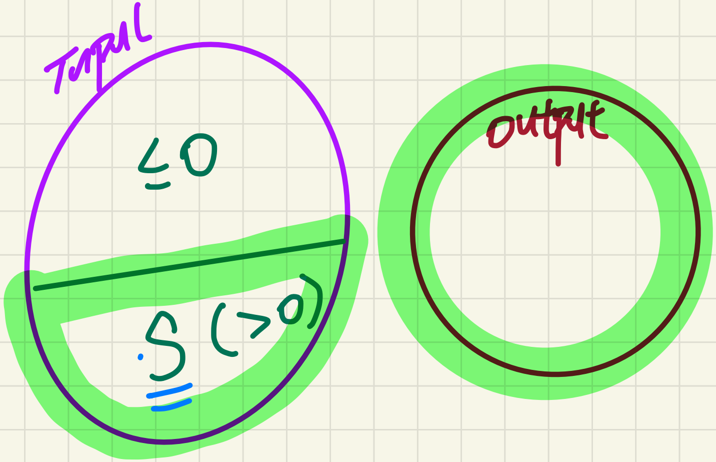
(N1)

(p1) output ⊆ S

(p2) S ⊆ output

} S = output

- What if only p1 is required? e.g. ∅ can satisfy p1
- What if only p2 is required? e.g. output contains elements ∉ S



✓ $\forall x \cdot x \in S \Leftrightarrow x \in output$

valid on paper, but inconvenient to put into Rodin.

(P1) $\forall x \mid x \in output \Rightarrow \overset{x > 0}{\boxed{x \in S}}$

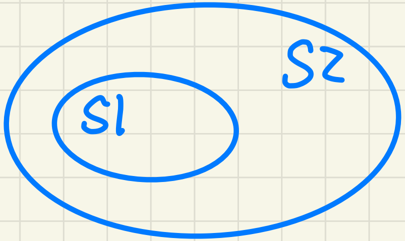
(P2) $\forall x \mid \underbrace{x \in input \land x > 0}_{x \in S} \Rightarrow x \in output$

$\emptyset \subset S$

True if $|S| > 0$

False if $|S| = 0$

i.e. $\emptyset \subset \emptyset$ F.



$S1 \subset S2$ T

$\underline{S1 \subseteq S2}$ T but $\nRightarrow$ $\underline{S1 = S2}$

Cardinality of Power Set: Interpreting Formula

flexible: e.g. how many subsets of card between l and u

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

$$|\mathbb{P}(S)| = \underbrace{\overset{\boxed{1}}{\binom{|S|}{0}}}_{\text{\# of subsets of card 0}} + \underbrace{\overset{|S|}{\binom{|S|}{1}}}_{\text{\# subsets of card 1}} + \cdots + \underbrace{\overset{\boxed{1}}{\binom{|S|}{\boxed{|S|}}}}_{\text{\# maximum subset of card } |S|}$$

{a, b, c}

↳ {a}, {b}, {c}

$2^{|S|}$

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

a	b	c	subset
0	0	0	∅
0	0	1	{c}
⋮			
1	1	1	{a, b, c}

Lecture 1b

Review on Math: Relations

Set of Tuples

$$|S_1 \underset{\text{cross product}}{\times} S_2 \times \dots \times S_n| = |S_1| \underset{\text{multiplication}}{*} |S_2| * \dots * |S_n|$$

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

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

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

structure of each member in the resulting set

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

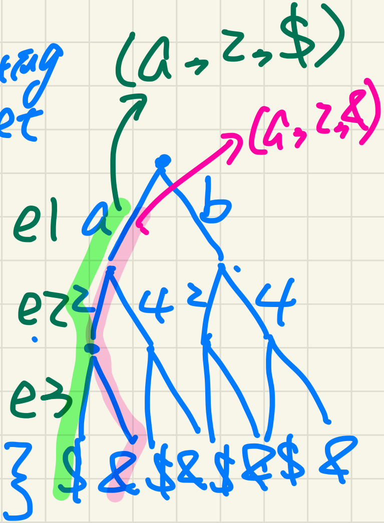
$$\underset{S_1}{\{a, b\}} \times \overset{S_2}{\{2, 4\}} \times \overset{S_3}{\{\$, \&\}}$$

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

$$= \{ \qquad \}$$

8 members.

exercise!



Relation : set of pairs

e.g. Relation on $\{1, 2, 3\}$ and $\{a, b\}$

S_1 S_2

- Is $(1, a)$ a relation on S_1 and S_2? No!

  not even a set!

- Is $\{(b, 2)\}$ a relation on S_1 and S_2?

  ↓ S_1 elements should come first in the ordered pair

- YES Is $\{(1, a), (3, b)\}$ a relation on S_1 and S_2?

- Minimum relation on S_1 and S_2? ϕ ~> empty relation!

- ✓ Maximum relation on S_1 and S_2? $S_1 \times S_2$

empty relation

$\{\{\}\}$

$\{\}$

① ×

$\{\phi\}$

② ϕ

or

Given two sets $\underline{S}$ and $\underline{T}$:

- min relation: $\emptyset$
- max relation: $S \times T$
- set of All possible relations on S and T:

$\{$

$\emptyset$ relation of card 0

$\boxed{\begin{array}{l} \vdots \quad 1 \\ \vdots \quad 2 \\ \vdots \quad \vdots \\ \quad\quad |S| * |T| - 1 \end{array}}$

$\underline{S \times T}$ relation of card of card $|S| * |T|$

$\}$

each member is a relation.

Lecture 6 - January 26

Math Review

Relations, Relational Operations

Announcement

- Lab1 submission due in a week

  + Help: scheduled office hours & TA

  + tutorial videos

  + problems to solve

  + Study along with the Math Review lecture notes.

set of possible relations on S and T

$r \in S \leftrightarrow T$

a single relation on S and T

$$\boxed{\begin{array}{l} S = \{1, 2, 3\} \\ T = \{a, b\} \end{array}}$$

$r : S \leftrightarrow T$

$\in$

could be:

(1, a) or (2, b)

$r2 \in \underbrace{\{(1, a), (2, b)\}}_{\in\, S \leftrightarrow T}$

$r \in S \leftrightarrow T$

r could be: (1) $\emptyset$

(2) $S \times T$

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

Set of Possible Relations

- Set of possible relations on S and T:
- Dedicated symbol for set of possible relations on S and T:
- Declare that set r is a relation on S and T:

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

card: $2^4 = 16$

Hint: How many?

$\{a, b\} \leftrightarrow \{2, 4\}$

$\mathbb{P}(\{a, b\} \times \{2, 4\})$

max relation on the two sets

$\rightarrow \{(a, 2), (a, 4), (b, 2), (b, 4)\}$

$\mathbb{P}(\{(a, 2), (a, 4), (b, 2), (b, 4)\})$

$= \{ \emptyset,$ /* relation of size 0 */

$\{(a, 2)\}, \{(a, 4)\}, \{(b, 2)\}, \{(b, 4)\}$ /* rel of size 1 */ $\binom{4}{1} = 4.$

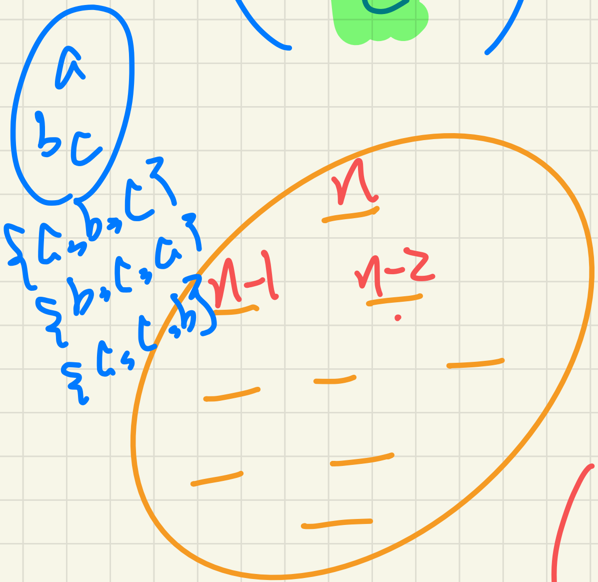
(a) relations of size 2 $\binom{4}{2} = \frac{4*3}{2!} = 6$

(b) relations of size 3 $\binom{4}{3} = \frac{4*3*2}{3!} = 4$

$\{(a, 2), (a, 4), (b, 2), (b, 4)\} \}$ /* relation of size 4 */

$$\binom{n}{r}$$

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



set of card. r

$$\frac{\overbrace{n\cdot(n-1)\cdot(n-2)\cdot\cdots\cdots(n-r+1)}^{r\ \text{terms}}}{r!}$$

$r!$ ↓ disregard duplicates

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

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

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

$$airline \in \boxed{Departure \leftrightarrow Destination}$$

↳ task: enumerate!

Relational Operations: Domain, Range, Inverse

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

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

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

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

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

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

$|r| = |r\sim|$

algebraic properties

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

(1) $\underline{\mathrm{dom}(r) = \mathrm{ran}(r\sim)}$ (2) $\mathrm{ran}(r) = \mathrm{dom}(r\sim)$

Relational Operations: Image

* $r \in S \leftrightarrow T$

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

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

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

Alphabet: all lower & upper letters

$r[s]$

s: a set!

$r[\{a, b\}] = \{r' \mid (d, r') \in r \land d \in \{a, b\}\}$

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

Exercises

- Image of {a, b} on r?
- Image of {1, 2} on r?* undefined
- Image of {1, 2} on the inverse of r? → $\{a, b, d, e\}$
- Calculate r's range via an image.
- Calculate r's domain via an image.

e.g. $r[\emptyset] = \emptyset$

e.g. $r[\{x, y\}] = \emptyset$

$\text{ran}(r) = r[\text{dom}(r)]$

$\text{dom}(r) = r\sim[\text{ran}(r)]$

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

$r[\{1, a\}]$ × undefined

$$\underline{\underline{r}} \in S \leftrightarrow T \qquad s \subseteq S$$

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

another relation

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

domain restriction

Relational Operations: Restrictions vs. Subtractions

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

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

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

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

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

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

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

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

$$r = (s \lhd r) \cup (s ⩤ r)$$

Relational Operations: Overriding

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

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.

$$r \lhd\!\!+ \underset{\text{a relation}}{\boxed{t}} = \{(d, r') \mid \underline{(d, r') \in t} \vee (\underline{(d, r') \in r \wedge d \notin \text{dom}(t)})\}$$

$$\boxed{r} \lhd\!\!+ \underbrace{\{(a,3), (c,4)\}}_{t} = \{(d, r') \mid (d, r') \in \{(a,3), (c,4)\} \vee \underbrace{(d, r') \in r \wedge d \notin \{a, c\}}_{\text{domain subtraction}}\}$$

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

Problems (don't look at the slides!)

(1) Rewrite the relational image $r[s]$ in terms of dom/ran and/or restrictions/subtractions.

(2) Rewrite the overriding $r \lhd\!\!\!+ t$ in terms of dom/rand and/or restrictions/subtractions and/or set operations.

Lecture 1b

Review on Math: Functions

Functional Property

relation

$\{\underline{\overset{s\ \ t1}{(a, 1)}}, (b, 2), \underline{\overset{s\ \ t2}{(a, 3)}}\}$

↳ a relation

not a function!

$$isFunctional(r) \Leftrightarrow$$

$\in S \leftrightarrow T$

$$\forall \underline{\overset{\in S}{s}}, \underline{\overset{\in T}{t1, t2}} \bullet$$

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

$$\Rightarrow$$

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

each domain value s maps to at most one range value

Q: Smallest relation satisfying the functional property.

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

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

Lecture 7 - January 31

Math Review

Functions, Modelling

Announcement

Lab1 solution → today

Lab2 → nex Monday

WT1

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?

set of range values

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

another relation

domain restriction

↓ should be: $s \subseteq \mathrm{dom}(r)$ otherwise: result is $\emptyset$.

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?

another relation

$$r \mathbin{\lhd\!\!\!+} \underbrace{t}_{\text{a relation}} = \boxed{t \cup (\boxed{\mathrm{dom}(t) ⩤ r})}$$

$$(\mathrm{dom}(r) \setminus \mathrm{dom}(t)) \lhd r$$

Functional Property

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

relation

e.g. $\{\underset{s\ \ t1}{(a,1)}, (b,2), \underset{s\ \ t2}{(a,3)}\}$

↳ a relation

not a function!

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

$r \in S \leftrightarrow T$

$$\forall \underset{\in S}{s}, \underset{\in T}{t1}, \underset{\in T}{t2} \bullet$$

cnt!

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

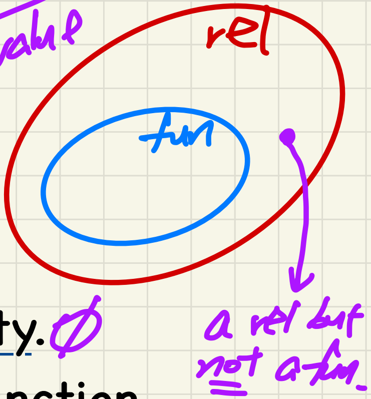
$$\Rightarrow$$

cnt.2

each domain value s maps to one.

at most one range value

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



Q: Smallest relation satisfying the functional property. ∅

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

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

Prove r is a fun

(1) for each pair in r, satisfying cnt1, satisfies cnt2 ⇒ con.

(2) Show $r = \emptyset$.

Disprove r is a fun

(1) r is not empty, there's $(s,t1) \in r \land (s,t2 \in r)$ but $t_1 \neq t2$

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

$$\begin{aligned}
&\text{isFunctional}(r) \Leftrightarrow \\
&\quad \forall\, s, t1, t2 \bullet \\
&\qquad (\, s \in S \land t1 \in T \land t2 \in T \,) \\
&\qquad \Rightarrow \\
&\qquad \boxed{(\, (s, t1) \in r \land (s, t2) \in r \Rightarrow t1 = t2 \,)}
\end{aligned}$$

$\equiv$ contra-positive

$$t1 \neq t2 \Rightarrow \boxed{\neg\, (\, (s, t1) \in r \land (s, t2) \in r)}$$

$$\downarrow$$

$$\boxed{(s, t1) \notin r \lor (s, t2) \notin r}$$

If t1 and t2 are distinct values, then we cannot have s mapping to both of them.

Partial Functions vs. Total Functions

⇸

+→

partial

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

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

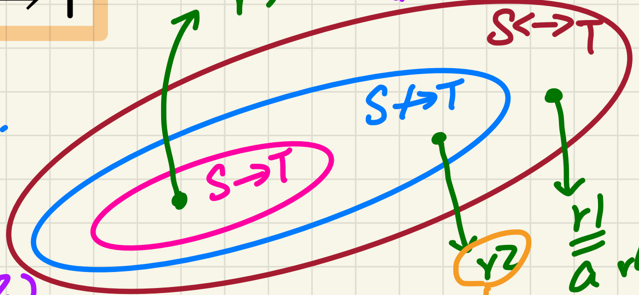
total

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

Every function is a partial function.

1. a relation (∵ set of pairs)
2. a partial fun (∵ not violate
3. a total fun (∵ dom = S) fun. prop.)

r3: a relation, a partial fun, and a total fun.



r1 = a relation which violates the functional property

r2: a function but dom(r) ⊂ S

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

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

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

1. a rel.
2. a partial fun.
3. not a total fun. ∵ $\underbrace{\text{dom}}_{\{2, 1\}} \neq \{1, 2, 3\}$

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

↳ 1. a rel. 2. not a fun.

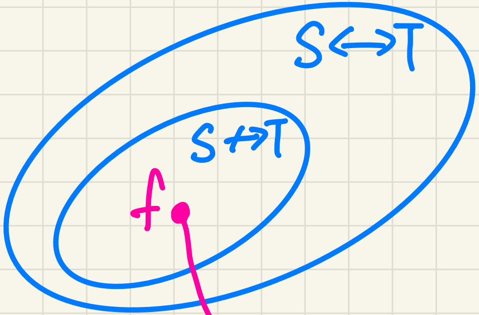
Relational Image vs. Functional Application

A function is a relation.

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

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

↳ a rel, a partial fun, not a total fun.



1. a relation f[s]
2. a function

Exercises:

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

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

$\boxed{f[\{2\}] = \emptyset}$ → for a function that's partial but not total (i.e. dom(f) ⊂ S, there's at least one value in S that maps to nothing in f).

Input: singleton sets

$f(3) = a$

$f(1) = b$

$f(2) =$ undefined

⊥ ⇝ bottom

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? X

→ {("alan", SB106), ("alan", VC102)}

Is where_is ∈ Employee → Location appropriate?

dom(where_is) = Employee X not realistic to expect all employees to be present in company all the time

Is where_is ∈ Employee ⇸ Location appropriate?

↳ a relation satisfying the fun. prop. but is not total

Functions

	dom injective	ran surjective	dom, ran bijective
partial	⤔	⤀	n.a.
total	↣	↠	⤖

Lecture 8 - February 2

Math Review

Injection vs. Surjection vs. Bijection
Formulating Arrays
Lab1 Solution Highlights

Injective Functions

no witness to prove violation of inj. prop.

$f \in S \leftrightarrow T$

functional property

isInjective(f)

⟺

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

$(s, t_1) \in f \wedge (s, t_2) \in f \Rightarrow t_1 = t_2$

rel, fun, partial fun, not total, inj!

to violate injective prop. $s_1 \neq s_2$ and s_1, s_2 both map to t.

- If f is a *partial injection*, we write: $f \in S ⤔ T$
  - e.g., $\{\varnothing, \{(1, \mathbf{a})\}, \{(2, \mathbf{a}), (3, \mathbf{b})\}\} \subseteq \{1, 2, 3\} ⤔ \{a, b\}$
  - e.g., $\{(1, \mathbf{b}), (2, a), (3, \mathbf{b})\} \notin \{1, 2, 3\} ⤔ \{a, b\}$
  - e.g., $\{(1, \mathbf{b}), (3, \mathbf{b})\} \notin \{1, 2, 3\} ⤔ \{a, b\}$

the set of all possible partial injections between two sets

- If f is a *total* injection*, we write: $f \in S \rightarrowtail T$
  - all possible total inj.
  - e.g., $\{1, 2, 3\} \rightarrowtail \{a, b\} = \varnothing$ $\{(1,a), (2,b), (3,a)\}$
  - e.g., $\{(2, d), (1, a), (3, c)\} \in \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$
  - e.g., $\{(2, d), (1, c)\} \notin \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$ not total, inj.
  - e.g., $\{(2, \mathbf{d}), (1, c), (3, \mathbf{d})\} \notin \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$ total, not inj.

rel, partial fun, total fun, t

↳ $\overset{s_1}{(1, b)} \in f \wedge \overset{s_2}{(3, b)} \in f \Rightarrow \overset{F}{\boxed{1 = 3}}$ violation

Surjective Functions

assumed: f is a function.

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

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

rel, partial, total, sur.

- e.g., $\{\{(1,\mathbf{b}),(2,\mathbf{a})\}, \{(1,\mathbf{b}),(2,\mathbf{a}),(3,\mathbf{b})\}\} \subseteq \{1,2,3\} \mathrel{+\!\!\!\!\twoheadrightarrow} \{a,b\}$
- e.g., $\{(2,\mathbf{a}),(1,\mathbf{a}),(3,\mathbf{a})\} \notin \{1,2,3\} \mathrel{+\!\!\!\!\twoheadrightarrow} \{a,b\}$
- e.g.,: $\{(2,\mathbf{b}),(1,\mathbf{b})\} \notin \{1,2,3\} \mathrel{+\!\!\!\!\twoheadrightarrow} \{a,b\}$

rel, part. fun. sur. not total

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

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

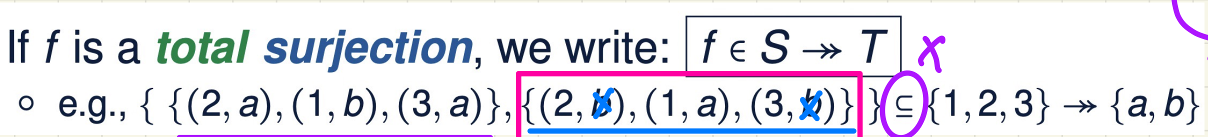
total, not sur.

partial ⇏ total

total ⇒ partial

total surjections ✓ ⇒ partial surjections.

F.

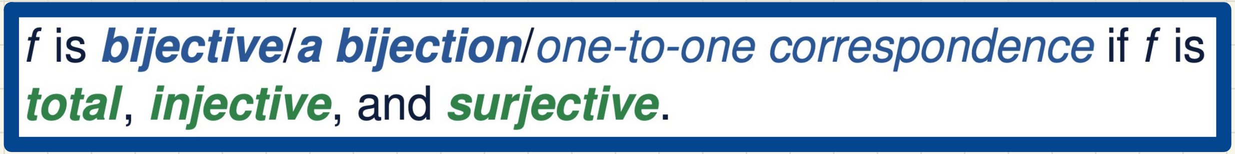


If f is a ***total surjection***, we write: $\boxed{f \in S \twoheadrightarrow T}$ $\times$

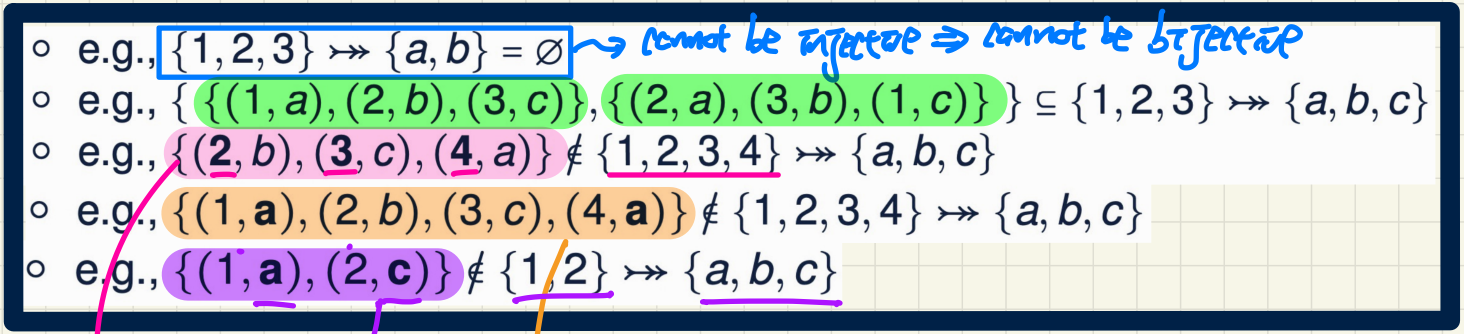
- e.g., $\{\ \underline{\{(2,a),(1,b),(3,a)\}}, \boxed{\underline{\{(2,\underset{a}{\cancel{b}}),(1,a),(3,\underset{a}{\cancel{b}})\}}}\ \} \subseteq \{1,2,3\} \twoheadrightarrow \{a,b\}$

$\notin \{1,2,3\} \twoheadrightarrow \{a,b\}$

Bijective Functions



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



- e.g., $\{1,2,3\} \rightarrowtail\!\!\!\twoheadrightarrow \{a,b\} = \varnothing$ → cannot be injective ⇒ cannot be bijective
- e.g., $\{\{(1,a),(2,b),(3,c)\},\{(2,a),(3,b),(1,c)\}\} \subseteq \{1,2,3\} \rightarrowtail\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., $\{(2,b),(3,c),(4,a)\} \notin \{1,2,3,4\} \rightarrowtail\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., $\{(1,\mathbf{a}),(2,b),(3,c),(4,\mathbf{a})\} \notin \{1,2,3,4\} \rightarrowtail\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., $\{(1,\mathbf{a}),(2,\mathbf{c})\} \notin \{1,2\} \rightarrowtail\!\!\!\twoheadrightarrow \{a,b,c\}$

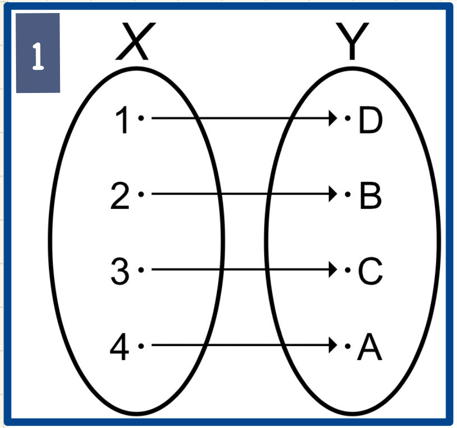
not total

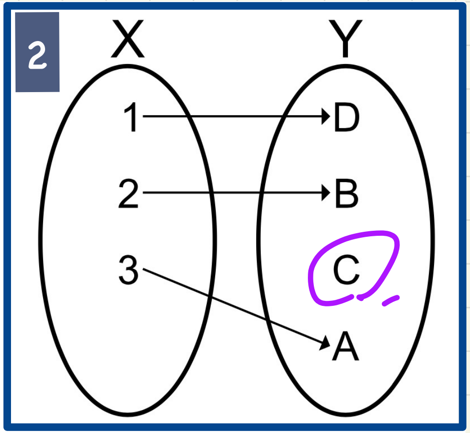
inj,
not sur,
total

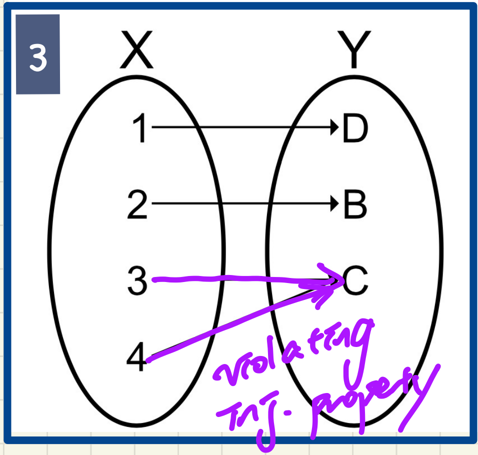
not inj.
sur.
total

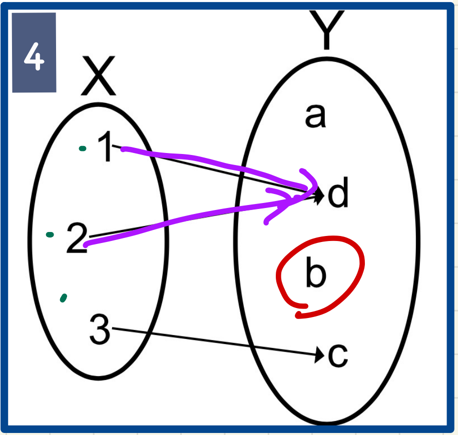
Exercise

$\in X \leftrightarrow Y$





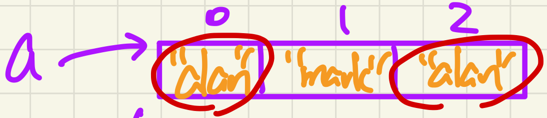




	1	2	3	4
partial	✓	✓	✓	✓
total	✓	✓	✓	✓
injection	✓	✓	✗	✗
surjection	✓	✗	✓	✗
bijection	✓	✗	✗	✗

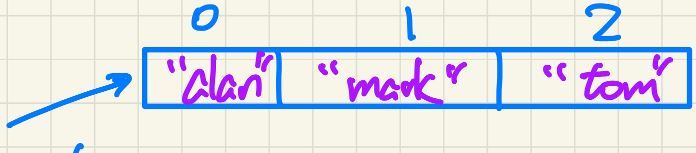
Formalizing Arrays as Functions

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



$\downarrow \{(0, \text{"alan"}), (1, \text{"mark"}), (2, \text{"alan"})\}$

not inj ⇒ duplicates



$\downarrow \{(0, \text{"alan"}), (1, \text{"mark"}), (2, \text{"tom"})\}$

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

not appropriate

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

No!

(1) e.g. $\{(0, \text{"alan"}), (0, \text{"jim"})\}$

an index should only hold at most one value

(2) e.g. $\{(-1, \text{"jonathan"}), (3, \text{"peter"})\}$

invalid indices

$a \in \mathbb{Z} \mathrel{+\!\!\!\twoheadrightarrow} \boxed{String}$

length: 0

1

2 "ab" "bc" . . .

⋮

Array

$$\begin{tikzcd} a \in \overset{\checkmark}{\mathbb{Z}} \arrow[r, "+"] \arrow[d] & \text{Object} \\ \checkmark\, \mathbb{N} & \end{tikzcd}$$

$$a \in \underbrace{\mathbb{N} \to \text{Object}}$$

<u>not</u> good ∵ indices may be too large.

CONTEXT C0

SETS

ACCOUNT carrier set: abstract without the need to enumerate content of the set

PERSON carrier set: details of each member in PERSON are abstracted away (ENV9) - Solution to Exercise 4 of Lab1

CONSTANTS

c credit limit (ENV3)

L pre-set upper bound (ENV3) - Solution to Exercise 3 of Lab1

AXIOMS

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

not theorem means an axiom; theorem means a proof is needed. In this case, the typing constraint should be an axiom.

thm1: ⟨theorem⟩ $c > 0$

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

typing constraint of variable L - Solution to Exercise 3 of Lab1

END

MACHINE Bank0

// Initial model of the bank system

SEES C0

VARIABLES

b balance (ENV2)

d cash drawer (REQ7)

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

INVARIANTS

inv1: $b \in ACCOUNT \rightarrow\!\!\!\!\!+ \mathbb{Z}$

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

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

(ENV3)

Cannot be satisfied simultaneously (e.g. when every account has -c balance)

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

(ENV3) - Solution to Exercise 3 of Lab1

inv5: $owner \in ACCOUNT \rightarrow\!\!\!\!\!+ PERSON$

(ENV9) - Solution to Exercise 4 of Lab1

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

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

inv7: $d > 0$

Contradicts with

REQ8 - this was not assigned as a tak for your Lab1. But encoding REQ8 as an invariant shows the value of a formal tool like Rodin: information requirements like E- and R-descriptions are likely to cotain contradictions which are not easy to detect.

EVENTS

Initialisation

begin

act1: $b := \varnothing$

act2:

$d := 0$

(REQ4)

act3: $owner := \varnothing$

Empty bank (ENV9) - Solution to Exercise 4 of Lab1

end

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

(REQ6) - Exercise 2 from Lab1: withdraw/inv3/INV cannot be proved.

any

a account to withdraw

v value to withdraw

where

type_of_a: $a \in ACCOUNT$

typing constraint of event parameter a

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

typing constraint of event parameter v

wd_for_b(a): $a \in dom(b)$

inv_3: $b(a) - v \geq -c$

Solution to Exercise 2 of Lab1

then

act1: $b(a) := b(a) - v$

updates the balance of a

act2: $d := d - v$

updates the cash drawer

end

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

(REQ5) - Solution to Exercise 3 of Lab1

any

a

v

where

grd1: $a \in dom(b)$

grd2: $v \in \mathbb{N}_1$

grd3: $b(a) + v \leq L$

then

act1: $b(a) := b(a) + v$

act2: $d := d + v$

end

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

(REQ4) - Solution to Exercise 4 of Lab1

any

p

a

where

grd1: $p \in PERSON$

grd2: $a \in ACCOUNT$

grd3: $a \notin dom(owner)$

then

act1: $b := b \cup \{a \mapsto 0\}$

Note. Might need the PP prover to discharge POs related to inv3/inv4

act2: $owner := owner \cup \{a \mapsto p\}$

end

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

(REQ10) - Solution to Exercise 4 of Lab1

any

a

where

grd1: $a \in dom(b)$

grd2: $b(a) = 0$

then

act1: $b := \{a\} ⩤ b$

act2: $owner := \{a\} ⩤ owner$

end

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

(REQ11) - Solution to Exercise 4 of Lab1

any

a1

a2

v

where

grd1: $a1 \in dom(b)$

grd2: $a2 \in dom(b)$

grd3: $a1 \neq a2$

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

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

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

Necessary to make POs related to inv3/inv4 discharged

overriding

then

act1: $b := b \boxed{} \overset{t\,.}{\underline{\{a1 \mapsto b(a1) - v, a2 \mapsto b(a2) + v\}}}$

Note. It's not allowed to have two actions involving the same LHS variable: b(a1) := ... , b(a2) := ...

end

END

Rewrite

$b := t \cup \{a1, a2\} ⩤ b$

Lecture 9 - February 7

Reactive System: Bridge Controller

Announcements

- Lab2 released
- WrittenTest1 coming

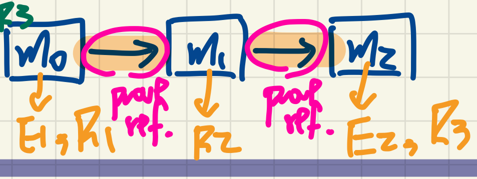
Lecture

Reactive System: Bridge Controller

Correct by Construction
State Space
Req. Doc.

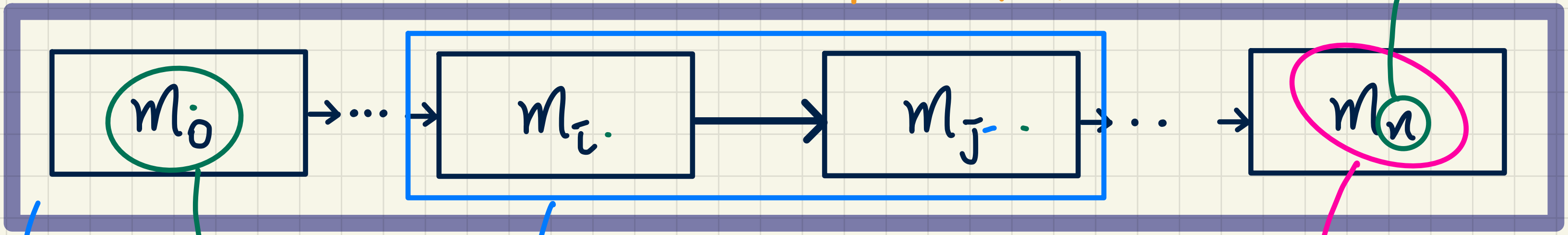
Correct by Construction

RD: E1, E2, R1 ~ R3



bridge controller

n=3



models: descriptions of SUD by filtering out irrelevant details

most abstract

m_j refines m_i

m_i is refined by m_j

by adding: 1. variables

2. actions / invariants

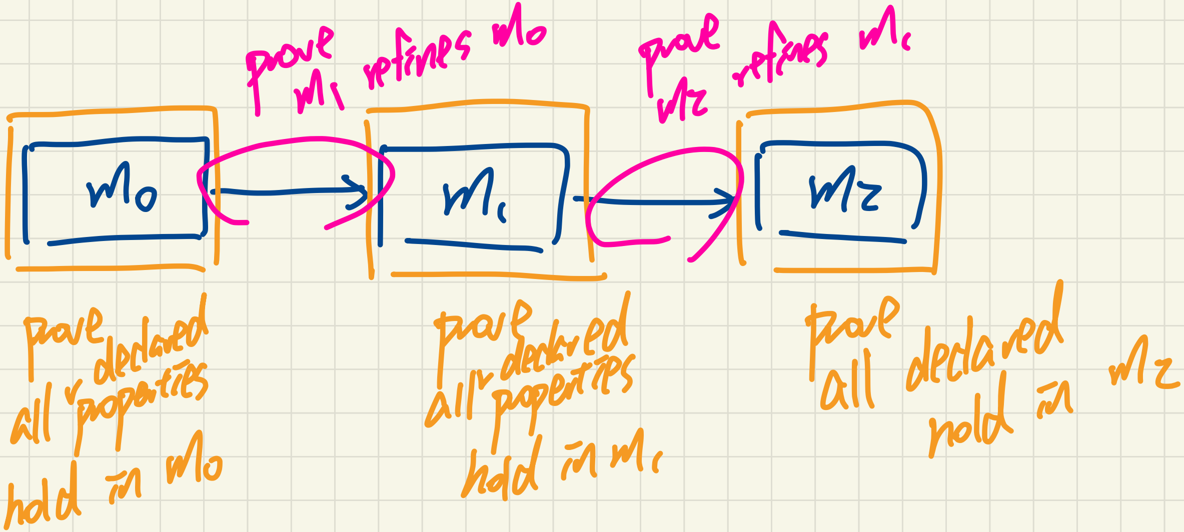
↳ more POs to discharge.

most concrete (closest to code)

RD

↳ E-descriptions

↳ R-descriptions


prove m_1 refines m_0
prove m_2 refines m_1
m_0
m_1
m_2
prove all declared properties hold in m_0
prove all declared properties hold in m_1
prove all declared hold in m_2

Is it necessary to <u>also</u> prove m_2 refines m_0?

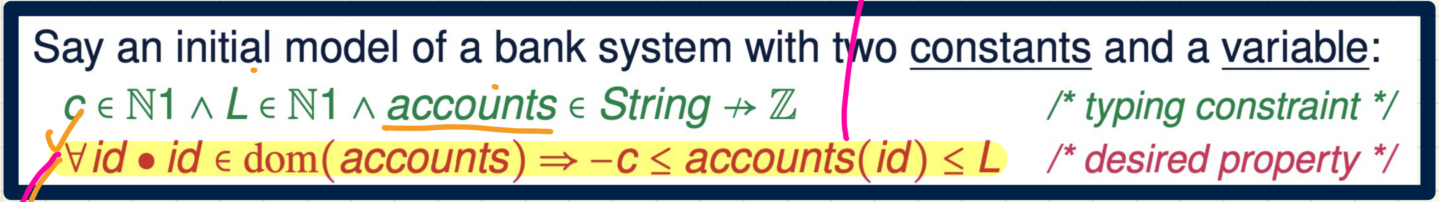
↳ No. Refinement relations are transitive.

(C2) State space allows: {c = 100, L = 200, accounts = {"alan" ↦ -200}} → Is this an axiom or a theorem/invariant?

State Space of a Model

invariant violation: model need to be fixed!

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



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

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

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

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

(C1) axiom: assume to be true (used to restrict the state space)

(C2) theorem/invariant: need to be shown to hold in all possible states

(C1) (c = 100, L = 200, accounts = {("alan", 150), ("mark", 149)}

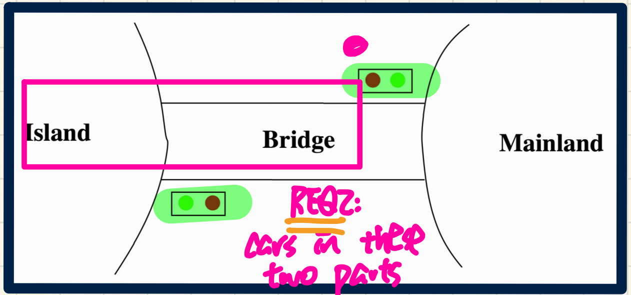
satisfies the axiom, that is no state can violate this

{c = 100, L = 200, accounts = {"alan" ↦ -200}}

should not be even considered! does not satisfy axiom

Bridge Controller:

Requirements Document



REQ2:
cars on these two parts should be limited!

↳ also this REQ makes no distinction on island & bridge.

ENV1	The system is equipped with two traffic lights with two colors: green and red.
ENV2	The traffic lights control the entrance to the bridge at both ends of it.
ENV3	Cars are not supposed to pass on a red traffic light, only on a green one.
ENV4	The system is equipped with four sensors with two states: on or off.
ENV5	The sensors are used to detect the presence of a car entering or leaving the bridge: “on” means that a car is willing to enter the bridge or to leave it.
REQ1	The system is controlling cars on a bridge connecting the mainland to an island.
REQ2	The number of cars on bridge and island is limited.
REQ3	The bridge is one-way or the other, not both at the same time.

without this, verification results would be unrealistic.

encode this by counting # of cars entering or exiting ML.

Lecture

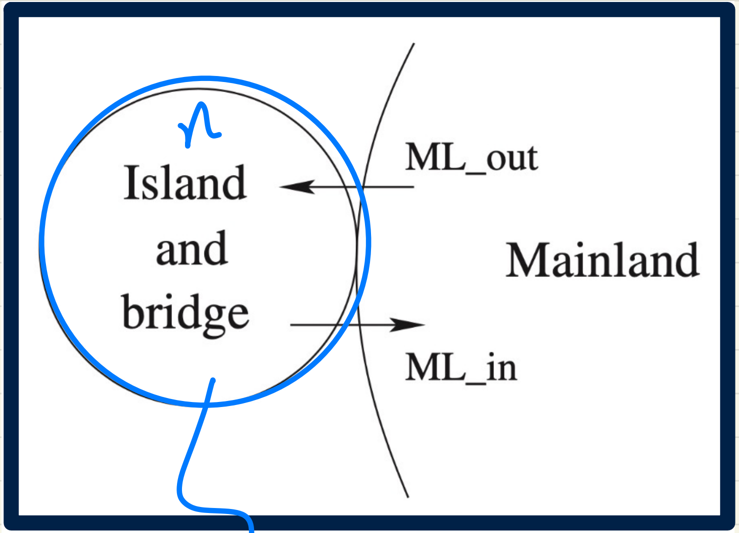
Reactive System: Bridge Controller

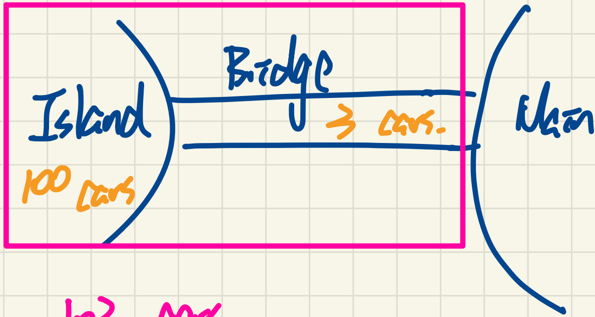
Initial Model: State and Events

Bridge Controller: Abstraction in the Initial Model

n

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





103 cars on the Island-Bridge Compound

the notion of bridge is abstracted away.

Bridge Controller: State Space of the Initial Model

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

Static Part of Model

constants: d

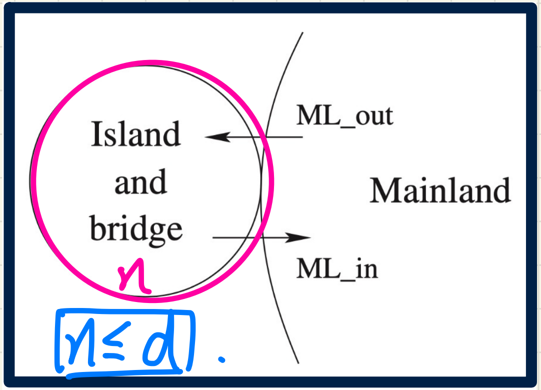
max # of cars in island and bridge. axiom

axioms:

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

model 0 (m0)

axiom 1



Dynamic Part of Model

variables: n

current # of cars in island and bridge

invariants:

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

inv0_2 : $n \leq d$

REQ2

Bridge Controller: State Transitions of the Initial Model

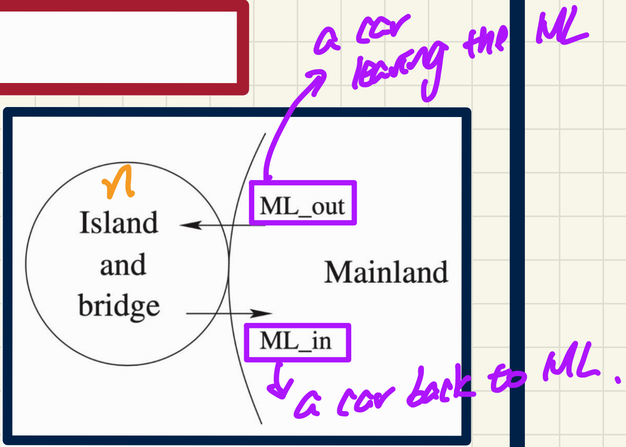
REQ2	The number of cars on bridge and island is limited.

constants: d

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

variables: n

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



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

actions

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

missing guards: implicitly TRUE as guards

↳ events are always enabled.

State Transition Diagram on an Example Configuration

d = 2

n initialized to 0

d = 2
n =

Lecture 10 - February 9

Reactive System: Bridge Controller

Announcements

- Lab2 released
- WrittenTest1 guide released
  - + Verify EECS account on a WSC machine
  - + Verify PPY account and Duo Mobile on eClass

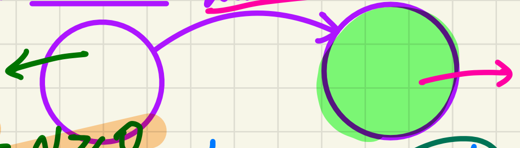
Invariants (I)

= Conditions that must hold true all the time

e.g. Lab1

1. accounts balance is ≥ -C

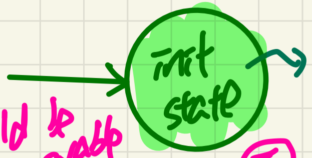
2. v ≥ 0



withdraw action: decrement balance by v.

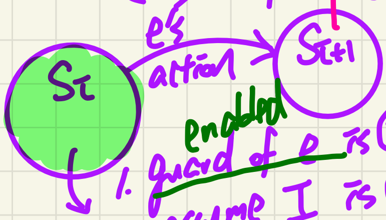
→ account's balance remains ≥ -C

(1) I established after initializing the system.



evaluating I on initial state should be (T)

(2) For every event e, if it's enabled, assuming that I is true, then I is preserved after actions of e take effect.



1. ~~guard of e is (T)~~

2. assume I is (T)

it should be provable that I remains (T)

Bridge Controller: State Transitions of the Initial Model

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

context

constants: d

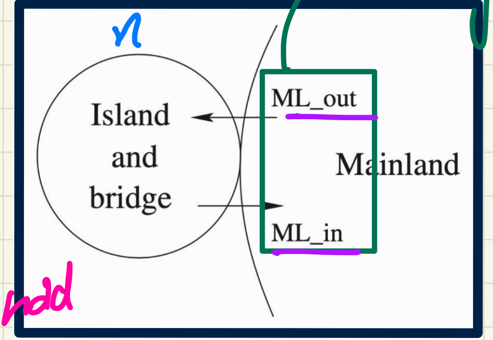
axioms:
axm0_1 : $d \in \mathbb{N}$

machine.

variables: n

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

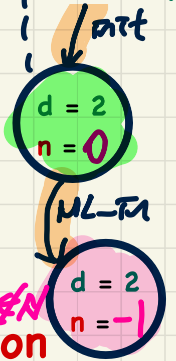
should always hold



occurrences of events change

values of variable(s)

<init, ML_in>



enabled

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

when True

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

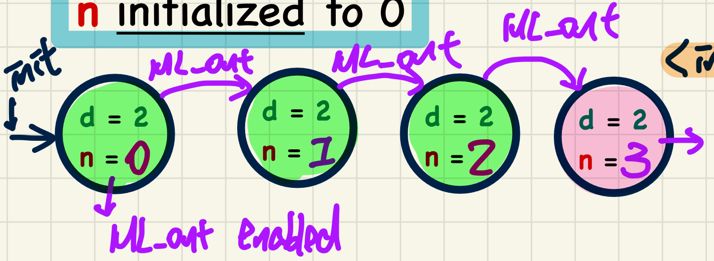
when True

State Transition Diagram on an Example Configuration

d = 2

n initialized to 0

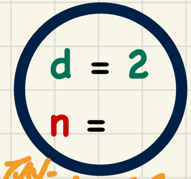
Is ML_out going to maintain I always?



3∈N ✓
3≤2 ✗
(¬I)

<init, ML_out, ML_out, ML_out>

a trace leading to inv. violation.



Before-After Predicates of Event Actions

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

pre-state value

becomes

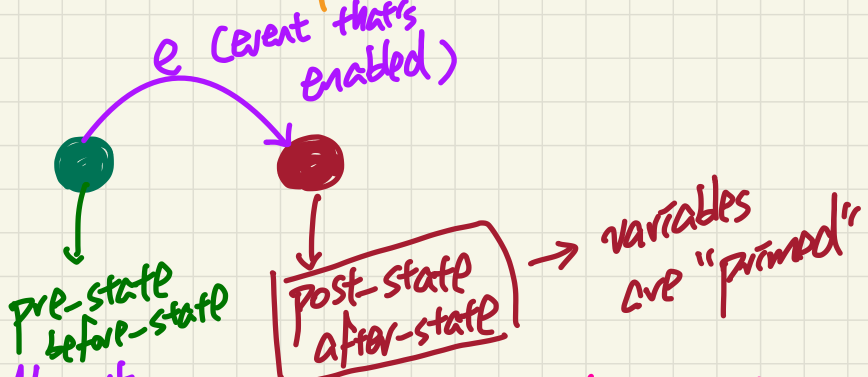
after ML_out's action takes effect, the

post-state value

actions

post-state value of n becomes the pre-state value of n plus 1.

- Pre-State
- Post-State
- Sate Transition



e.g. ML_out

n=2	n'=3

→ $n' = n + 1$ befor-after predicate characterising the effect of event.

event actions

$v \boxed{:=} v + 1$

1. becomes
2. <u>not</u> variable assignment!!

$x, y, temp$

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

X

$v' :=$ <u>not</u> variable assign.

Just:

$x := y$

$y := x$

BAP:

$x' = y$

$\wedge$ $y' = x$

```
evt
  begin
    x := x + 1
    x := x - 1
  end
```

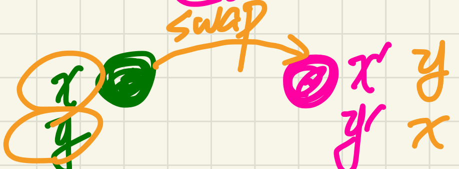
cannot have the same variable as LHS multiple times!

$x' = x + 1$

$\wedge$

$x' = x - 1$

$\equiv$ $\boxed{F}$.



Lecture

Reactive System: Bridge Controller

Initial Model: Invariant Preservation

Design of Events: Invariant Preservation

variables: n → state space

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

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

invariants:

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

inv0_2 : $n \leq d$

$\left.\begin{matrix} \\ \\ \end{matrix}\right] I$

$$\forall \underline{\overset{n}{state}} \cdot \underline{\overset{n \in \mathbb{Z}}{state \in StateSpace}} \Rightarrow \underline{\overset{\substack{n \in N \\ \wedge \\ n \leq d.}}{I(state)}}$$

$$\Downarrow$$

witness of violation

$$\neg \exists\, state \cdot state \in StateSpace \wedge \boxed{\neg I(state)}$$

Sequents: Syntax and Semantics

Both H and G are sets of predicates.

Syntax

turnstile.

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

assumptions/ hypotheses

goal conclusion

Semantics

$$\underbrace{H \vdash G}_{} \iff H \Rightarrow G$$

G is provable, given H ← T or F → G is not provable given H

↳ assuming H, G should be provable.

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

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

$\vdash G \overset{?}{=} \text{False} \vdash G$

$\equiv \text{False} \Rightarrow G \equiv \text{True}.$ ✗

not appropriate

$\vdash G \equiv \text{True} \vdash G$

$\equiv \text{True} \Rightarrow G \equiv G$

PO/VC Rule of Invariant Preservation

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

BAP: $n' = n+1$

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

BAP: $n' = n-1$

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ True \\ \vdash \\ \overset{n-1}{\cancel{n}} \in \mathbb{N} \\ \wedge \\ \overset{n-1}{\cancel{n}} \leq d \end{array}$$

ML_in

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \leq d \\ True \\ \vdash \\ \overset{n+1}{n'} \in \mathbb{N} \\ \wedge \\ \overset{n+1}{n'} \leq d \end{array}$$

ML_out

assumed to be true

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

INV

should be provable

Lecture 11 - February 14

Reactive System: Bridge Controller

Announcements

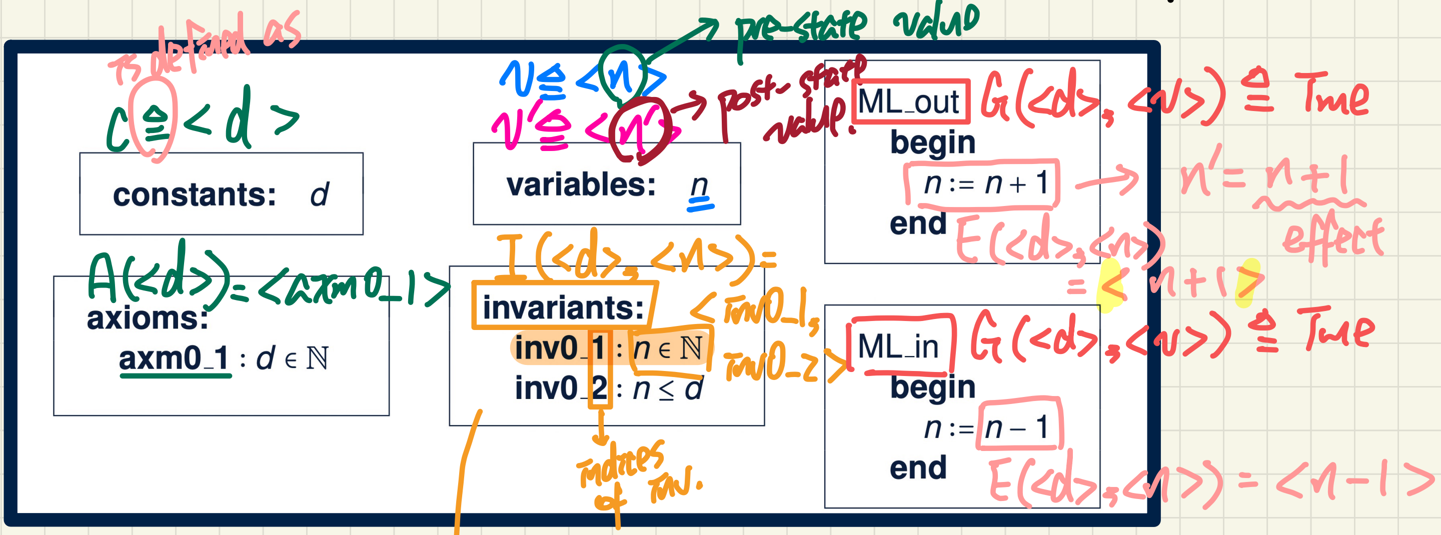
- Lab2 released
- WrittenTest1 guide released
  - No Radix
  - given Radix syntax → answer
  - write Radix syntax
  -

  + Verify EECS account on a WSC machine
  + Verify PPY account and Duo Mobile on eClass
- Review Session at 7pm, Wednesday? (Zoom)

  confirmed

Θ: event paramepters ?

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 : $d \in \mathbb{N}$

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

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

variables: n

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

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

indices of inv.

ML_out $G(\langle d \rangle, \langle v \rangle) \triangleq True$
begin
$n := n + 1$ → $n' = n + 1$ effect
end
$E(\langle d \rangle, \langle n \rangle) = \langle n+1 \rangle$

ML_in $G(\langle d \rangle, \langle v \rangle) \triangleq True$
begin
$n := n - 1$
end
$E(\langle d \rangle, \langle n \rangle) = \langle n-1 \rangle$

$I_1(\langle d \rangle, \langle n \rangle) \triangleq n \in N$

c: list of constants

A(c): list of axioms

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

I(c, v): list of invariants

constants variables

G(c, v): guards of an event’s

E(c, v): effect of an event’s actions

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

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

PO/VC Rule of Invariant Preservation: Sequents

constants: d

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

variables: n

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

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

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

$$A(c)$$

axioms

$$I(c, v)$$

all invariant conditions

invariant in pre-state

$$* \; G(c, v)$$

guard of some event is evaluated True in pre-state

$$\vdash$$

$$** \; I_i(c, E(c, v))$$

index of invariants (1 or 2)

invariant I should be expr. using the effect of the event.

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

* guard of some event → # of events (2)

** some invariant condition → # of invariant conditions. (2)

Overall: 2 * 2 = 4 POs

PO1 : ML_out / inv0_1 / INV

evt name   inv. cond.   nature of PO

PO2 : ML_out / inv0_2 / INV

PO3, PO4 : Exercise!

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
  begin
    n := n + 1
  end
```

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

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

effect of an event only occurs if event is enabled

effect of ML_out

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

ML_out/inv0_1/INV

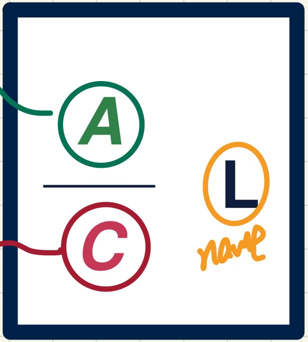
Exercise: Formulate PO3, PO4

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

ML_out/inv0_2/INV

Inference Rule: Syntax and Semantics

Syntax



antecedent
↳ a set of sequents

consequent
↳ a single sequent

name

Semantics

$$A \overset{T①}{\Rightarrow} \underset{T②}{\underline{C}} \equiv \text{True}$$

② must be true

Sequent

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

provable or not.

Inference Rule

$$\boxed{\frac{A}{C}} \quad A \Rightarrow C \equiv \text{True}.$$

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

$$\boxed{\frac{}{C}}$$

$$\underline{\underline{\text{True}}} \Rightarrow C \equiv \text{True}$$

↳ $C \equiv \text{True}$

the consequent itself is an axiom.

Examples

IR1

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

To prove $H_1, H_2 \vdash G$,

it's sufficient to prove (by dropping a hypothesis):

$H_1 \vdash G$

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

$$\frac{A}{C}\ \underline{\text{Mon}}$$

→ monotonicity.

→ To prove C, it's sufficient to prove A

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

Mon

$$\boxed{\begin{array}{l} n \in \mathbb{N} \\ \vdash \\ n+1 \in \mathbb{N} \end{array}}\ \text{P2}$$

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

nothing to prove for the consequent.

Justifying Inference Rule: OR_L

$$\frac{\boxed{\cancel{H}, P \vdash R \qquad \cancel{H}, Q \vdash R}\; A}{\boxed{\cancel{H}, P \vee Q \vdash R}\; C} \;\text{OR\_L}$$

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

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

(demo video).

Example Inference Rules

terminating rules.

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

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

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

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

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

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

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

to the < of ⊢

non-terminating.

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

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

to the R of ⊢

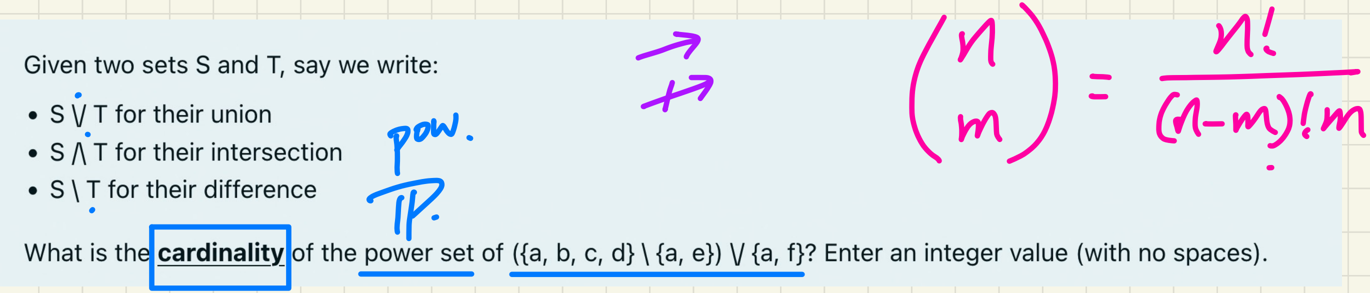
disjunction

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

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

Wednesday, February 15

Written Test 1 Review



Given two sets S and T, say we write:

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

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

pow.

$\mathbb{P}$.

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

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

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

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

$$\binom{5}{3} = \binom{5}{2}$$

$$\binom{5}{2} = \frac{5*4}{2!} = \ldots$$

How many subsets in $\mathbb{P}$ are of card 2?

$$\binom{5}{2} = \frac{5*4}{2!} = 10$$

$$2^5 = 32$$

$$\boxed{\binom{n}{m} = \binom{n}{n-m}}$$

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

$$\{S \mid S \in \mathbb{P}(\{a, b, c, d, f\}) \wedge |S| = 2\}$$

$\boxed{10}$

$\{a, b\}$

$\{a, c\}$

$\{a, d\}$

$\{a, f\}$

$\{b, c\}$

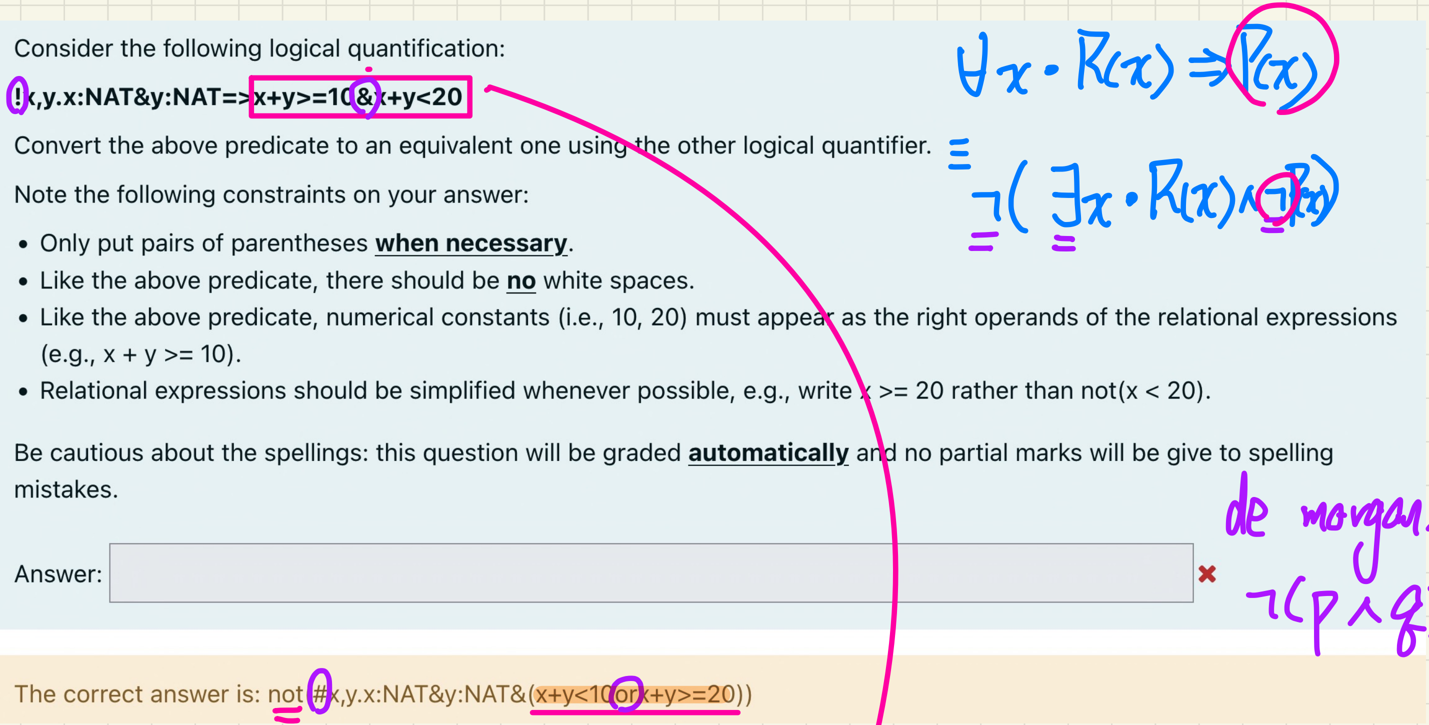
$\{b, d\}$

$\{b, f\}$

$\{c, d\}$

$\{c, f\}$

$\{d, f\}$



Consider the following logical quantification:

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

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

Note the following constraints on your answer:

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

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

Answer: ✖

The correct answer is: not(#x,y.x:NAT&y:NAT&(x+y<10orx+y>=20))

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

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

de morgan:

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

$$\neg(x+y \geq 10 \wedge x+y < 20)$$

$$\equiv \neg(x+y \geq 10) \vee \neg(x+y < 20)$$

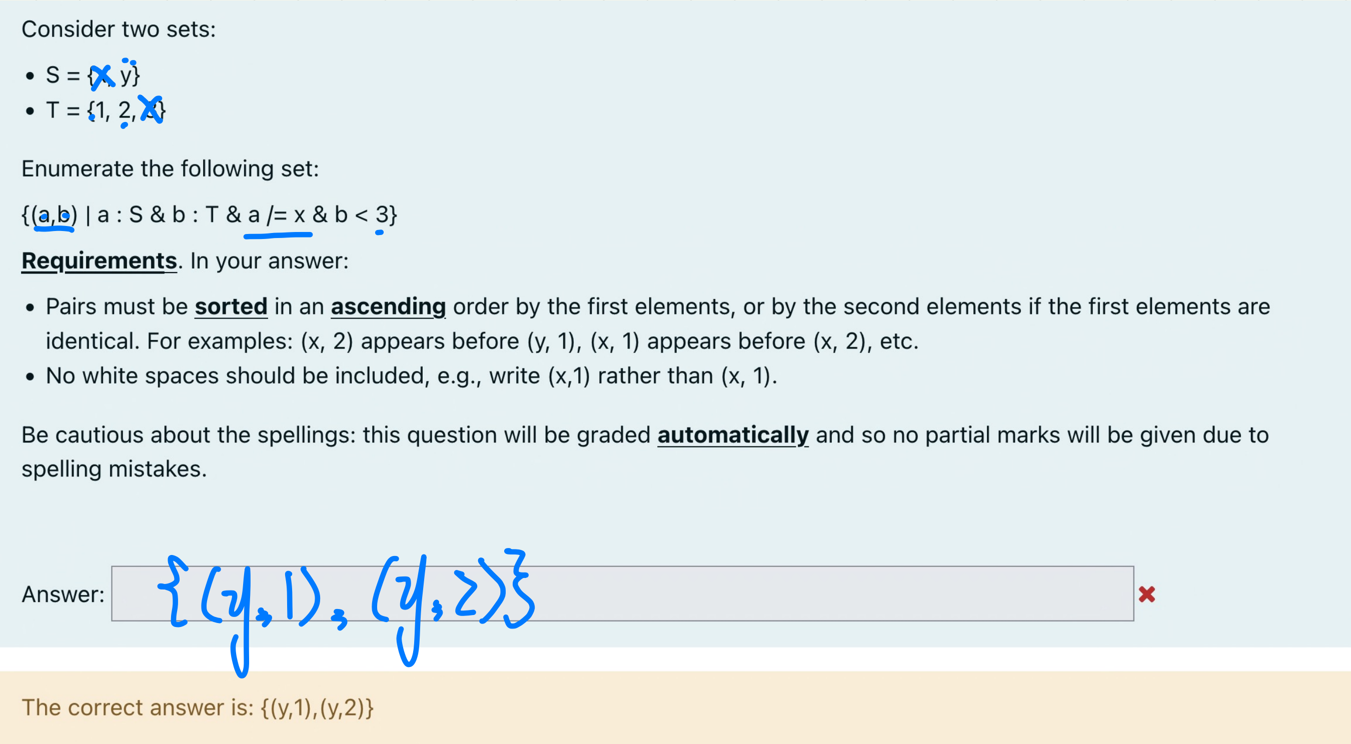
$$\equiv x+y < 10 \vee x+y \geq 20$$

$$\{a,b,c,d\} \lhd \{(\underline{\underline{a}},2),(\underline{\underline{b}},3)\} = \{(a,2),(b,3)\}$$

$$S \lhd R = \{(x,y) \mid \boxed{(x,y) \in R} \land x \in S\}$$

only consider whats in R

((



Consider two sets:

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

Enumerate the following set:

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

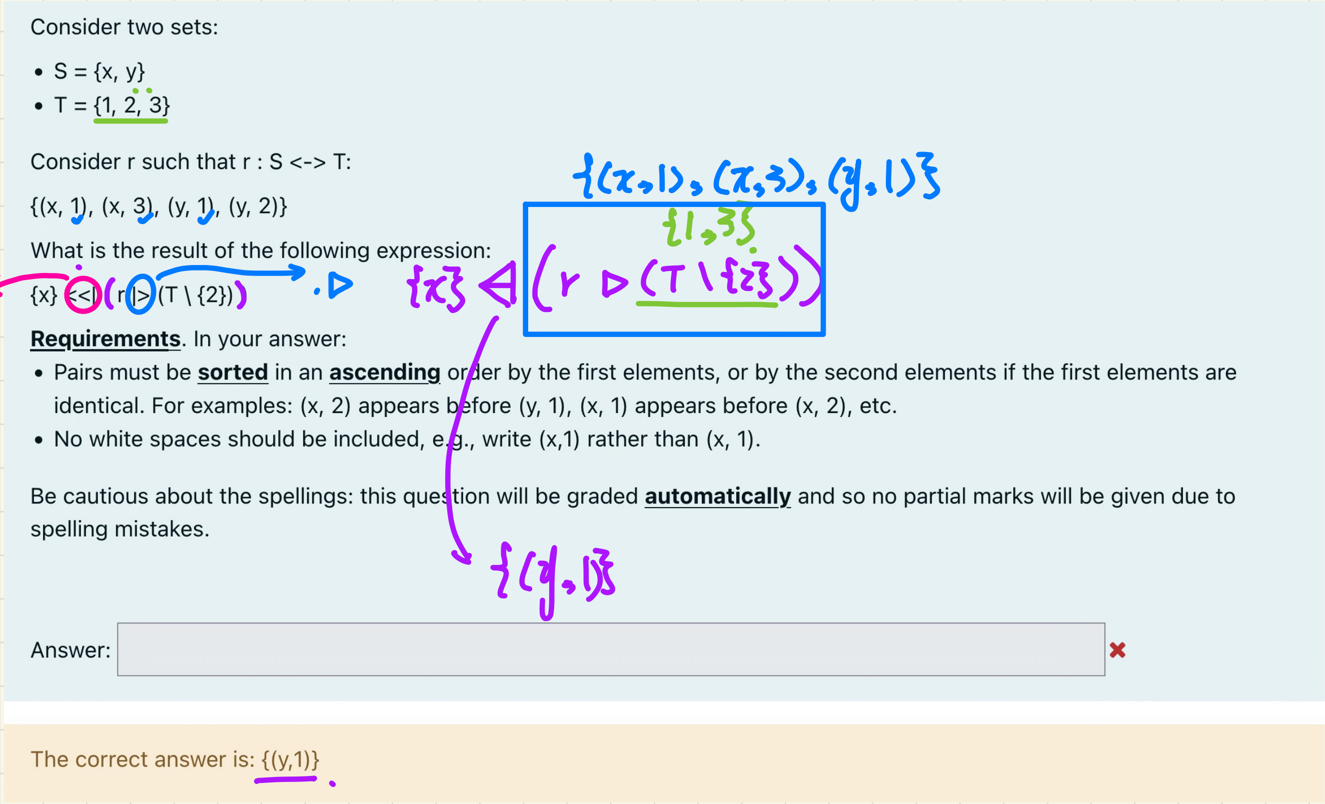
Requirements. In your answer:

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

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

Answer: {(y,1),(y,2)} ✖

The correct answer is: {(y,1),(y,2)}



Consider two sets:

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

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

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

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

What is the result of the following expression:

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

⩤ · ▷

$\{x\} ⩤ \boxed{(r \rhd \overset{\{1,3\}}{(T \setminus \{2\})})}$

$\{(y,1)\}$

Requirements. In your answer:

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

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

Answer: ✖

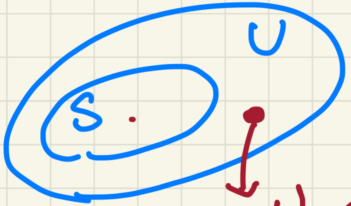
The correct answer is: {(y,1)}

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

$T = \{1, 3\}$

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

$S \subset U$



$\in U \notin S$

✓ Subset

$\underline{\underline{S}} \overset{\times}{\subseteq} \underline{\underline{T}}$ (3).

$T \overset{✓}{\subseteq} S$ $\quad$ $S \overset{✓}{\subseteq} U$ $\quad$ $U \overset{✓}{\subseteq} S$

Proper subset

$S \overset{\times}{\subset} T$ $\quad$ $T \overset{✓}{\subset} S$ $\quad$ $S \overset{\times}{\subset} U$ $\quad$ $U \overset{\times}{\subset} S$

$$S \subset T \iff S \subseteq T \land |S| \overset{\times}{<} |T|$$

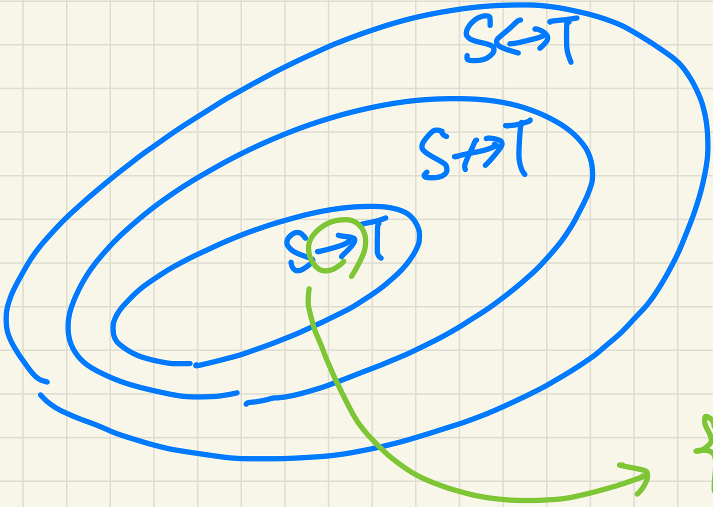
$\{a,b\}$ $\{1,2,3\}$

$r \in S \leftrightarrow T$

r satisfies functional property

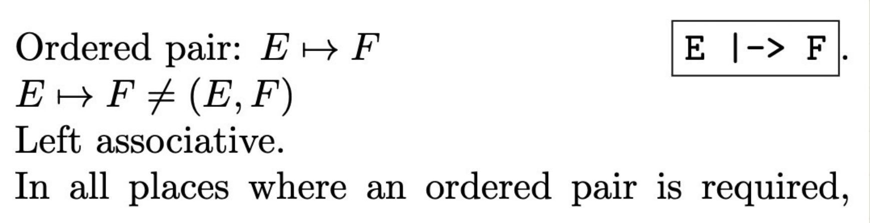
↳ r is a partial function

↳ only those partial functions whose domain is S are total



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

↳ total, not injective.



Ordered pair: $E \mapsto F$ $\boxed{\texttt{E |-> F}}$.

$E \mapsto F \neq (E, F)$

Left associative.

In all places where an ordered pair is required,

clarify after reading week

Lecture 12 - February 28

Reactive System: Bridge Controller

Announcements

- Released: WrittenTest1, Lab2 solution
- To be released:
  - + ProgTest1 Guide (by the end of Wednesday)
  - + ProgTest1 practice questions (by Thursday class)

Recap of Previous Classes

Req. Doc.

+ Before-After Predicates
+ Example IRs

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

constants: d

variables: n

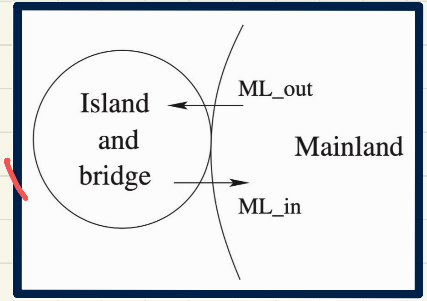
axioms:
axm0_1 : $d \in \mathbb{N}$

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

ML_out
begin
$n := n + 1$
end

$n' = n + 1$

ML_in
begin
$n := n - 1$
end



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

sequent

$H \overset{?}{\Rightarrow} G$

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

Proof obligation

ML_out/inv0_1/INV

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

inference rule.

$$\frac{A}{C} \; \mathbf{L}$$

1. $A \Rightarrow C \equiv$ True

2. To prove C, it's sufficient to prove A

ART : basic arithmetic

Discharging POs of original m0: Invariant Preservation

ML_out/inv0_1/INV

$d \in \mathbb{N}$

$n \in \mathbb{N}$ H1

$n \leq d$

$\vdash$

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

X P2 (∵ too many hypotheses).

MON

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

P2

ML_in/inv0_1/INV

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

$\vdash$

$n - 1 \in \mathbb{N}$ → $n-1 \geq 0$

$n \geq 1$ $(n > 0)$

MON

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

?? may need to add a guard to ML_in.

ML_out/inv0_2/INV

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

$\vdash$

$n + 1 \leq d$

MON

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

?? may need to add a guard to ML_out

ML_in/inv0_2/INV

$d \in \mathbb{N}$

$n \in \mathbb{N}$

$n \leq d$

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

$n - 1 \leq d$

DEC X (can't apply directly)

ARI

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

MON

$$\boxed{\begin{array}{l} n \leq d \\ \vdash \\ n-1<d \lor n-1=d \end{array}}$$

OR_R1

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

DEC

$n-1 \leq d$ vs. $n-1 < \overset{d}{m}$

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

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

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

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

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
$n := \boxed{n + 1}$ $\le d$
end

ML_in
when $n > 0$
begin
$n := \boxed{n - 1}$ $\overset{?}{\ }$ $n \in \mathbb{N}$
end

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

ML_in/inv0_1/INV

$d \in \mathbb{N}$
$n \in \mathbb{N}$
$n \le d$ → $n > 0$
$\vdash$
$\boxed{n - 1 \in \mathbb{N}}$ → $n - 1 \ge 0$
$n \ge 1$ $(n > 0)$

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

ML_out/inv0_2/INV

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

MON
$\boxed{n \le d}$
$\vdash$
$n + 1 \le d$
??

Exercise

Discharging POs of revised m0: Invariant Preservation

ML_out/inv0_1/INV

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

ML_in/inv0_1/INV

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

ML_out/inv0_2/INV

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

ML_in/inv0_2/INV

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

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

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

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

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

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

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

Model

↳ static : constants, axioms

↳ dynamic : variables, invariants

Q: Is this model correct (w.r.t. inv. preservation)

sequents formulating the POs of inv. preservation

↳ any unprovable sequents

↳ fix model →

re-generate sequents

↳ prove again.

Lecture

Reactive System: Bridge Controller

Initial Model: Invariant Establishment

Initializing the System

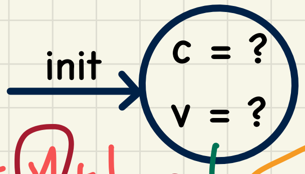
diff pre-states for evt. occurrence

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

resulting post states

Analogy to Induction:

Base Cases ≈ Establishing Invariants



right after vars init.

a single state to prove.

only

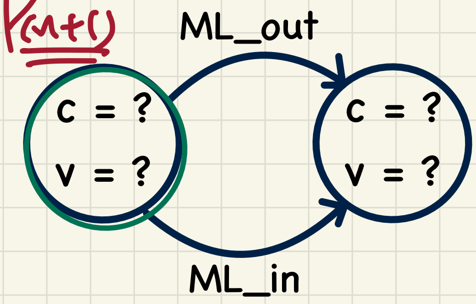
e.g. $n := \cancel{n}+1$

not appropriate

Analogy to Induction:

Inductive Cases ≈ Preserving Invariants

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



The Initialization Event

no pre-state ← init → post-state

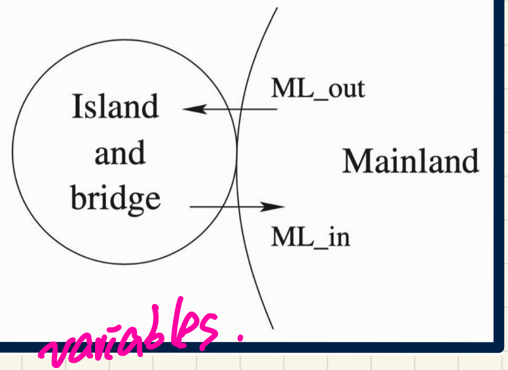
```
init
begin
  n := 0
end
```

name of evt.

variables only

when this evt occurs, none of the variables have been initialized

↳ RHS of := should not refer to variables.



PO of Invariant Establishment

constants: d

variables: n

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

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

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

$n' = 0$

Compare with effect of a non-init event: $E(c, \underline{v})$

Components

$\boxed{K(c)}$: effect of init's actions

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

Rule of Invariant Establishment

$$\dfrac{\begin{matrix} A(c) \\ \vdash \\ I_i(c, K(c)) \end{matrix}}{} \quad \textbf{INV}$$

axioms

Exercise:

Generate Sequents from the INV rule.

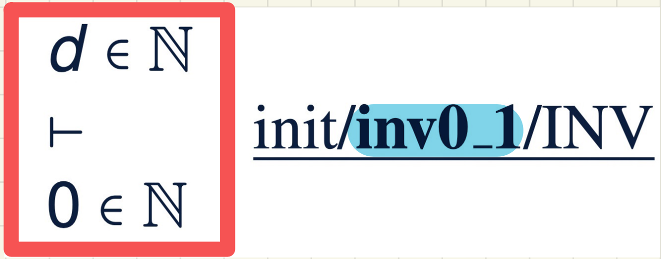
$$\boxed{\begin{matrix} d \in \mathbb{N} \\ \vdash \\ \underset{0}{\cancel{n'}} \in \mathbb{N} \end{matrix}}$$

init/inv0_1/INV

$$\boxed{\begin{matrix} d \in \mathbb{N} \\ \vdash \\ \underset{0}{\cancel{n'}} \leq d \end{matrix}}$$

init/inv0_2/INV

Discharging PO of Invariant Establishment



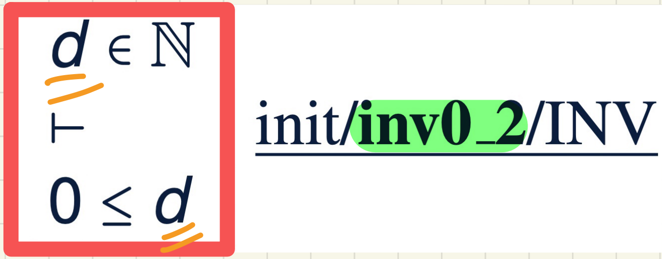
$$\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ 0 \in \mathbb{N} \end{array}$$

init/**inv0_1**/INV

MON

$$\boxed{\begin{array}{l} \vdash \\ 0 \in \mathbb{N} \end{array}}$$

P1

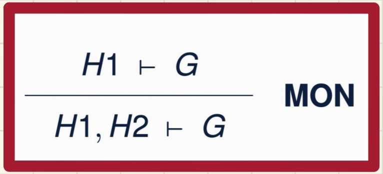


$$\begin{array}{l} d \in \mathbb{N} \\ \vdash \\ 0 \leq d \end{array}$$

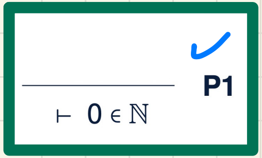
init/**inv0_2**/INV

P3

where n is instantiated by d

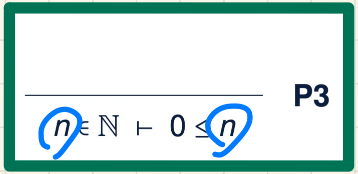


$$\frac{H1 \;\vdash\; G}{H1, H2 \;\vdash\; G} \quad \textbf{MON}$$



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

✓



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

Lecture

Reactive System: Bridge Controller

Initial Model: Deadlock Freedom

want to prove:

system is deadlock-free: $G(ML\_out) \lor G(ML\_in)$

REACTIVE SYSTEMS

↳ deadlocks

↳ no reaction to the user/env.

↳ no events can occur

↳ None of events' guards is satisfied.

deadlock cond.

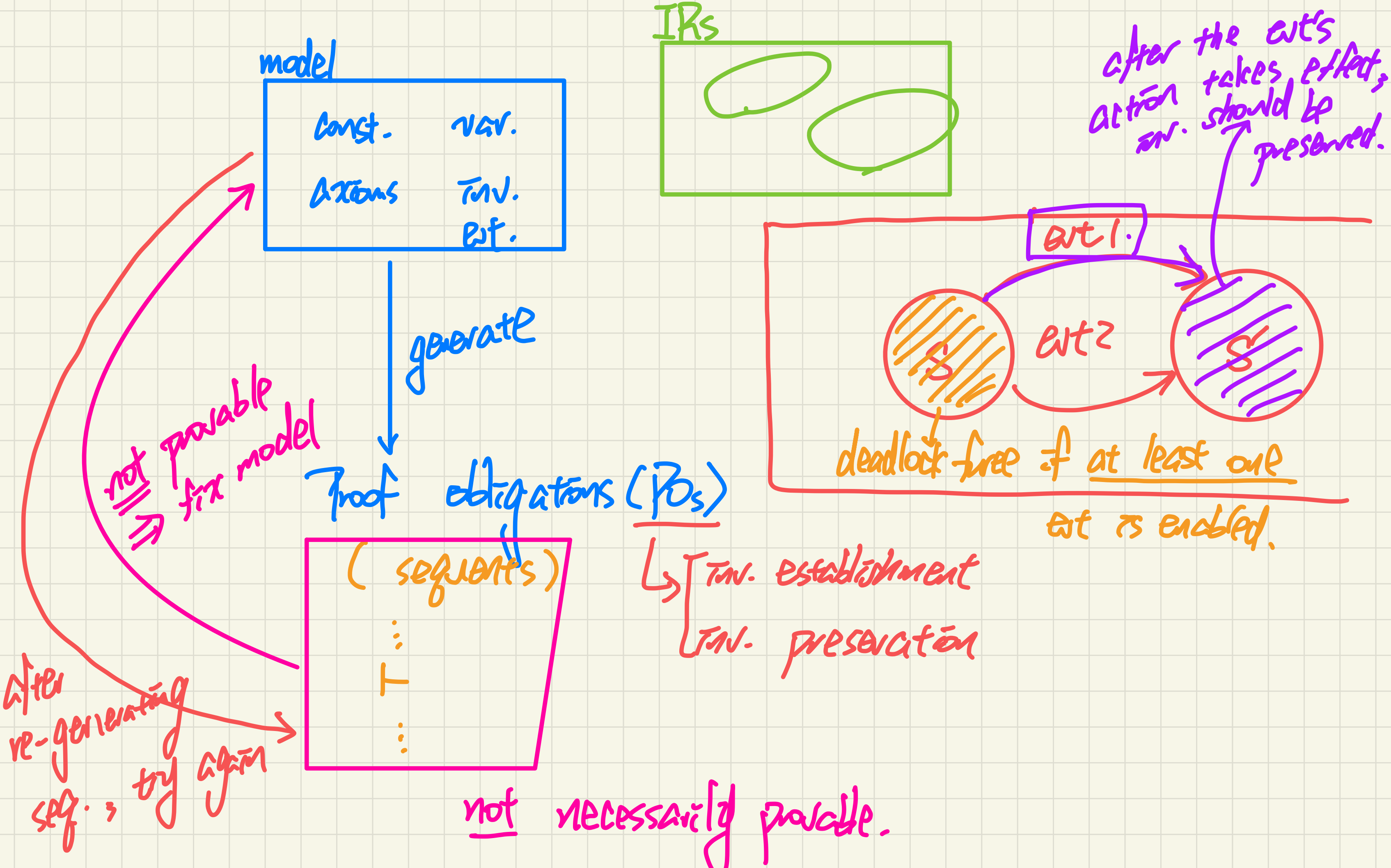
$$\neg(G(ML\_out) \lor G(ML\_in))$$

not the case that some event is enabled.

$$\equiv \neg G(ML\_out) \land \neg G(ML\_in)$$

Lecture 13 - March 2

Reactive System: Bridge Controller



- Inv. establishment
- Inv. preservation

not necessarily provable.

not provable ⇒ fix model

after re-generating seq.; try again

after the evt's action takes effect, inv. should be preserved.

deadlock-free if at least one evt is enabled.

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 : $n \le d$

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

m=2

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

$A(c)$ axioms

$I(c, \mathbf{v})$ invariant held at pre-state

$\vdash$

$G_1(c, \overset{\text{pre-state}}{\mathbf{v}}) \vee \cdots \vee G_m(c, \mathbf{v})$

DLF

↳ disjunction of guards of all events held True

- c: list of ***constants*** $\langle d \rangle$
- $A(c)$: list of ***axioms*** $\langle$**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*** $\langle$**inv0_1**, **inv0_2**$\rangle$
- $G(c, v)$: the event's ***guard***

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

Exercise: Generate Sequent from the DLF rule.

$$\begin{array}{l} d \in \mathbb{N} \\ n \in \mathbb{N} \\ n \le d \end{array} \vdash \underbrace{n<d}_{G_{ML\_out}} \vee \underbrace{n>0}_{G_{ML\_in}}$$

PO	(v) pre-state	(v') post-state
INV est.	✗	✓
INV pre.	✓	✓
DLF	✓	✗

Example Inference Rules

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

appears to the left of $\vdash$

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

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

H(E) replaced for free occurrences of E by F

application from L to R

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

$P \Rightarrow (E = E)$

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

application from R to L

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

$$\frac{H(E), E = F \vdash P(E)}{H(F), \underset{F}{\cancel{E}} = \underset{E}{\cancel{F}} \vdash P(F)} \quad \textbf{EQ\_}\underset{LR}{\cancel{\textbf{RL}}}$$

Discharging PO of DLF: First Attempt

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

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

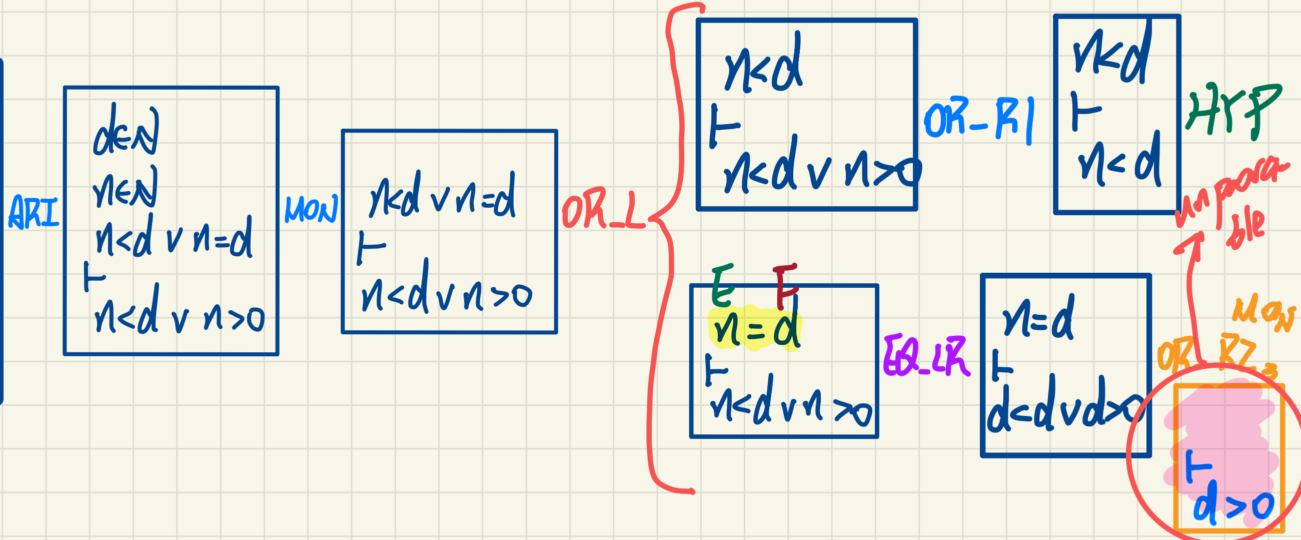
$$\frac{H \vdash P}{H \vdash P \lor Q}\ \textbf{OR\_R1}$$

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

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

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

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



Understanding the Failed Proof on DLF

constants: d

variables: n

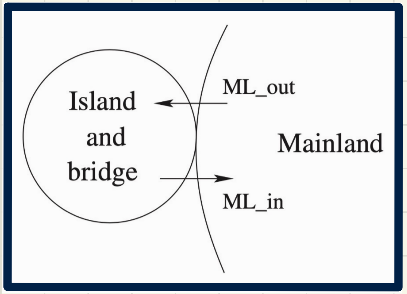
axioms:
axm0_1 : $\boxed{d \in \mathbb{N}}$ ✓

fix: axm0_2: $\boxed{d > 0}$

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

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

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



→ **Unprovable Sequent:** $\vdash \boxed{d > 0}$ may be violated

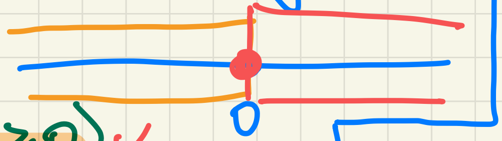
↳ its negation may be true

$\neg(d > 0)$ is allowed by the current model

↳ ① $d \leq 0$ ✓

② $d \in \mathbb{N}$ $(d \geq 0)$ ✓

↳ $\boxed{d = 0}$.



Say $d = 0$,

after init: $n = 0$

deadlock free: $[\underset{0}{\cancel{n}} < \underset{0}{\cancel{d}} \lor \underset{0}{\cancel{n}} > 0]$

$0 < 0$
$\lor$
$0 > 0$
$\equiv$ F

Discharging PO of DLF: Second Attempt

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

$\equiv$

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

MON

$$\begin{array}{l} d>0 \\ \boxed{\begin{array}{l} n < d \lor n = d \\ \vdash \\ n < d \lor n > 0 \end{array}} \end{array} \quad \textbf{OR\_L} \left\{ \begin{array}{l} \boxed{\begin{array}{l} n < d \\ \vdash \\ n < d \lor n > 0 \end{array}} \quad \textbf{OR\_R1} \quad \boxed{\begin{array}{l} n < d \\ \vdash \\ n < d \end{array}} \quad \textbf{HYP} \\ \\ d>0 \\ \boxed{\begin{array}{l} \underline{n = d} \\ \vdash \\ n < d \lor n > 0 \end{array}} \quad \textbf{EQ\_LR, MON} \quad \boxed{\begin{array}{l} d>0 \\ \vdash \\ d < d \lor d > 0 \end{array}} \quad \textbf{OR\_R2} \quad \boxed{\begin{array}{l} d>0 \\ \vdash \\ d > 0 \end{array}} \quad \textbf{?} \end{array} \right.$$

HYP.

Discharging PO of DLF: Second Attempt

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

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

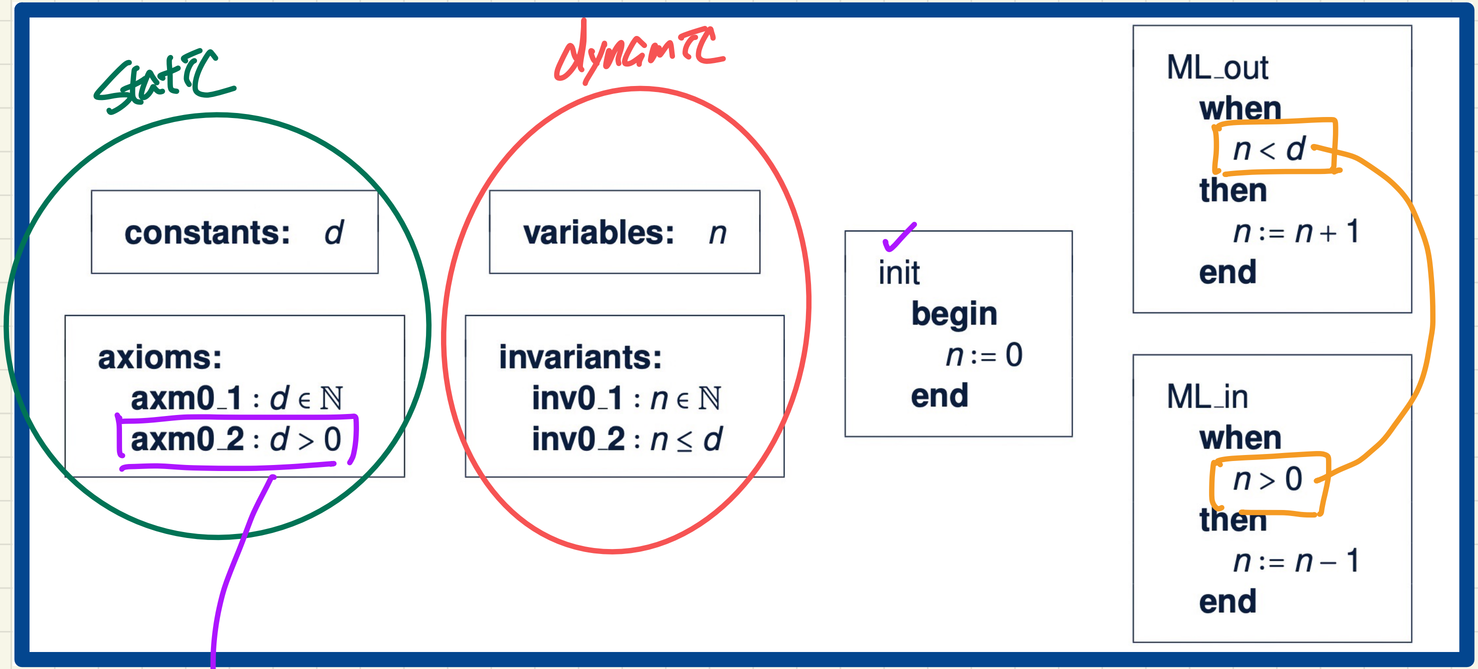
$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, P \lor Q \vdash R}\ \textbf{OR\_L}$$

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

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

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

Summary of the Initial Model: Provably Correct



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
```

Correctness Criteria:

+ Invariant Establishment
+ Invariant Preservation
+ Deadlock Freedom

Monday, March 6

Lab2 Solution Walkthrough

Lab2 Solution: Context Celebrity_c0

CONTEXT Celebrity_c0

CONSTANTS

k knows relation

c celebrity

P Set person

AXIOMS

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

model persons via unique IDs.

axm2: $c \in P$

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

axm4: $\overset{\text{Inv.}}{k^{-1}}[\{c\}] = P \setminus \{c\}$ R3

axm5: $k \cap id = \varnothing$

END R1.

the set of persons by whom c is known

the celebrity is known by everyone, except themselves

the set of possible relations

$(x, y) \in k$

↳ x knows y

$(y, x) \in k^{-1}$ — c

↳ y is known by x

c

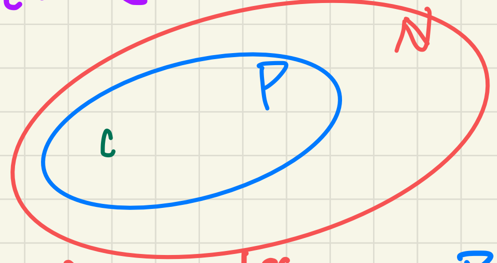
$P = \{$Alan, Mark, Tom$\}$

c = Tom

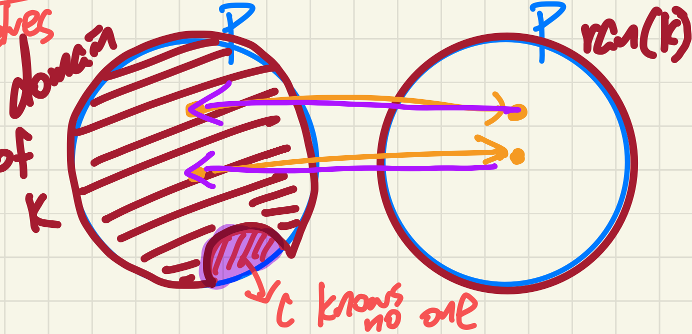
$k = \{$(Alan, Mark), (Alan, Tom), (Mark, Tom)$\}$

$k^{-1} = \{$(Mark, Alan), (Tom, Alan), (Tom, Mark)$\}$

$k^{-1}[\{$Tom$\}] = \{$Alan, Mark$\} = P \setminus \{$Tom$\}$

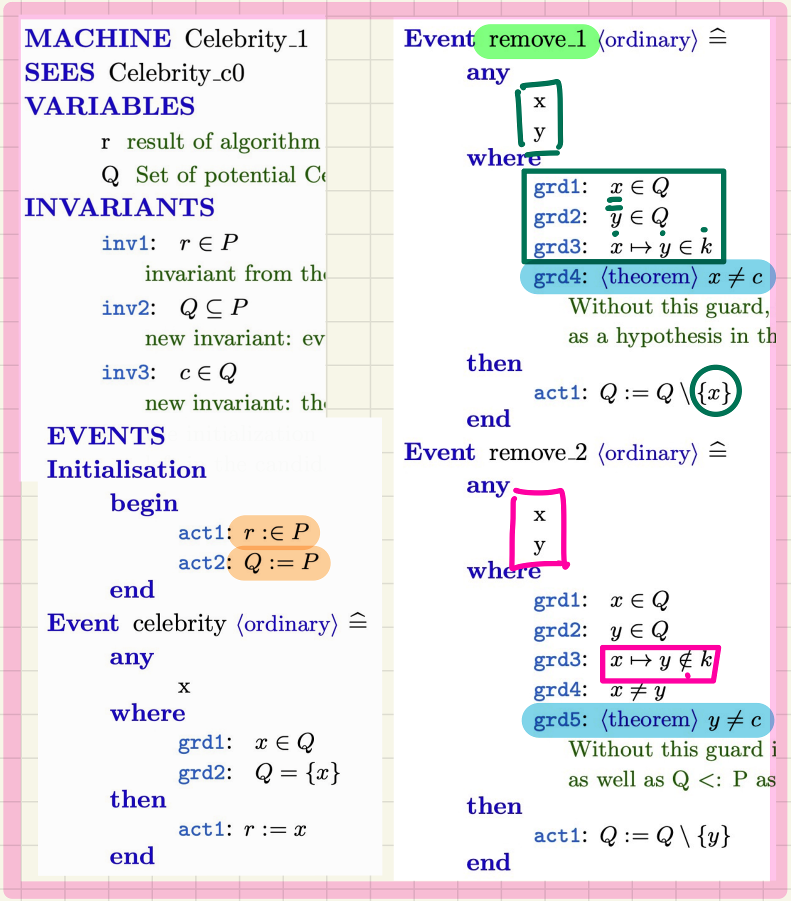


k^{-1}



Lab2 Solution: Machine Celebrity_1

Tip May have to add extra constraints (which may be logically redundant) to guide the prover.



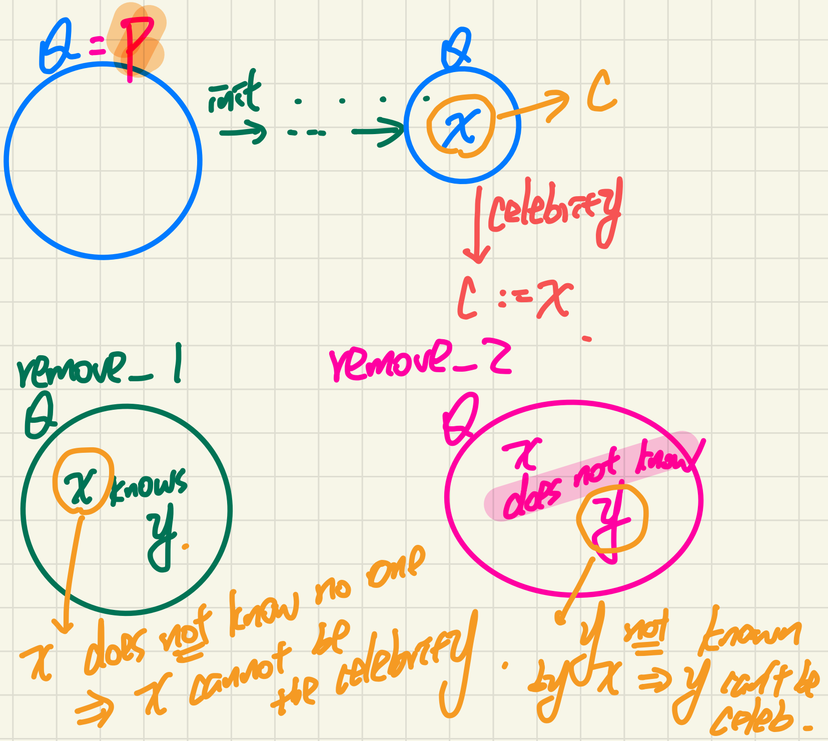
```
MACHINE Celebrity_1
SEES Celebrity_c0
VARIABLES
        r   result of algorithm
        Q   Set of potential Ce
INVARIANTS
        inv1:   r ∈ P
              invariant from th
        inv2:   Q ⊆ P
              new invariant: ev
        inv3:   c ∈ Q
              new invariant: th
  EVENTS
  Initialisation
        begin
              act1: r :∈ P
              act2: Q := P
        end
  Event celebrity ⟨ordinary⟩ ≙
        any
              x
        where
              grd1:   x ∈ Q
              grd2:   Q = {x}
        then
              act1: r := x
        end
```

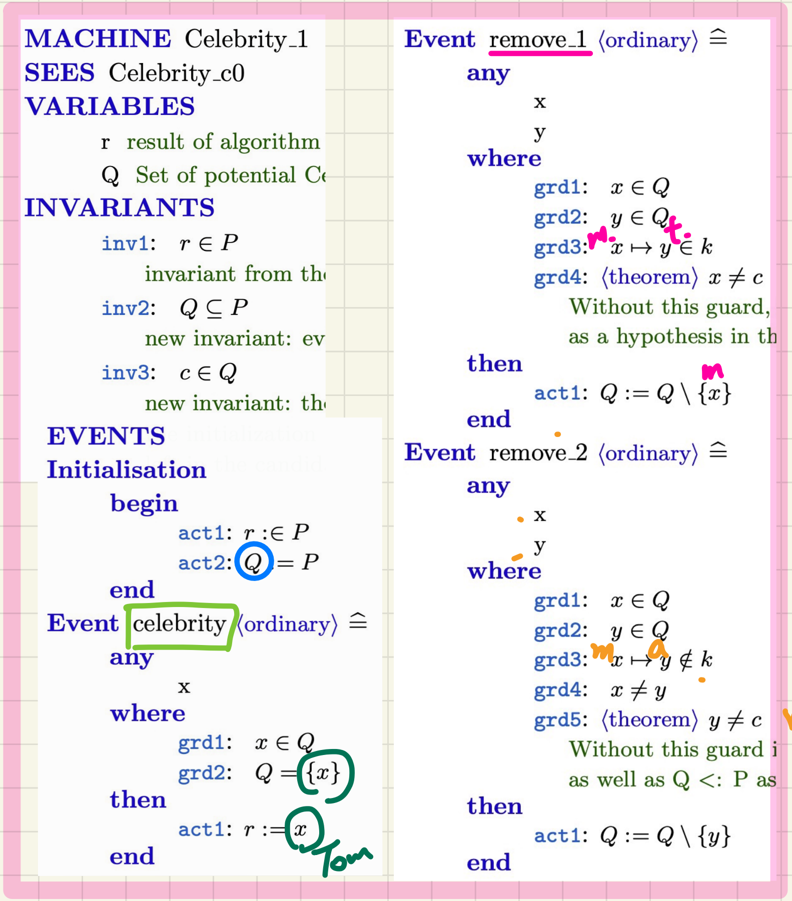
```
Event remove_1 ⟨ordinary⟩ ≙
      any
            x
            y
      where
            grd1:   x ∈ Q
            grd2:   y ∈ Q
            grd3:   x ↦ y ∈ k
            grd4: ⟨theorem⟩ x ≠ c
                Without this guard,
                as a hypothesis in th
      then
            act1: Q := Q \ {x}
      end
Event remove_2 ⟨ordinary⟩ ≙
      any
            x
            y
      where
            grd1:   x ∈ Q
            grd2:   y ∈ Q
            grd3:   x ↦ y ∉ k
            grd4:   x ≠ y
            grd5: ⟨theorem⟩ y ≠ c
                Without this guard i
                as well as Q <: P as
      then
            act1: Q := Q \ {y}
      end
```

P = {Alan, Mark, Tom}

c = Tom

k = {(Alan, Mark), (Alan, Tom), (Mark, Tom)}





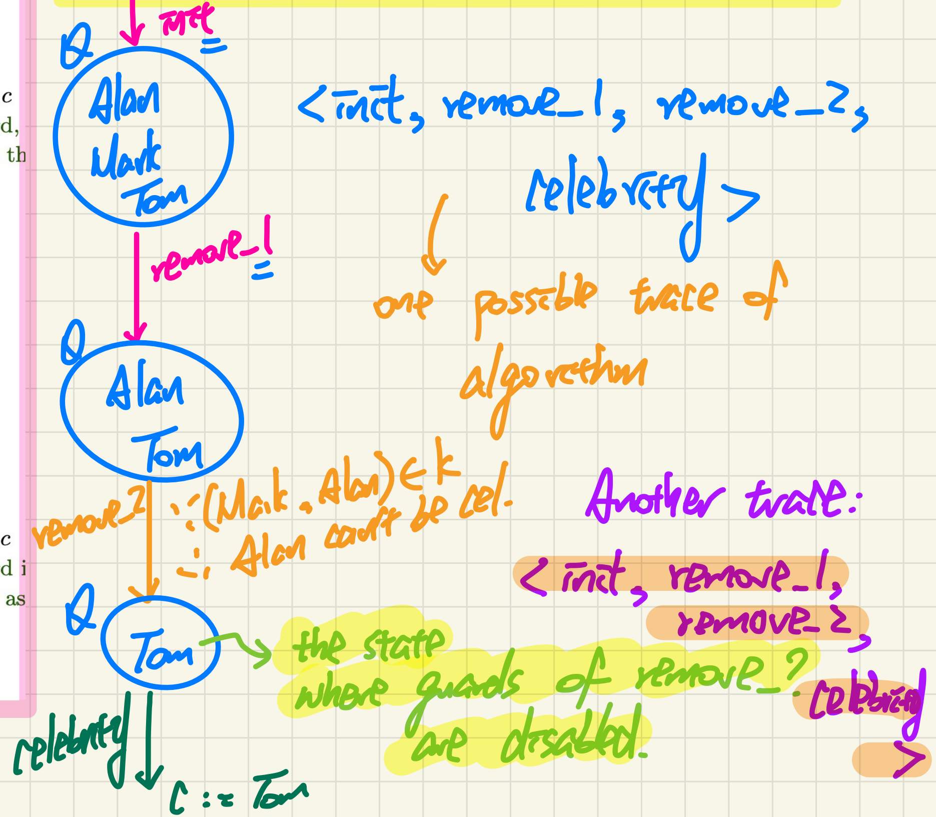
```
MACHINE Celebrity_1
SEES Celebrity_c0
VARIABLES
        r   result of algorithm
        Q   Set of potential Ce
INVARIANTS
        inv1:   r ∈ P
                invariant from th
        inv2:   Q ⊆ P
                new invariant: ev
        inv3:   c ∈ Q
                new invariant: th
EVENTS
Initialisation
        begin
                act1: r :∈ P
                act2: Q := P
        end
Event celebrity ⟨ordinary⟩ ≙
        any
                x
        where
                grd1:   x ∈ Q
                grd2:   Q = {x}
        then
                act1: r := x   Tom
        end
```

```
Event remove_1 ⟨ordinary⟩ ≙
        any
                x
                y
        where
                grd1:   x ∈ Q
                grd2:   y ∈ Q    t.
                grd3: m. x ↦ y ∈ k
                grd4: ⟨theorem⟩ x ≠ c
                      Without this guard,
                      as a hypothesis in th
        then
                                 m
                act1: Q := Q \ {x}
        end
Event remove_2 ⟨ordinary⟩ ≙
        any
                x
                y
        where
                grd1:   x ∈ Q
                grd2:   y ∈ Q
                grd3: m x ↦ y ∉ k   a
                grd4:   x ≠ y
                grd5: ⟨theorem⟩ y ≠ c
                      Without this guard i
                      as well as Q <: P as
        then
                act1: Q := Q \ {y}
        end
```

P = {Alan, Mark, Tom}

c = Tom

k = {(Alan, Mark), (Alan, Tom), (Mark, Tom)}



∵ (Mark, Alan) ∈ k
∴ Alan can't be cel.

the state where guards of remove_? are disabled.

<init, remove_1, remove_2, celebrity>

one possible trace of algorithm

Another trace:

<init, remove_1, remove_2, remove_2, celebrity>

Lecture 14 - March 7

Reactive System: Bridge Controller

Announcements

- Slides updated to include First Refinement
- Released: Lab2 solution video, PracticeTest1 solution
- To be completed by the final exam:

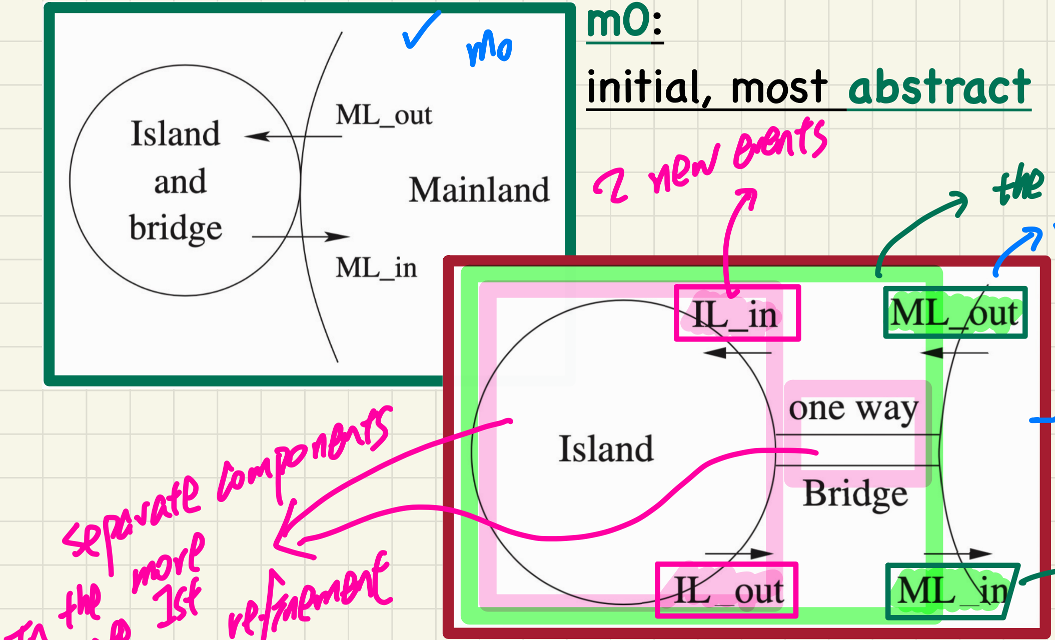
  Makeup lectures for WT1, WT2, ProgTest1, ProgTest2

Lecture

Reactive System: Bridge Controller

First Refinement: State and Events

Bridge Controller: Abstraction in the 1st Refinement



m0:

initial, most abstract

2 new events

the initial abstraction between
→ m1 (no distinction IL and bridge)

m1:

second, more concrete

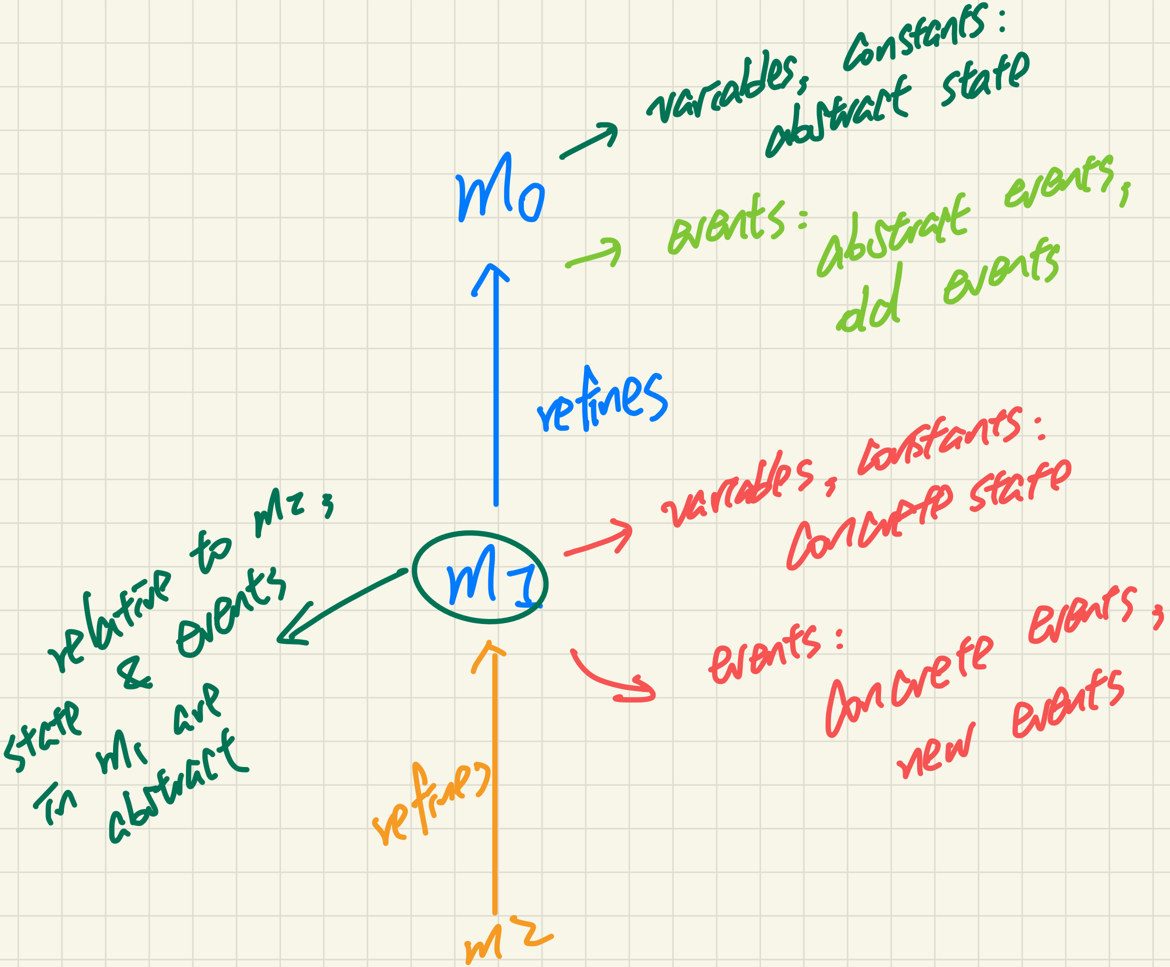
- Both m0 and m1 model the same system.

→ 2 old events

- m1 refines m0 by adding more state vars. & events

separate components in the more concrete 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.



m0 → variables, constants: abstract state

m0 → events: abstract events, old events

m1 → variables, constants: concrete state

m1 → events: concrete events, new events

relative to m2; state & events in m1 are abstract

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.

What's wrong if m1 is constrained by inv1_1 to inv1_4 only?

Witness: a: 2, c: 1, b: 4 → n = 7

Dynamic Part of Model

variables: a, b, c ✓ ✓ ✓

invariants:

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

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

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

inv1_4 : ??

inv1_5 : ??

m1

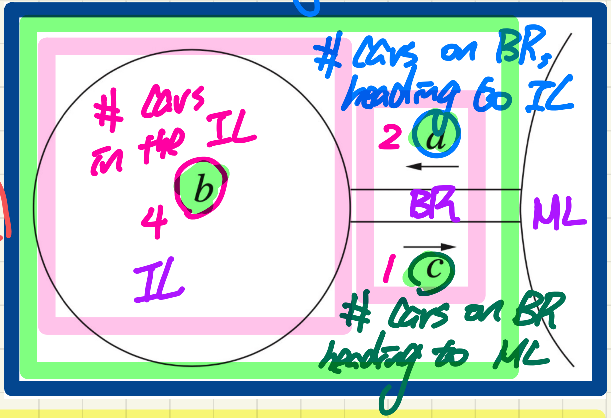
$n = a + b + c$

con. state

abs. state

$a = 0$ ∨ $c = 0$

alt for inv1_5? $\boxed{a * c = 0}$ may work logically, but may be hard to help proofs.



Static Part of Model

constants: d

axioms:

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

axm0_2 : $d > 0$

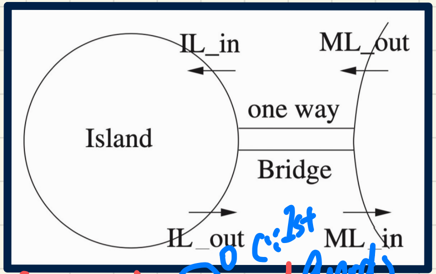
Exercises

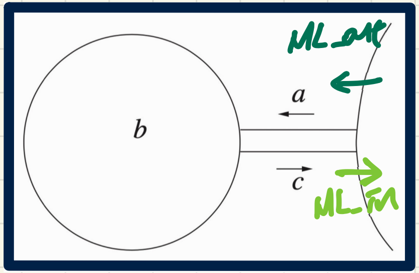
inv1_4: linking abstract & concrete states

n a, b, c

inv1_5: bridge is one-way

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





^{0} c: 1st guard)

③ $(a+1)+b+c \le d$

→ $(a+1)+b \le d$

constants: d

$a+b \le d-1$

$\boxed{a+b < d}$

axioms:

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

axm0_2 : $d > 0$

① From m_0: $\cancel{n}' \le d$

② From m_1: $n' = \cancel{a}+b+c$ → $(a+1)$ effect of ML_out

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 : $a = 0 \lor c = 0$

$$x \le y-1 \equiv x < y$$

x y

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

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

I after

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

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

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

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

Q. Necessary to add a guard "$a = 0$"?

No. ∵ $c > 0 \land (a = 0 \lor c = 0)$ → $a = 0$

States, Invariants, Events: Abstract vs. Concrete

constants: d

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

Abstract m0

variables: n

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

old, abstract events

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

modify abstract state

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

abs. guard and con. guard should be related.

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 : $a = 0 \lor c = 0$

linking inv: involves both abs. & con. vars

Concrete invariants (involving ≥1 con. var).

old, refined events

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

modify concrete state

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

Lecture 15 - March 14

Reactive System: Bridge Controller

Announcements

- ProgTest1 result to be released by Friday
- Lab~~2~~3 to be released by the end of Thursday
- ✓ To be completed by the final exam:

  Makeup lectures for WT1, WT2, ProgTest1, ProgTest2

Before-After Predicates of Event Actions: 1st Refinement

concrete events

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 guards

concrete actions

b, c absent

↳ stay unchanged

- Pre-State
- Post-State
- Sate Transition

Consider an exec: <init, ML_out, ML_in>

Bridge Controller: Abstract vs. Concrete State Transitions

Abstract m0

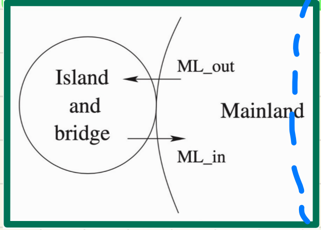
variables: n

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

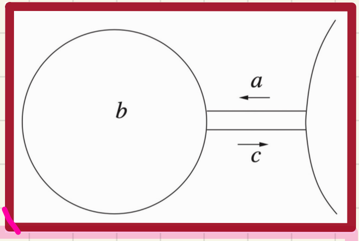
```
ML_out
  when
    n < d
  then
    n := n + 1
  end
```

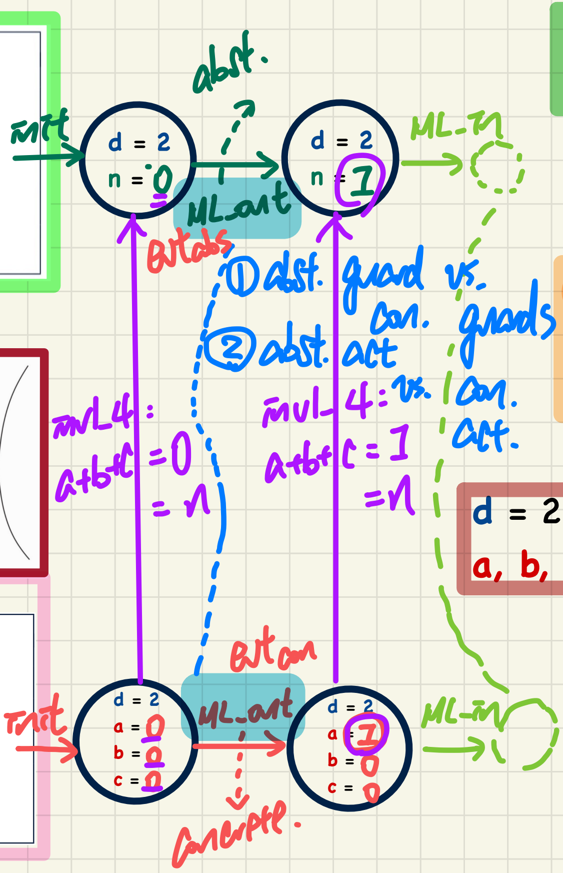
```
ML_in
  when
    n > 0
  then
    n := n − 1
  end
```

d = 2
n initialized to 0



? ?





Scenario
- car leaving ML
- car entering ML

d = 2
a, b, c initialized to 0

evt_con is simulated by evt_abs

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
```

PO Rule of Invariant Preservation in Refinement: Components

Abstract m0

v

v' → n'

variables: n

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

abs. inv.

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

abs. effect

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

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

Concrete m1

w: $\langle a, b, c \rangle$

w': $\langle a', b', c' \rangle$

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

con. inv.

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

con. effect

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

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

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

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

$F(\langle d \rangle, \langle a, b, c \rangle)$ of ML_out : $\langle a+1, b, c \rangle$

Lecture

Reactive System: Bridge Controller

First Refinement: Guard Strengthening

$P \Rightarrow q$

↳ "P is stronger than q"

↳ "q is weaker than P"

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

$q(x) = \{x \mid q(x)\}$

sets of satisfying values

① → $q(x) \subseteq P(x)$?

② $p(x) = q(x)$

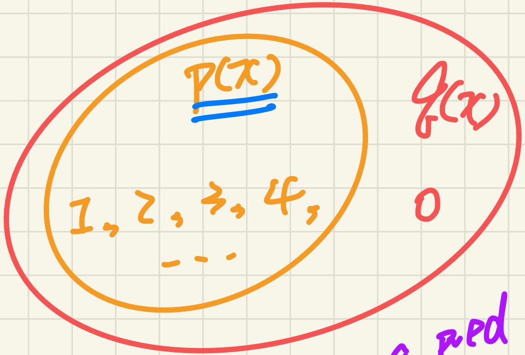
③ $\boxed{p(x) \subseteq q(x)}$ * ?

④ non-overlapping.

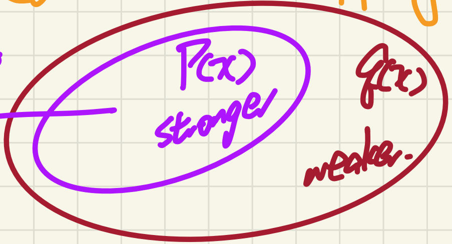
$P(x) = x > 0$

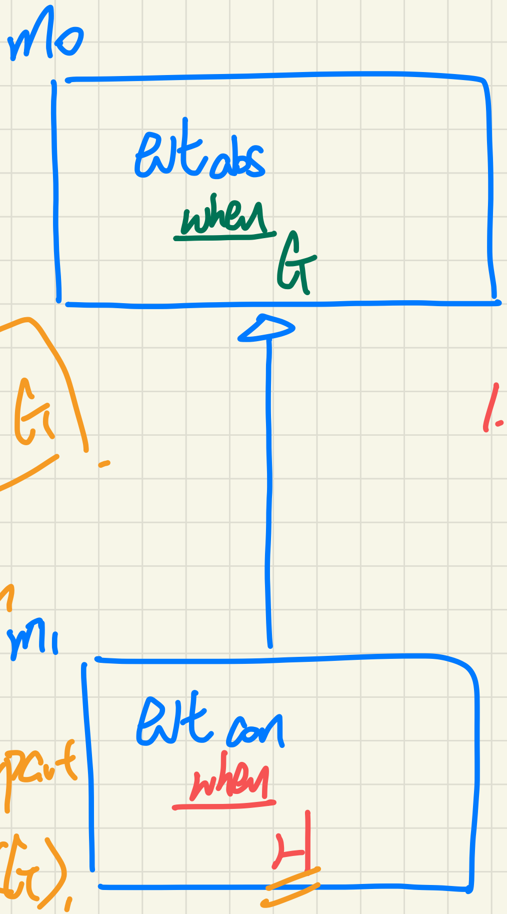
$q(x) = x \geq 0$

$P(x) \Rightarrow q(x)$ ✓



the stronger a pred is, the more values it filters out





$H \Rightarrow G$

① If a conc. transition is enabled (H), then its abs. counterpart is also enabled (G).

② $\neg G \Rightarrow \neg H$

What's <u>not</u> allowed in the abs. transition is also <u>not</u> allowed for con. transition. (In the con. model, <u>no</u> new behaviour is created)

1. When a refinement is created, guards of each event can only be ==strengthened==/ ==stronger==

PO/VC Rule of Guard Strengthening: Sequents

Abstract m0

variables: n

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

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

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

$A(c)$ axioms
$I(c, \mathbf{v})$ abs. invariants
$J(c, \mathbf{v}, \mathbf{w})$ con. invariants
$\boxed{H(c, \mathbf{w})}$ concrete guard
$\vdash$ ..
$\boxed{G_i(c, \mathbf{v})}$ abstract guard

i: index of abs. guard.

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
```

Exercise: Formulat guard strengthening PO of ML_in.

ML_in/GRD

ML_out/GRD

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

$\vdash n < d$

con. guards of ML_out

Q. How many PO/VC rules for model m1? → 2 (# abs. guards)

Discharging POs of m1: Guard Strengthening in Refinement

ML_out/GRD

axioms

$d \in \mathbb{N}$

$d > 0$

abs. I

$n \in \mathbb{N}$

$n \leq d$

con. I.

$a \in \mathbb{N}$

$b \in \mathbb{N}$

$c \in \mathbb{N}$

$a + b + c = n$

$a = 0 \lor c = 0$

con. gu. of ML_out

$a + b < d$

$c = 0$

$\vdash$

abs. gu. of ML_out

$n < d$

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

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

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

MON

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

EQ_LR

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

MON

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

ARI

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

EQ_LR, MON

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

HYP

Discharging POs of m1: Guard Strengthening in Refinement

ML_in/GRD

$d \in \mathbb{N}$

$d > 0$

$n \in \mathbb{N}$

$n \le d$

$a \in \mathbb{N}$

$b \in \mathbb{N}$

$c \in \mathbb{N}$

$a + b + c = n$

$a = 0 \lor c = 0$

con. guard of ML_in $\boxed{c > 0}$

$\vdash$

abs. guard of ML_in $\boxed{n > 0}$

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

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

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

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

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

Lecture 16 - March 16

Reactive System: Bridge Controller

Announcements

- ProgTest1 result to be released by the end of Friday
- Lab3 released
- Example Questions for Written Test 2 released

  review session.
- To be completed by the final exam:

  Makeup lectures for WT1, WT2, ProgTest1, ProgTest2

Lecture

Reactive System: Bridge Controller

First Refinement: Invariant Preservation Concrete, Refined Events

"old" events, existing in both m0 and m1.

PO/VC Rule of Invariant Preservation: Sequents

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

a $\quad c-1$

$|||$

$a = 0 \vee$

$c - 1 = 0$

Abstract m0

variables: n

invariants: I

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

inv0_2 : $n \leq d$

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

$n' = n + 1$

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

$A(c)$ → axioms

$I(c, v)$ → abstract inv.

$J(c, v, w)$ → concrete inv.

$H(c, w)$ concrete guard.

$\vdash$

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

1..5

effect of e in the abs. state

effect of e in con. state

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

$a+1 \quad b \quad c \quad n+1$

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

Concrete m1

variables: a, b, c

invariants: J

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
```

H

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

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

H

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

ML_out, ML_in → 5 inv. in m1

10 (2 * 5)

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

ML_out/inv1_4/INV

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

$\vdash$ *

ML_in/inv1_5/INV ✓

$d \in \mathbb{N}$	axm0_1
$d > 0$	axm0_2
$n \in \mathbb{N}$	inv0_1
$n \leq d$	inv0_2
$a \in \mathbb{N}$	
$b \in \mathbb{N}$	
$c \in \mathbb{N}$	
$a + b + c = n$	
$a = 0 \vee c = 0$	
$c > 0$	grd of ML_in

$\vdash$ ** ✓

Commuting diagram.

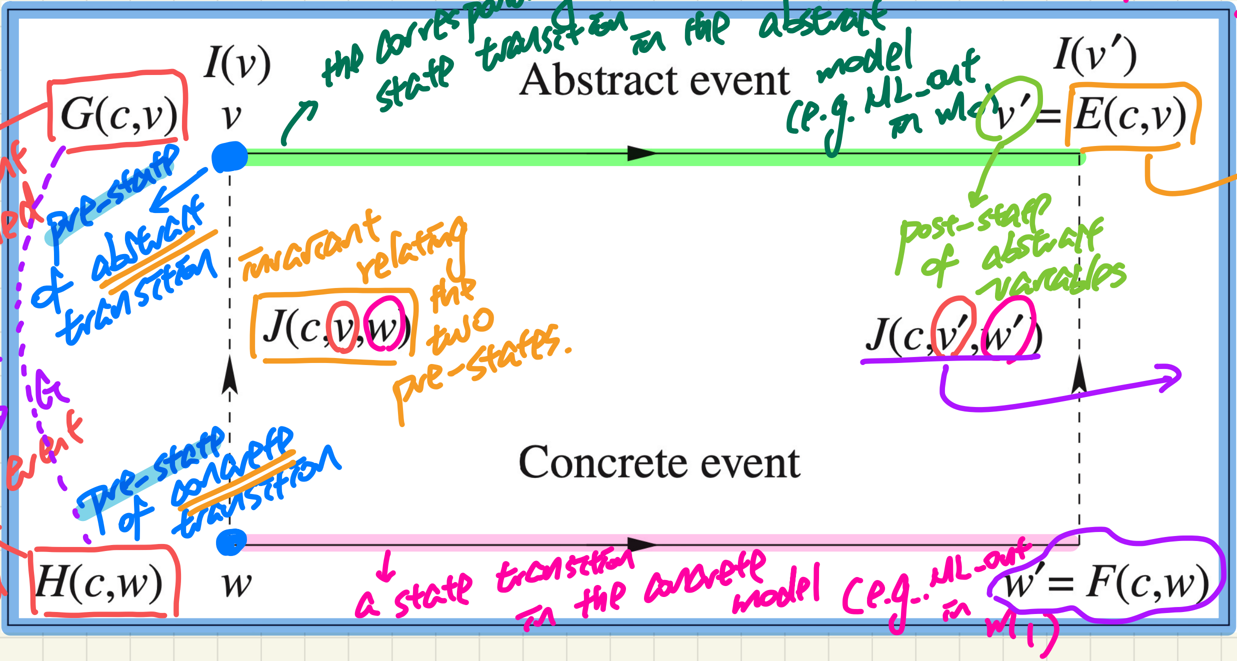
Visualizing Invariant Preservation in Refinement

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

effect of concrete transition

$w' = F(c, w)$

post-state of concrete variables



$I(v)$
v
the corresponding state transition in the abstract model (e.g. ML_out in m0)
Abstract event
$I(v')$
$v' = E(c,v)$
$G(c,v)$
true means the abs. event is enabled
guard streng.
$H \Rightarrow G$
pre-state of abstract transition
invariant relating the two pre-states.
$J(c,v,w)$
post-state of abstract variables
$J(c,v',w')$
Concrete event
pre-state of concrete transition
true means the con. event is enabled
$H(c,w)$
w
a state transition in the concrete model (e.g. ML_out in m1)
$w' = F(c,w)$

effect of abs. transition.

same linking inv. holds at the post-state

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

$d \in \mathbb{N}$

$d > 0$

$n \in \mathbb{N}$

$n \leq d$

$a \in \mathbb{N}$

$b \in \mathbb{N}$

$c \in \mathbb{N}$

$a + b + c = n$

$a = 0 \lor c = 0$

$c > 0$

$\vdash$

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

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

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

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

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

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

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

Lecture 17 - March 21

Reactive System: Bridge Controller

Announcements

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

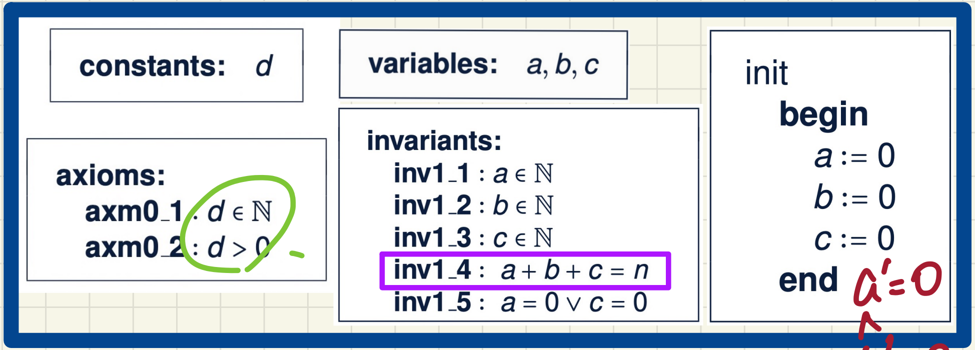
Zoom

Lecture

Reactive System: Bridge Controller

First Refinement: Inv. Establishment

PO of Invariant Establishment in Refinement



constants: d

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

$a' = 0$

$b' = 0$ $c' = 0$

Components

K(c): effect of abstract init

L(c): effect of concrete init

Rule of Invariant Establishment

$$A(c)$$

~~$J(c, v, w)$~~ ✗ ∵ no pre-state ✓ when init. the system

$$\vdash$$

$$J_i(c, K(c), L(c))$$

abs. init. con. init. steps.

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

↓ 5 (init, 5 inv.)

Exercise:

Generate Sequents from the INV rule.

init/inv1_4/INV

$d \in \mathbb{N}$

$d > 0$

$\vdash$

$* \boxed{0 + 0 + 0 = 0}$

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

$0 + 0 + 0 = 0$

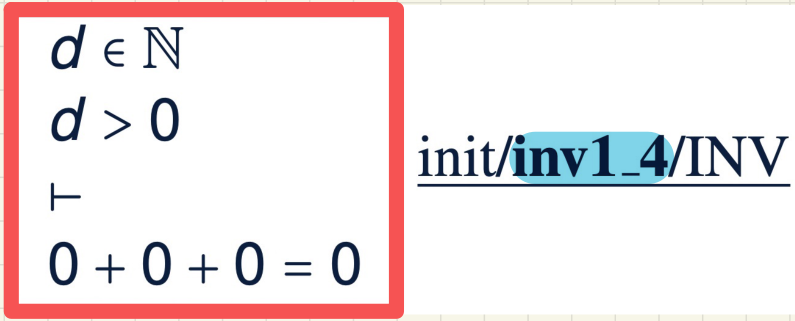
exercise:

init/inv1_5/INV :

formulate + prove

ARI : simplification

Discharging PO of Invariant Establishment in Refinement



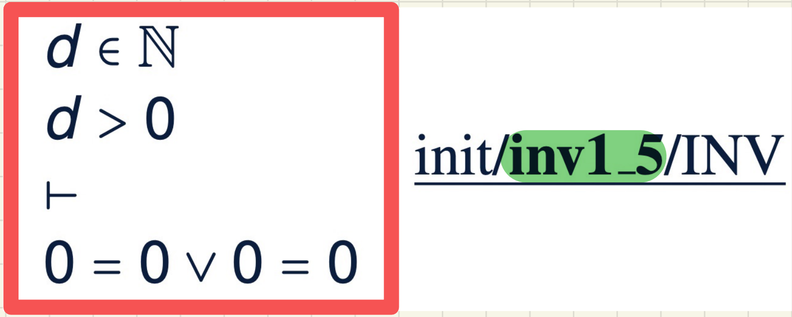
$d \in \mathbb{N}$

$d > 0$

$\vdash$

$0 + 0 + 0 = 0$

init/**inv1_4**/INV



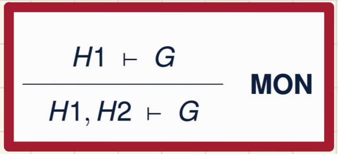
$d \in \mathbb{N}$

$d > 0$

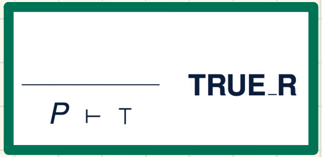
$\vdash$

$0 = 0 \lor 0 = 0$

init/**inv1_5**/INV

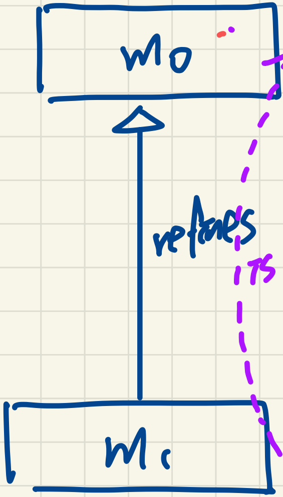


$$\frac{H1 \vdash G}{H1, H2 \vdash G} \quad \textbf{MON}$$



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

Events



abstract events

ML_out

ML_in

its simulated by

Concrete events

old event

ML_out

ML_in

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

new events

IL_in, IL_out

(1) guards, actions

(2) IL_in is simulated by which abstract evt. ?

(3) divergence

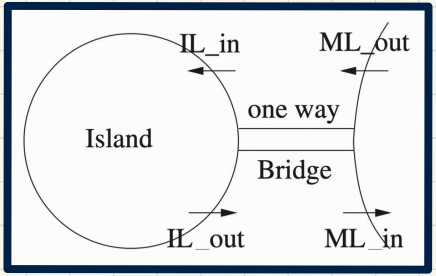
(4) variant of model

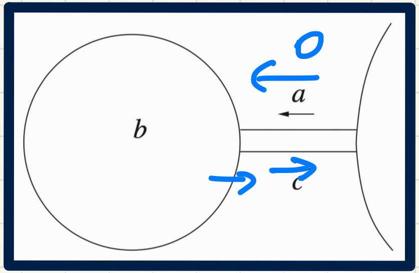
Lecture

Reactive System: Bridge Controller

First Refinement: Invariant Preservation
New Events

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





constants: d

axioms:

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

axm0_2 : $d > 0$

variables: a, b, c

invariants:

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

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

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

inv1_4 : $a + b + c = n$

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

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

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

$a > 0$

$\boxed{c = 0}$ not necessary ∵ inv1-5

Q. Guard: $\boxed{a + b < d}$

not necessary:

① ML_out already checks it

② $a' + b' = (a-1) + (b+1) = a + b$

$a := a - 1$

$b := b + 1$

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

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

$b > 0$

$a = 0$

$b := b - 1$

$c := c + 1$

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

Before-After Predicates of Event Actions: 1st Refinement

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

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

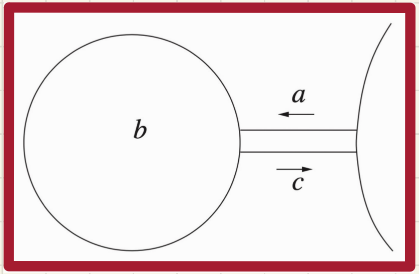
```
IL_out
  when
    $b > 0$
    $a = 0$
  then
    $b := b - 1$
    $c := c + 1$
  end
```

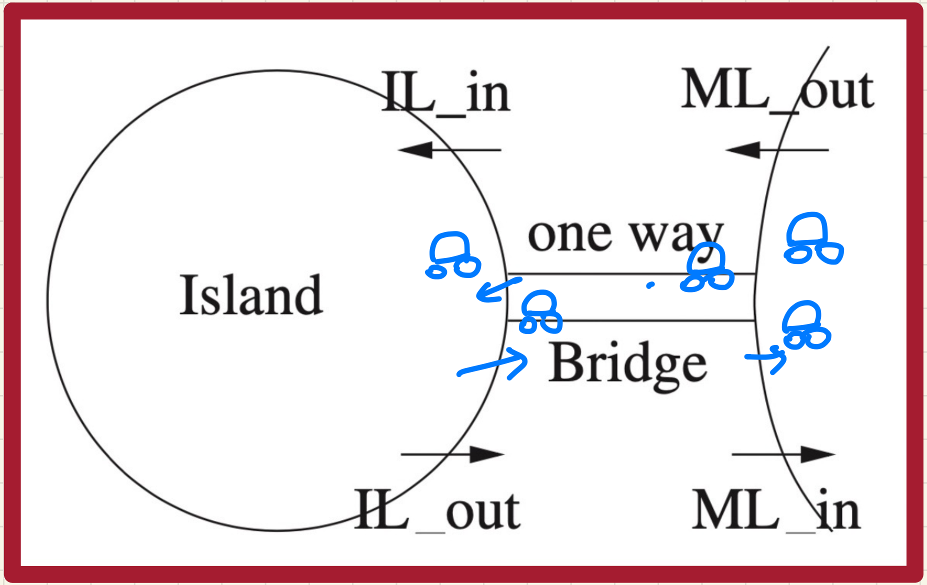
BAP: $\boxed{b' = \underline{\underline{b-1}} \wedge c' = \underline{\underline{c+1}} \wedge a' = \underline{\underline{a}}}$

$a' + b' + c' = a + b + c = \boxed{n}$

- Pre-State
- Post-State
- Sate Transition

Concrete State Space





Trace: 1 car travelling

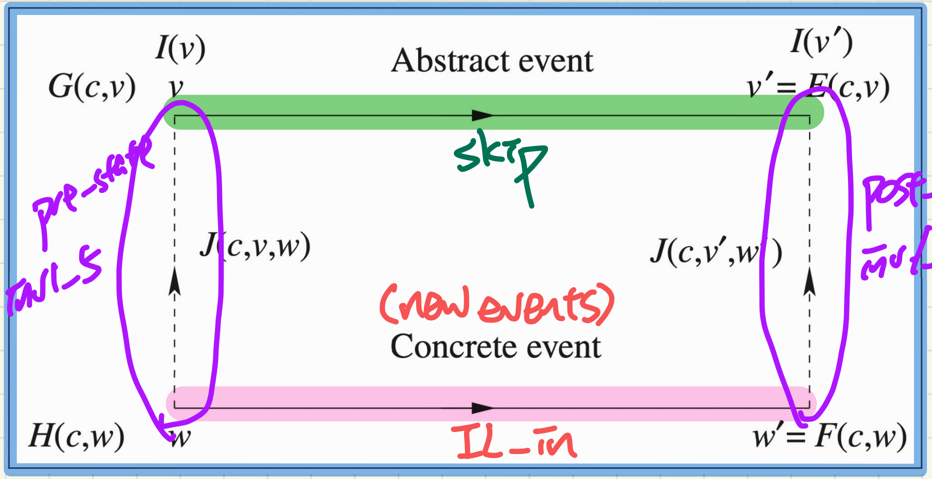
<init, ML_out, IL_in, IL_out, ML_in>

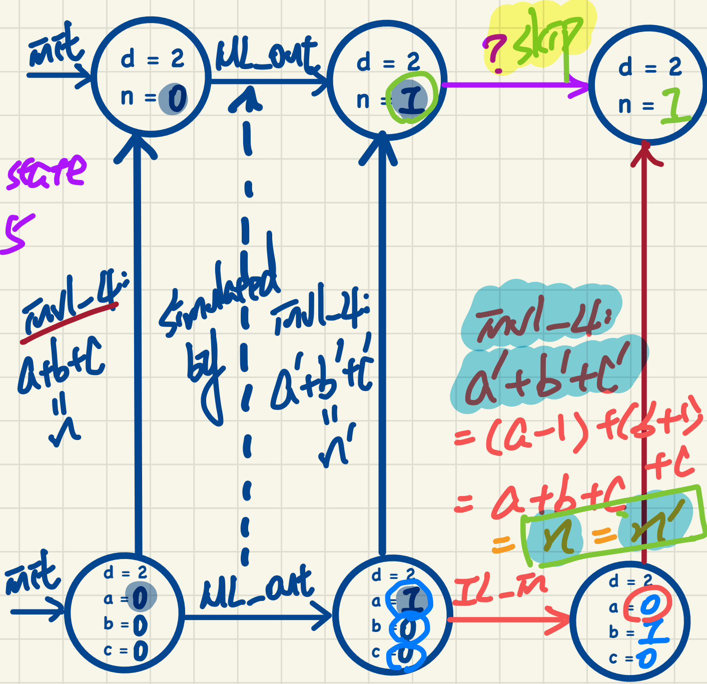
Exercise 2 cars travelling

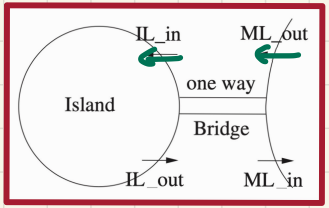
Visualizing Invariant Preservation in Refinement

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

skip
begin
(n′=n)
end







inv1_4: a+b+c = n

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 : $n \leq d$

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

Concrete m1

variables: a, b, c

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

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

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

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

IL_in/INV1_4/INV

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

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

? $a' + b' + c' = \overset{n}{n'}$

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

IL_in/INV1_5/INV

(Exercise: formulate & prove).

skip event

Discharging POs of m1: Invariant Preservation in Refinement

IL_in/inv1_4/INV

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

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

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

Discharging POs of m1: Invariant Preservation in Refinement

ML_in/inv1_5/INV

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

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

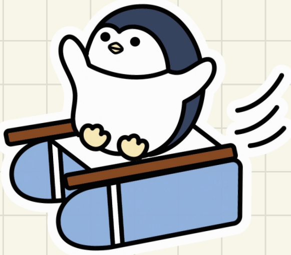
$$\frac{H \vdash Q}{H \vdash P \lor Q}\ \textbf{OR\_R2}$$

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

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

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

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



Lecture

Reactive System: Bridge Controller

First Refinement: Convergence
New Events

Livelock Caused by New Events Diverging

An alternative m1 (for demonstration)

while(true){
waiting
}

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 : $c \in \mathbb{Z}$

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

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

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

IL_in
  begin
    a := a − 1
    b := b + 1
  end

IL_out
  begin
    b := b − 1
    c := c + 1
  end
```

Abstract Transitions: <init, ML_out, skip, skip, skip, skip, --->

Concrete Transitions: <init, ML_out, IL_in, IL_out, IL_in, IL_out, --->

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

Exercise

POs related to INV preservation can be discharged. How?

Wednesday, March 22

Written Test 2 Review

Invariant Preservation

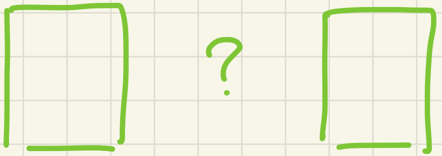
Concrete events

↓ old (ML_out, ML_in) 10 slide 59

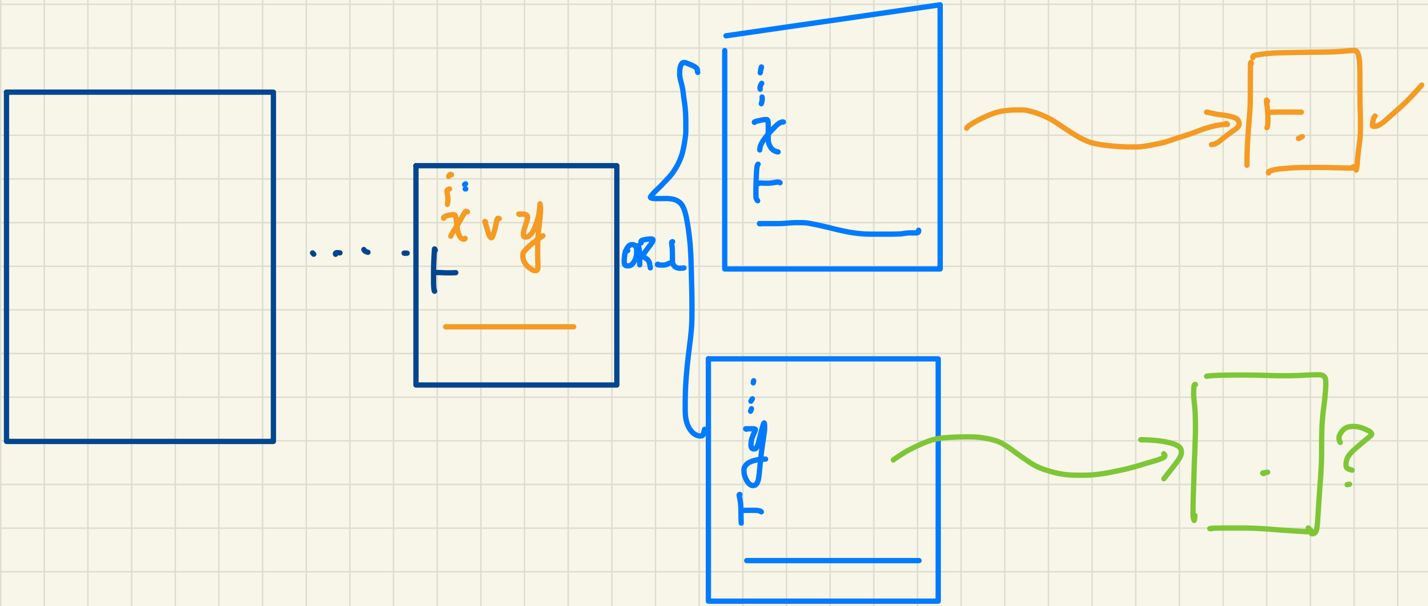
new (IL_in, IL_out) 10 slide 71

$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, \underline{\underline{P \vee Q}} \vdash R}$$

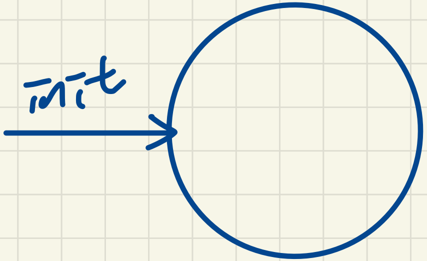
$\underline{\underline{\text{OR}}}\text{-}\underline{\underline{L}}$



Now, OR-L, - -



slide 58, EQ_LR^2 typo
(lower branch)



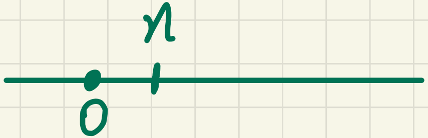
1. no notion of pre-state of init

2. init always enabled

$c \leq \underline{\underline{n}}$

✓ $c > 0$

ARI



<u>release ProgTest avg</u>.

Lecture 18 - March 28

Reactive System: Bridge Controller

Announcements

- Bonus Opportunity - **Course Evaluation**
- **ProgTest1**: Andy (eMail, Zoom); Jackie (Office Hour)
- **Lab3 Part 2** released
- **ProgTest2** → format identical to Labs
  - 60% Part 1 : complete context
  - Part 2 : complete manual proofs
- **Final Exam**: Review Q&A Sessions

Tue : 1pm Maria

Thur : 2:30pm Andy

Exam

↳ 3 hours] Sunday, April 16 2pm (tennis centre)

↳ papper (no Rodin, but you may be asked to read or write in Rodin syntax)

↳ a piece of data sheet allowed

↳ 1. one side

2. computer-typed (font ≥ 10pt)

Livelock Caused by New Events Diverging

An alternative m1 (for demonstration)

constants: d

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

variables: a, b, c

invariants:
inv1_1 : $a \in \mathbb{Z}$
inv1_2 : $b \in \mathbb{Z}$
inv1_3 : $c \in \mathbb{Z}$

IL

old events

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

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

new events.

```
IL_in
  begin
    a := a − 1
    b := b + 1
  end
```

```
IL_out
  begin
    b := b − 1
    c := c + 1
  end
```

guardless
↳ always enabled

as if:
while(1){
;
}

Abstract Transitions: < init, skip, skip, skip, skip, ... >

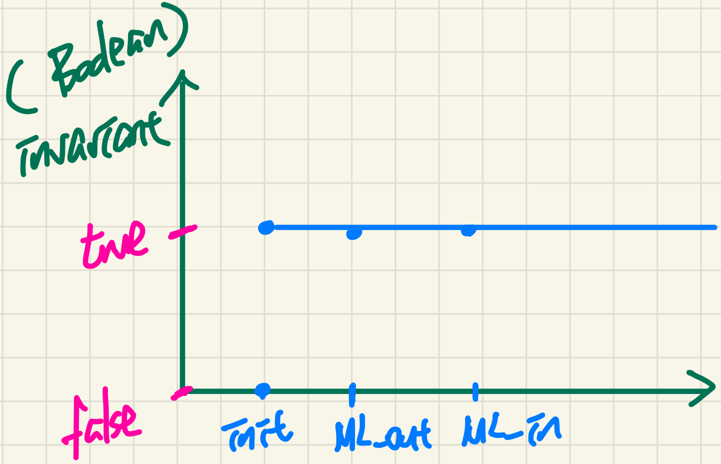
2. none of the old events is allowed to occur

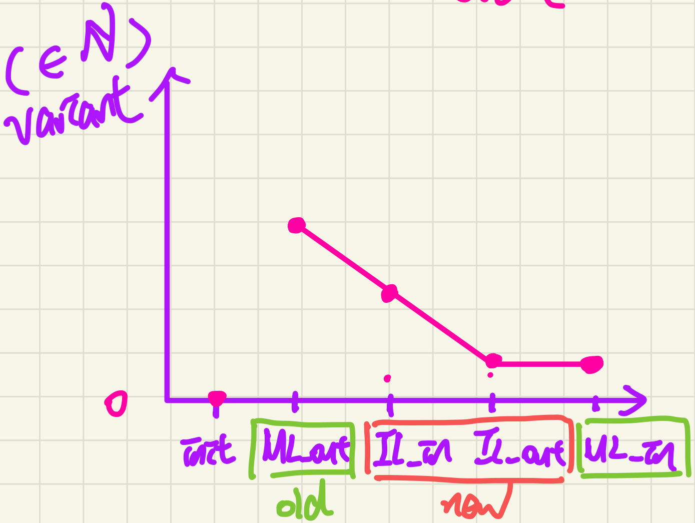
Concrete Transitions: < init, IL_in, IL_out, IL_in, IL_out, ... >

divergence : livelock
↳ a set of events keep interleaving. ↳ 1. new events interleave indefinitely

invariant : Boolean exp. that should always hold true after each event occurrence.

variant : Integer exp. that may change after event occurrence.





Q. Is an infinite interleaving of old events bad?

concrete < init, ML_out, ML_out, ... >

abstract < init, ML_out, ML_out, ... >

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 : $a = 0 \lor c = 0$

old

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

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

new

```
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
```

IL ═ ML

Exercise: VAR: a+b.

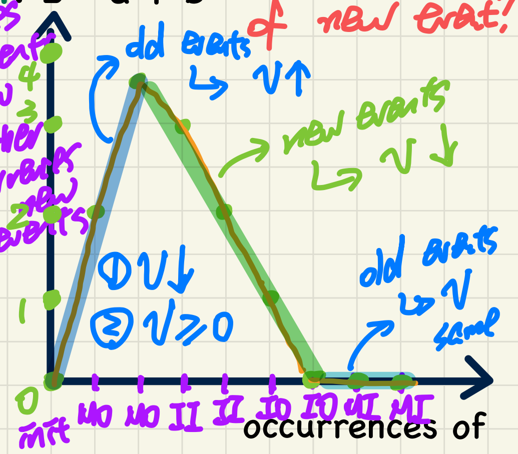
Variants for New Events: $2 \cdot a + b$

Is it still possible to have an occurrence of new event?

<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

occurrences of old events allow further occurrences of new events



PO of Convergence/Non-Divergence/Livelock Freedom

Variant Stays Non-Negative

$A(c)$ actions

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

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

$H(c, w)$ con. grd.

$\vdash$

$V(c, w) \in \mathbb{N}$

NAT

IL_in/NAT

$d \in \mathbb{N}$

$d > 0$

$n \in \mathbb{N}$

$n \leq d$

$a \in \mathbb{N}$ $\quad c \in \mathbb{N}$ $\quad a = 0 \lor c = 0$

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

A New Event Occurrence Decreases Variant

$A(c)$

$I(c, v)$

$J(c, v, w)$

$H(c, w)$

$\vdash$ ✓

$V(c, F(c, w)) < V(c, w)$

post-state ✓ pre-state ✓

VAR

IL_in/VAR

$d \in \mathbb{N}$

$d > 0$

$n \in \mathbb{N}$

$n \leq d$

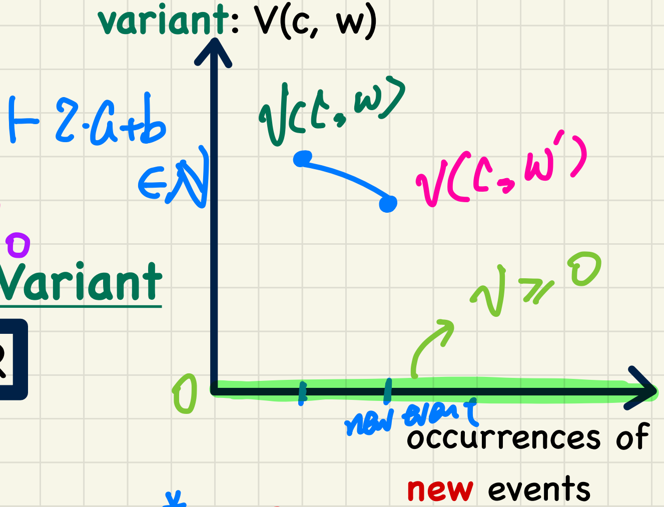
$a \in \mathbb{N}$ $\quad c \in \mathbb{N}$ $\quad a = 0 \lor c = 0$

$b \in \mathbb{N}$ $\quad a + b + c - n$ $\quad a > 0$

Variants for New Events: 2 · a + b

How many NAT POs to generate?

# concrete (old + new) events



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

$$2\cdot\overset{a-1}{\cancel{a}}+\overset{b+1}{\cancel{b}} < 2\cdot a+b$$

Example Inference Rules

$$\frac{S_2}{S_1}$$ To prove S1, sufficient to prove S2.

Justify:

$$H \Rightarrow P \vee Q \overset{\Leftrightarrow}{\equiv} H \wedge \neg P \Rightarrow Q$$

$$\frac{H, \neg P \vdash Q}{H \vdash P \vee Q}$$ OR_R

$$\frac{H \vdash P}{H \vdash P \vee Q}$$ OR_R1

$$\boxed{H \vdash P \vee Q}$$ ARI $$\boxed{H \vdash Q \vee P}$$ OR_R $$\boxed{H, \neg Q \vdash P}$$

$$\frac{H, P, Q \vdash R}{H, P \wedge Q \vdash R}$$ AND_L

$$\frac{H \vdash P \qquad H \vdash Q}{H \vdash P \wedge Q}$$ AND_R

Compare
↳ OR_L
↳ given a disjunctive hypothesis, split ...

Lecture 19 - March 30

Reactive System: Bridge Controller

Announcements

2:30 - 3:30

1:30pm - 3:30pm

- ProgTest1: Andy (eMail, Zoom); Jackie (Office Hour)
- Lab3 due soon
- ProgTest2

guide

Lecture

Reactive System: Bridge Controller

First Refinement:
Relative Deadlock Freedom

Example Inference Rules

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

Justify:

$$H \Rightarrow P \vee Q \overset{\Leftrightarrow}{\equiv} H \wedge \neg P \Rightarrow Q$$

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

$$\boxed{\begin{matrix} H \\ \vdash \\ P \vee Q \end{matrix}} \ \text{ARI} \ \boxed{\begin{matrix} H \\ \vdash \\ Q \vee P \end{matrix}} \ \text{OR\_R} \ \boxed{\begin{matrix} H \\ \neg Q \\ \vdash \\ P \end{matrix}}$$

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

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

Idea of Relative Deadlock Freedom

m_0 : DLF
m_1 : relative DLF

m_0
|
m_1
|
m_2

$A(c)$ axioms
$I(c, v)$ abs. I.
$J(c, v, w)$ con. I.

$G_1(c, v) \vee \cdots \vee G_m(c, v)$

disjunction of guards of all <u>abstract</u> events

circumstance for abs. model not to deadlock

$\vdash \Rightarrow$

$H_1(c, w) \vee \cdots \vee H_n(c, w)$

disjunction of guards of all con. events

DLF

relative DLF

Guard Strengthening

$$\begin{array}{l} \vdots \\ H_i \\ \vdash \\ G_i \end{array}$$

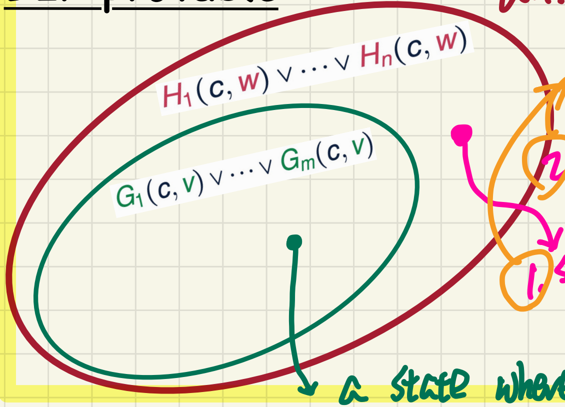
if concrete evt is enabled, then the abstract counterpart is enabled.

PRINCIPLES

1. DL is bad!
2. a refinement should <u>not</u> introduce a bad scenario

(unacceptable if there's a state where the con. model DL but the abs. model does not)

DLF provable



$H_1(c, w) \vee \cdots \vee H_n(c, w)$
$G_1(c, v) \vee \cdots \vee G_m(c, v)$

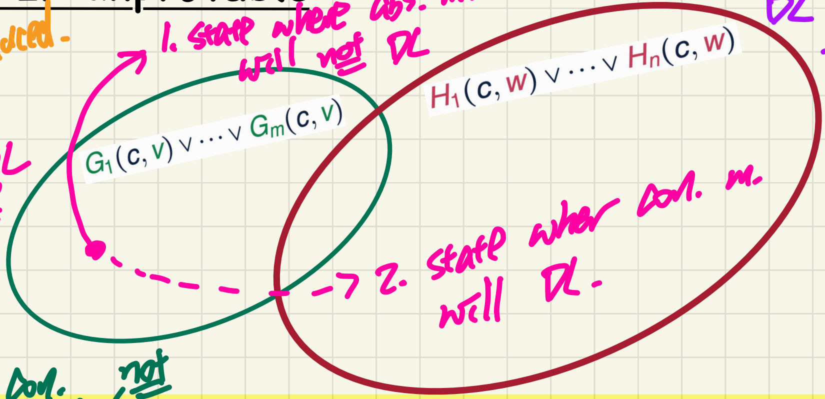
a state where abs. & con. models will not DL.

nothing bad introduced.

2. state when con. model not DL

1. state where abs. model DL.

DLF unprovable



$G_1(c, v) \vee \cdots \vee G_m(c, v)$
$H_1(c, w) \vee \cdots \vee H_n(c, w)$

1. state where abs. m. will not DL

2. state when con. m. will DL.

PO of Relative Deadlock Freedom

Abstract m0

variables: n

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

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

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

Concrete m1

variables: a, b, c

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

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

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

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

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

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

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

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

① $(a + b < d \land c = 0)$
$\lor$
② $(c > 0)$
$\lor$
③ $(a > 0)$
$\lor$
④ $(b > 0 \land a = 0)$

Discharging POs of m1: Relative Deadlock Freedom

Part 1

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

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

$$\frac{H, \neg P \;\vdash\; Q}{H \;\vdash\; P \vee Q}\quad \textbf{OR\_R}$$

$$
\begin{array}{ll}
\multicolumn{2}{l}{d \in \mathbb{N}} \\
\multicolumn{2}{l}{d > 0} \\
\multicolumn{2}{l}{n \in \mathbb{N}} \\
\multicolumn{2}{l}{n \leq d} \\
\multicolumn{2}{l}{a \in \mathbb{N}} \\
\multicolumn{2}{l}{b \in \mathbb{N}} \\
\multicolumn{2}{l}{c \in \mathbb{N}} \\
\multicolumn{2}{l}{a + b + c = n} \\
\multicolumn{2}{l}{a = 0 \vee c = 0} \\
\multicolumn{2}{l}{n < d \vee n > 0} \\
\multicolumn{2}{l}{\vdash} \\
 & a + b < d \wedge c = 0 \\
\vee & c > 0 \\
\vee & a > 0 \\
\vee & b > 0 \wedge a = 0
\end{array}
$$

$$
\begin{array}{ll}
\multicolumn{2}{l}{d > 0} \\
\multicolumn{2}{l}{b = 0 \vee b > 0} \\
\multicolumn{2}{l}{\vdash} \\
 & b < d \wedge 0 = 0 \\
\vee & b > 0 \wedge 0 = 0
\end{array}
$$

Discharging POs of m1: Relative Deadlock Freedom

Part 2

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

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

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

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

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

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

$$\begin{array}{ll}
\multicolumn{2}{l}{d > 0} \\
\multicolumn{2}{l}{b = 0 \lor b > 0} \\
\multicolumn{2}{l}{\vdash} \\
 & b < d \land 0 = 0 \\
\lor & b > 0 \land 0 = 0
\end{array}$$

Initial Model and 1st Refinement: Provably Correct

Abstract m0

constants: d

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

variables: n

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

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

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

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

Concrete m1

constants: d

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

variables: a, b, c

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

variants:
$2 \cdot a + b$

-NAT
-VAR

measure interleaving of new events

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

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

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

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

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

Correctness Criteria:

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

Lecture

Reactive System: Bridge Controller

2nd Refinement: State and Events

Bridge Controller: Abstraction in the 2nd Refinement

without this assumptions m2 would have to be much more complicated (e.g. red-light camera)

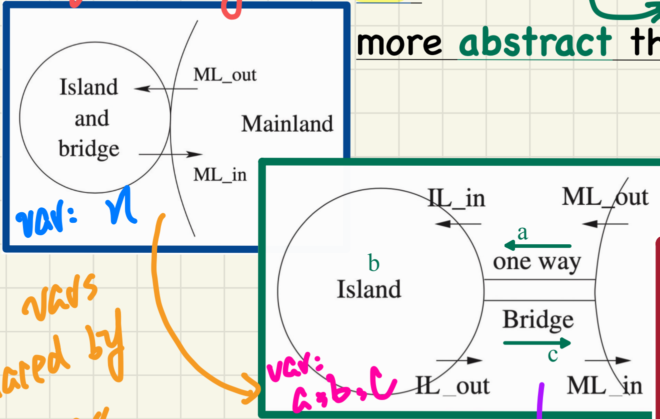
ENV1	The system is equipped with two traffic lights with two colors: green and red.
ENV2	The traffic lights control the entrance to the bridge at both ends of it.
ENV3	Cars are not supposed to pass on a red traffic light, only on a green one.

E-descriptions

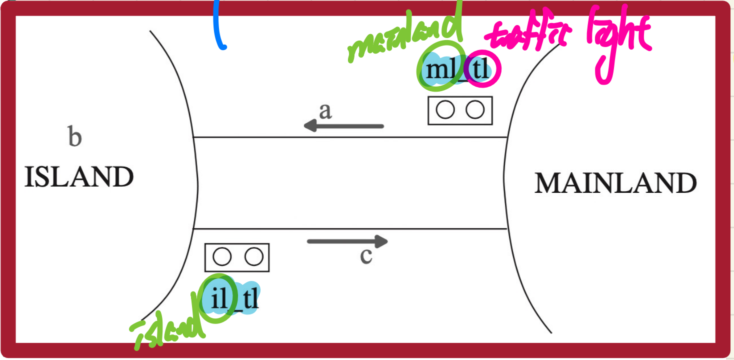
m0: more abstract than m1

m1: more concrete than m0, more abstract than m2

1. inv. est. & preservation
2. guard strengthening
3. relative DLF
4. convergence



m2: more concrete than m1



abs. vars replaced by con. vars

superposition
1. abs. vars inherited
2. new con. variables

$\{2, 2\} = \{2\}$

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:
a, b, c
ml_tl
il_tl

invariants:

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

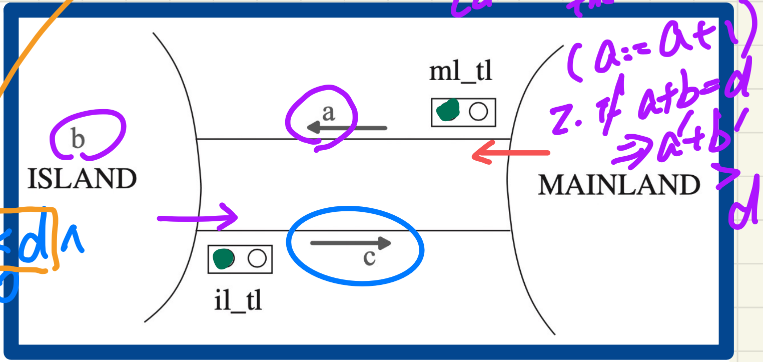
typing invariant

m2

why not: $a+b \leq d$?

$\because$ 1. $ml\_tl = g$ means a car can move to the bridge $(a := a+1)$

2. If $a+b = d$ $\Rightarrow a'+b' > d$



Static Part of Model

sets: *COLOR*

constants: *red, green*

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

$g \in COLOR \land r \in COLOR$

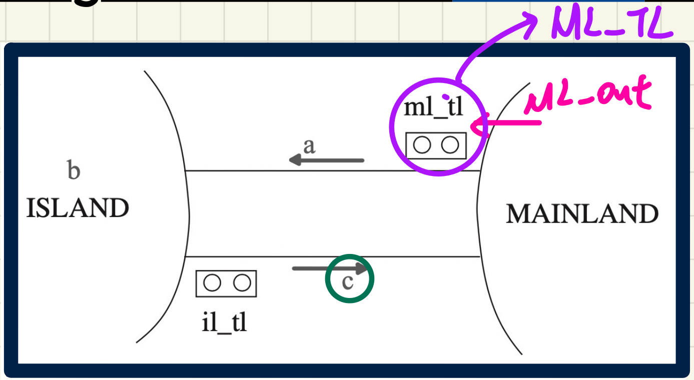
Exercises

inv2_3: being allowed to exit ML means limited cars & no crash

inv2_4: being allowed to exit IL means some car in IL & no crash

$\Rightarrow$

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



sets: *COLOR*

constants: *red*, *green*

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

variables:
a, b, c
ml_tl
il_tl

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

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

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

old event

ml_tl = green

for driver to follow

abstract guard from ml:

$c = 0 \land a + b < d$

guard for new event ML_TL

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

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

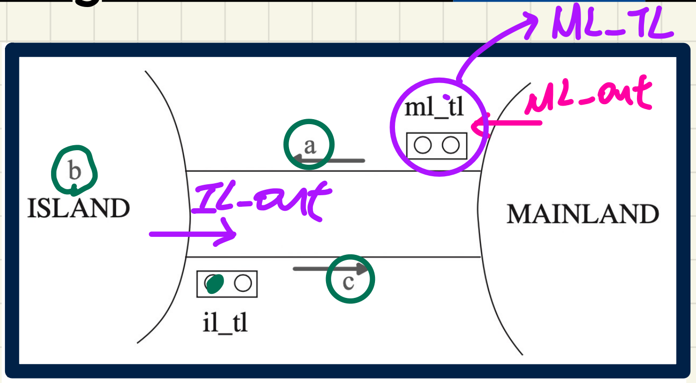
Lecture 20 - April 4

Reactive System: Bridge Controller

Announcements

- ProgTest1: Andy (eMail, Zoom); Jackie (Office Hour)
- Lab4 released
- ProgTest2
- Exam guide to be released
- Final makeup lecture to be released

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



sets: *COLOR*

constants: *red*, *green*

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

variables:
a, b, c
ml_tl
il_tl

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

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

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

old event

ml_tl = green

for driver to follow

→ abstract guard from ml:

$c = 0 \land a+b<d$

guard for new event ML_TL

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

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

IL_tl = green

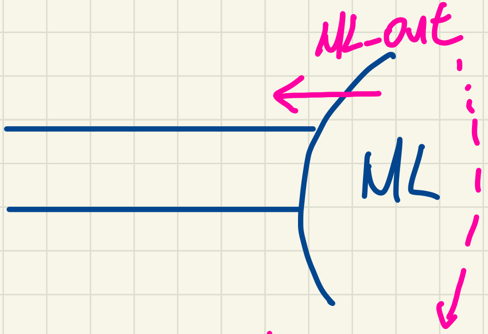
→ abstract guard from ml:

$a = 0 \land b > 0$

m1

(1st refinement)

no notion of traffic light

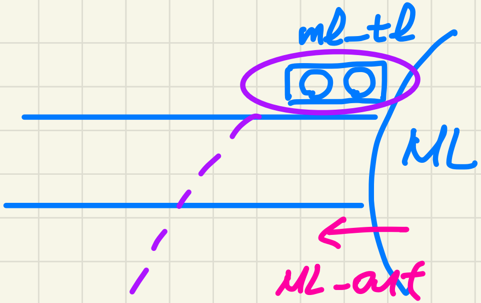


guard:

$c = 0$
$\wedge\ a+b<d$

m2

(2nd refinement)



New Event:

ML_tl_green

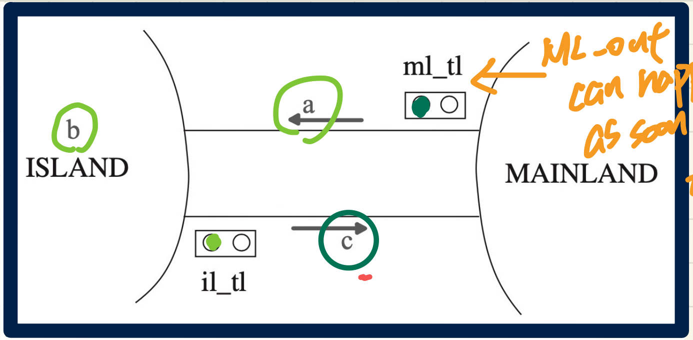
⇒ guard

driver following TL.

guard:

$ml\_tl = green$

Bridge Controller: Guards of “new” Events 2nd Refinement



ML-out can happen as soon as ml-tl turned green.

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
```

$ml\_tl = red$

$c = 0$
$a+b<d$

abstract guards of ML-out in m_1

IL_tl_green:

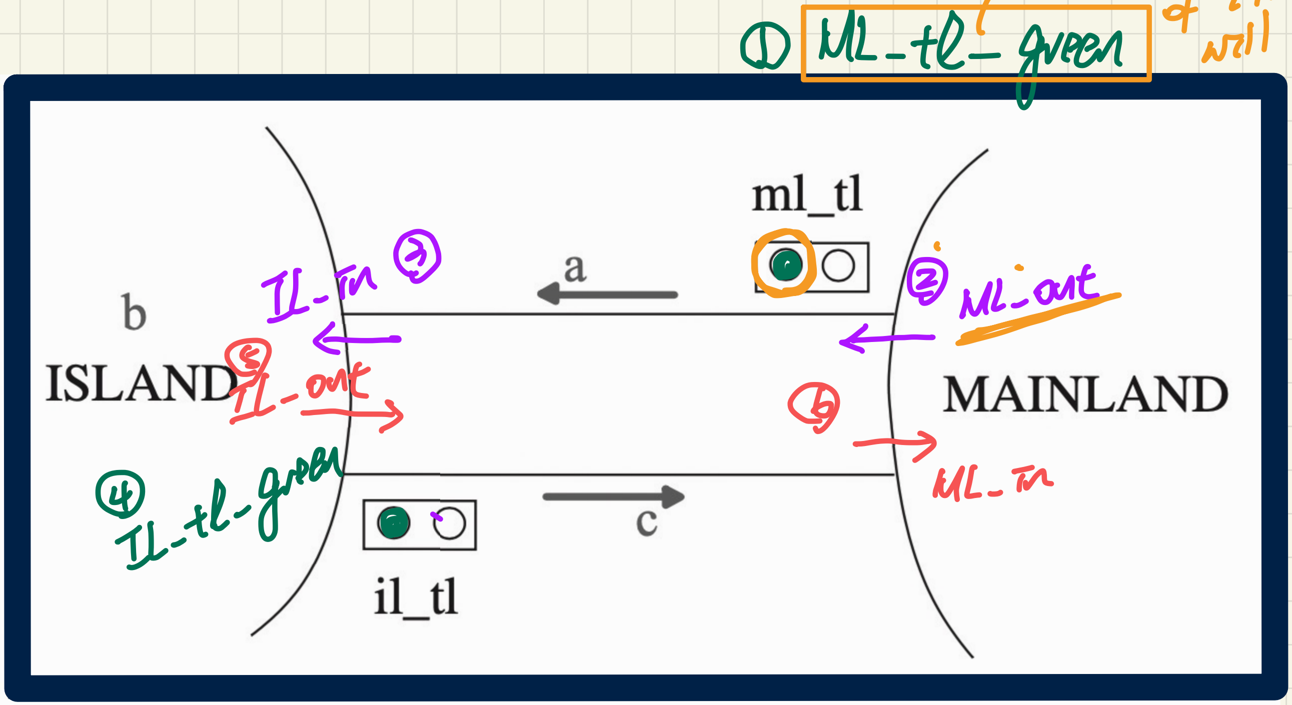
turn the traffic light **il_tl** to green

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

$il\_tl = red$

$b > 0$
$a = 0$

abstract guard of IL-out in m_1



occurrence of this event will make ML_out enabled.

How can the order of events be enforced?

A. By the design of event guards.

Lecture

Reactive System: Bridge Controller

2nd Refinement: Invariant Preservation

PO/VC Rule of Invariant Preservation: Sequents

Abstract m1

variables: a, b, c

invariants:

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

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

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

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

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

Concrete m2

variables:
a, b, c
ml_tl
il_tl

invariants:

- **inv2_1** : $ml\_tl \in COLOUR$
- **inv2_2** : $il\_tl \in COLOUR$
- **inv2_3** : $ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
- $*$ $\boxed{\textbf{inv2\_4} : il\_tl = green \Rightarrow b > 0 \wedge a = 0}$

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

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

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

ML_out/inv2_4/INV

axm0_1	{ $d \in \mathbb{N}$
axm0_2	{ $d > 0$
axm2_1	{ $COLOUR = \{green, red\}$
axm2_2	{ $green \neq red$
inv0_1	{ $n \in \mathbb{N}$
inv0_2	{ $n \leq d$
inv1_1	{ $a \in \mathbb{N}$
inv1_2	{ $b \in \mathbb{N}$
inv1_3	{ $c \in \mathbb{N}$
inv1_4	{ $a + b + c = n$
inv1_5	{ $a = 0 \vee c = 0$
inv2_1	{ $ml\_tl \in COLOUR$
inv2_2	{ $il\_tl \in COLOUR$
inv2_3	{ $ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
inv2_4	{ $il\_tl = green \Rightarrow b > 0 \wedge a = 0$
Concrete guards of *ML_out*	$ml\_tl = green$
	$\vdash$
$*$ *Concrete* invariant **inv2_4** with *ML_out*'s effect in the post-state	{ $il\_tl = green \Rightarrow b > 0 \wedge (a + 1) = 0$

abs. inv. from m1

con. inv. from m2

Exercise: Specify IL_out/inv2_3/INV

Example Inference Rules

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

Modus Ponens

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

$$\frac{H, P \overset{\Rightarrow}{\vdash} Q}{H \overset{}{\underset{\Rightarrow}{\vdash}} \boxed{P \Rightarrow Q}} \quad \text{IMP\_R}$$

$H \wedge P \Rightarrow Q$

$H \Rightarrow (p \Rightarrow q)$

Shunting

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

$$\frac{\cancel{H}, \neg Q \vdash P}{\cancel{H}, \boxed{\neg P} \overset{\Rightarrow}{\vdash} Q} \quad \text{NOT\_L}$$

$\neg Q \Rightarrow P$

$\neg p \Rightarrow Q$

Contra-positive

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

Discharging POs of m2: Invariant Preservation

First Attempt

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

MON

ML_out/inv2_4/INV

Outstanding/Unprovable Seq

Green ≠ red
ml_tl = green
Tl_tl = green
⊢
1 = 0

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

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

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

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

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

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

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

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

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

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

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

MON

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

IMP_R

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

IMP_L

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

AND_L

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

AND_R

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

MON

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

ARI

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

EQ_LR, MON

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

ARI

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

??

Understanding the Failed Proof on INV

Exercise

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**

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

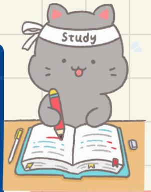
Contradiction

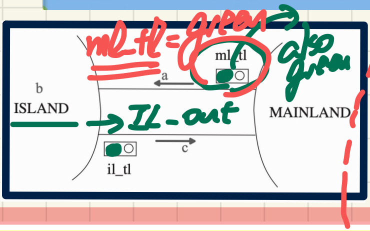
ML_out/inv2_4/INV

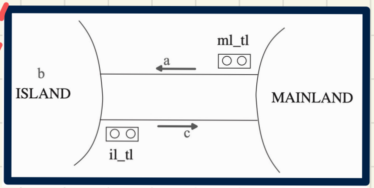
$d \in \mathbb{N}$
$d > 0$
$COLOUR = \{green, red\}$
$green \neq red$
$n \in \mathbb{N}$
$n \leq d$
$a \in \mathbb{N}$
$b \in \mathbb{N}$
$c \in \mathbb{N}$
$a + b + c = n$
$a = 0 \lor c = 0$
$ml\_tl \in COLOUR$
$il\_tl \in COLOUR$
$ml\_tl = green \Rightarrow a + b < d \land c = 0$
$il\_tl = green \Rightarrow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

Unprovable Sequent:

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







⟨ 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**	$il\_tl' = green$	$il\_tl' = green$

Lecture

Reactive System: Bridge Controller

2nd Refinement: Fixing the Model
Adding an Invariant

Fixing m2: Adding an Invariant

Abstract m1

variables: a, b, c

invariants:

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

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

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

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

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

Concrete m2

variables:
a, b, c
ml_tl
il_tl

invariants:

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

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

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

ML_out/inv2_4/INV

axm0_1	$d \in \mathbb{N}$
axm0_2	$d > 0$
axm2_1	$COLOUR = \{green, red\}$
axm2_2	$green \neq red$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
inv1_1	$a \in \mathbb{N}$
inv1_2	$b \in \mathbb{N}$
inv1_3	$c \in \mathbb{N}$
inv1_4	$a + b + c = n$
inv1_5	$a = 0 \lor c = 0$
inv2_1	$ml\_tl \in COLOUR$
inv2_2	$il\_tl \in COLOUR$
inv2_3	$ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4	$il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5	$ml\_tl = red \lor il\_tl = red$
Concrete guards of *ML_out*	$ml\_tl = green$
	$\vdash$
Concrete invariant **inv2_4** with *ML_out*'s effect in the post-state	$il\_tl = green \Rightarrow b > 0 \land (a + 1) = 0$

Exercise: Specify IL_out/inv2_3/INV

Discharging POs of m2: Invariant Preservation

Second Attempt

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

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

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

inv2_5

used to be unprovable before inv2_5 was added

ML_out/inv2_4/INV

green ≠ red
ml_tl = green
ml_tl = red ∨ il_tl = red
il_tl = green
⊢
1 = 0

OR_L

X

green ≠ red
ml_tl = green
ml_tl = red
il_tl = green
⊢ 1=0

EQ_LR, MON

Exercise

Approach 1: NOT_L

green ≠ red
green = red
il_tl = green
⊢
1=0

Approach 2: ARI

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

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

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

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

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

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

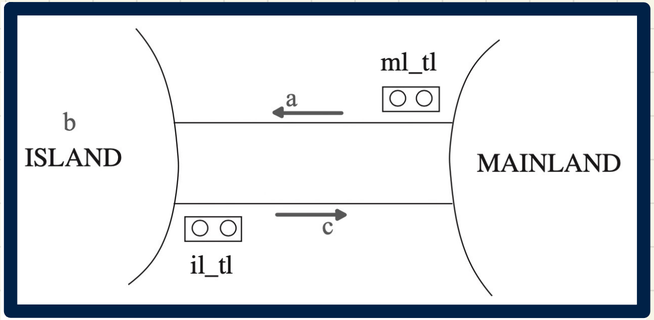
$$\frac{H, P \vdash R \qquad H, Q \vdash R}{H, P \lor Q \vdash R} \ \textbf{OR\_L}$$

Lecture

Reactive System: Bridge Controller

2nd Refinement: Fixing the Model
Adding Actions

Fixing m2: Adding Actions



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

$ml\_tl' = g$
$\wedge\ il\_tl' = r \wedge a' = a \wedge b' = b \wedge c' = c$

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

ML_tl_green/inv2_5/INV

axm0_1	$d \in \mathbb{N}$
axm0_2	$d > 0$
axm2_1	$COLOUR = \{green, red\}$
axm2_2	$green \neq red$
inv0_1	$n \in \mathbb{N}$
inv0_2	$n \leq d$
inv1_1	$a \in \mathbb{N}$
inv1_2	$b \in \mathbb{N}$
inv1_3	$c \in \mathbb{N}$
inv1_4	$a + b + c = n$
inv1_5	$a = 0 \vee c = 0$
inv2_1	$ml\_tl \in COLOUR$
inv2_2	$il\_tl \in COLOUR$
inv2_3	$ml\_tl = green \Rightarrow a + b < d \wedge c = 0$
inv2_4	$il\_tl = green \Rightarrow b > 0 \wedge a = 0$
inv2_5	$ml\_tl = red \vee il\_tl = red$

Concrete guards

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

$\vdash$

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

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

Exercise: Proof

Exercise: Specify IL_tl_green/inv2_5/INV

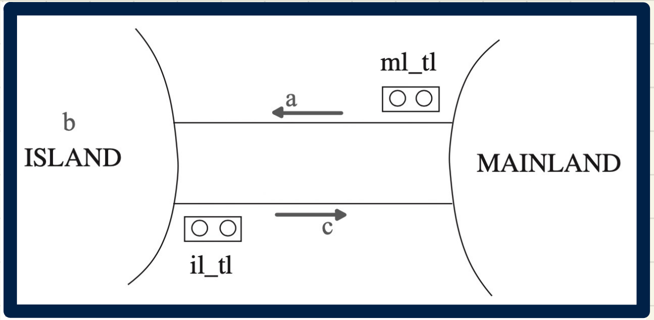
Lecture

Reactive System: Bridge Controller

2nd Refinement: Fixing the Model Splitting Events

Invariant Preservation: ML_out/inv2_3/INV

ML_out/inv2_4 discussed earlier



```
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
```

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/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	$\{\ ml\_tl = green \Rightarrow (a + 1) + b < d \land c = 0$

Exercise: Specify IL_out/inv2_4/INV

IL_out/inv2_3 discussed earlier

Discharging POs of m2: Invariant Preservation

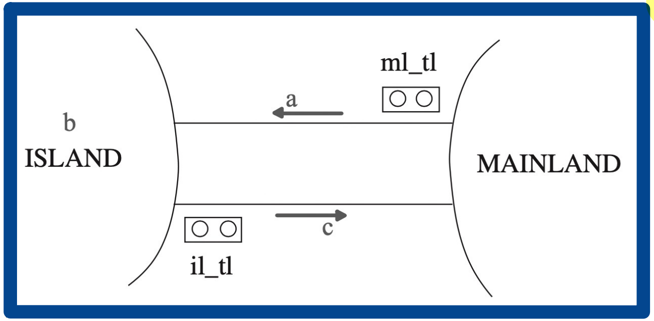
First Attempt

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

ML_out/inv2_3/INV → Exercise

IL_out/inv2_4/INV

↳ expected to see: a similar unprovable sequent



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

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

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

MON

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

IMP_R

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

IMP_R

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

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

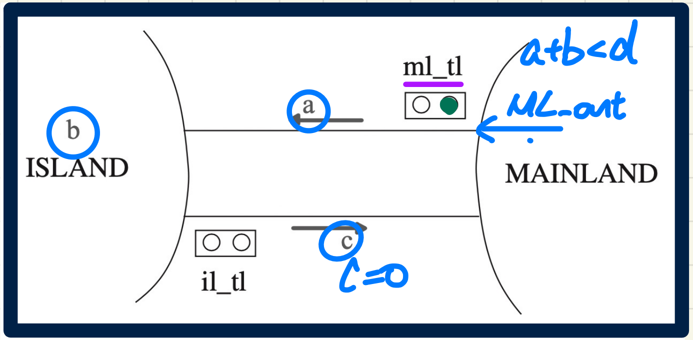
Understanding the Failed Proof on INV

variables:
a, b, c
ml_tl
il_tl

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

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

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



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

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

Unprovable Sequent from ML_out/inv2_3/INV

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

another ML_out allowed

ML_out

inv2_3 is preserved
∵ false ⇒ _ ✓

$d = 3$, $b = 0$, $a = 0$	$(a+1)+b \boxed{\neq d}$	[$\boxed{(a+1)+b<d}$ 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***]
$d = 3$, $b = 0$, $a = 2$	$(a+1)+b \boxed{= d}$	[$(a+1)+b<d$ 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***]

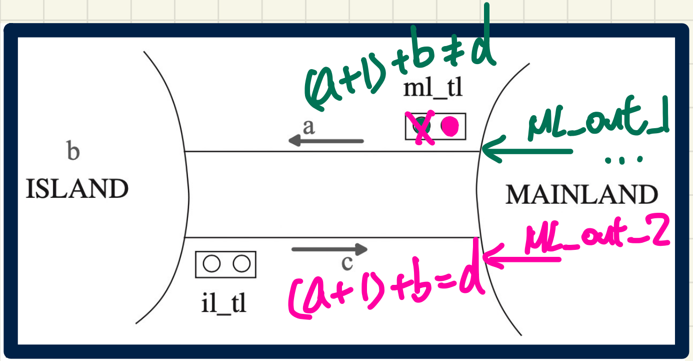
no more ML_out allowed ⇒ $\boxed{ml\_tl := red}$

Fixing m2: Splitting Events

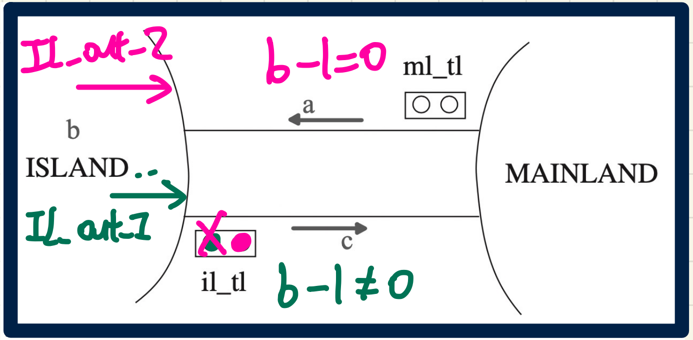
m1: ML_out IL_out

refines

m2: ML_out_1 ML_out_2 IL_out_1 IL_out_2

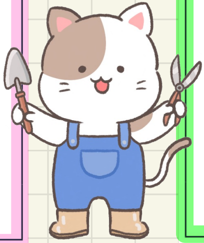


old, concrete events



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

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



```
IL_out_1
  when
    il_tl = green
    b ≠ 1          ≡ b-1≠0
  then
    b := b − 1
    c := c + 1
  end
```

```
IL_out_2
  when
    il_tl = green
    b = 1          ≡ b-1=0
  then
    b := b − 1
    c := c + 1
    il_tl := red
  end
```

6 ↑ 8 : ML_out split
IL_out split

# of sequents for INV:
8 × 5 = 40

Lecture

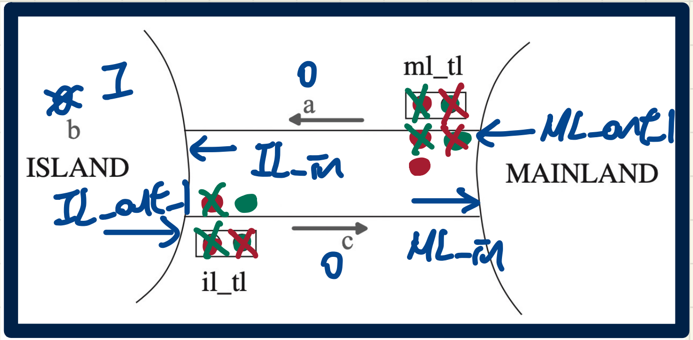
Reactive System: Bridge Controller

2nd Refinement: Livelock/Divergence

Current m2 May Livelock

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

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



d=2

Expected trace: no divergence from this trace

⟨init, ML_tl_green, ML_out_1, IL_in,
IL_tl_green, IL_out_1, ML_in⟩

a new event

old events

also a valid trace of m2, but leading to livelock

Is ML_tl_g. enabled?

Is IL_tl_g. enabled?

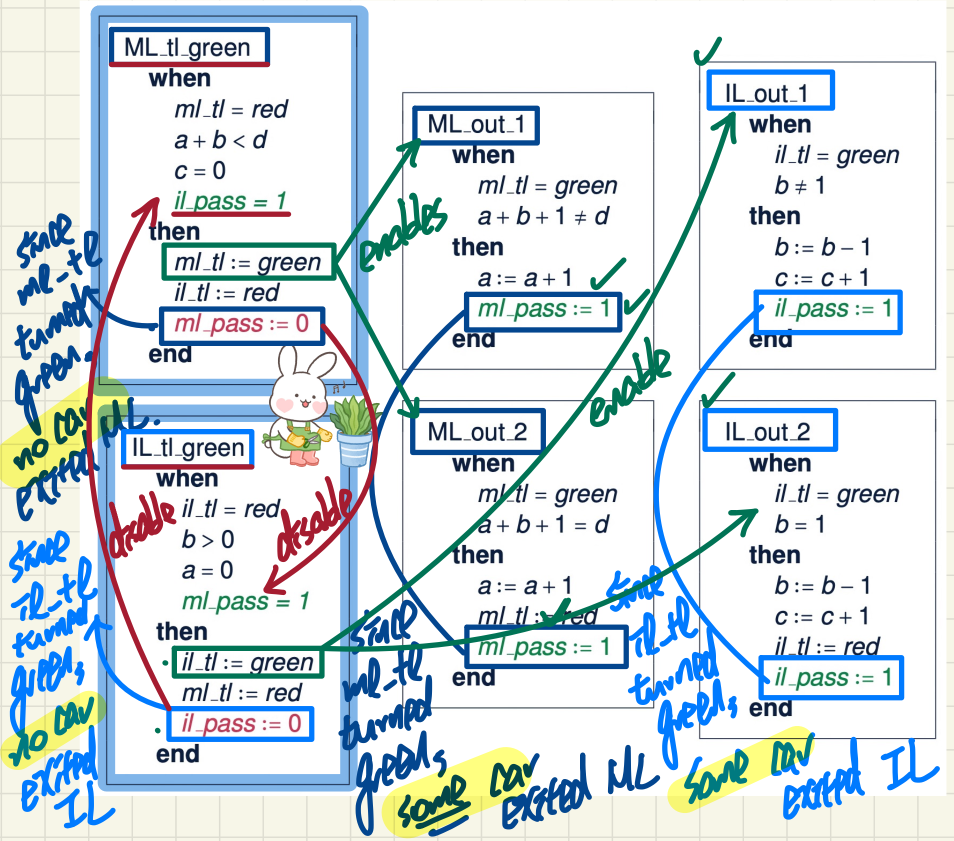
⟨ init ,	ML_tl_green ,	ML_out_1 ,	IL_in ,	IL_tl_green ,	ML_tl_green ,	IL_tl_green , . . . ⟩
$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$	$d = 2$
$a' = 0$	$a' = 0$	$a' = 1$	$a' = 0$	$a' = 0$	$a' = 0$	$a' = 0$
$b' = 0$	$b' = 0$	$b' = 0$	$b' = 1$	$b' = 1$	$b' = 1$	$b' = 1$
$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$	$c' = 0$
ml_tl = red	ml_tl′ = green	ml_tl′ = green	ml_tl′ = green	ml_tl′ = red	ml_tl′ = green	ml_tl′ = red
il_tl = red	il_tl′ = red	il_tl′ = red	il_tl′ = red	il_tl′ = green	il_tl′ = red	il_tl′ = green

pattern of divergence

Fixing m2: Regulating Traffic Light Changes

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

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



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

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

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

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

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

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

since ml_tl turned green,
no car exited ML.
since IL_tl turned green,
no car exited IL
disable
disable
enables
enable
since ml_tl turned green,
some car exited ML
since IL_tl turned green,
some car exited IL
✓
✓
✓
✓

: ml_tl, IL_tl both red

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

Fixing m2: Measuring Traffic Light Changes

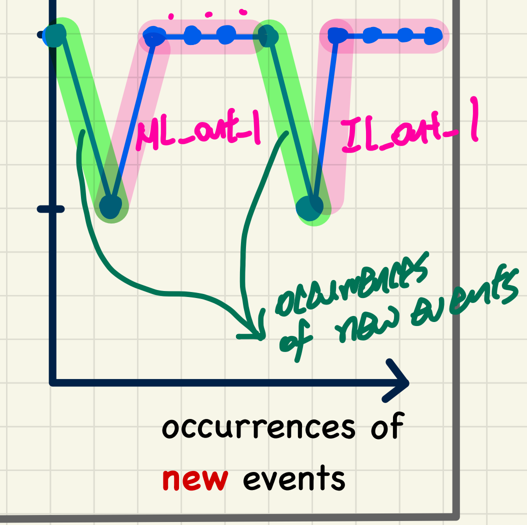
```
ML_tl_green
  when
    $ml\_tl = red$
    $a + b < d$
    $c = 0$
    $il\_pass = 1$
  then
    $ml\_tl := green$
    $il\_tl := red$
    $ml\_pass := 0$
  end
```

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

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

variants : *ml_pass* + *il_pass*

variant: V(c, w)



occurrences of **new** events

PO of Convergence/Non-Divergence/Livelock Freedom

A New Event Occurrence Decreases Variant

$A(c)$

$I(c, v)$

$J(c, v, w)$

$H(c, w)$

$\vdash$

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

post-state evaluation

pre-state evaluation

VAR

applicable to new events

Variants: ml_pass + il_pass

$* \ \overset{0}{\cancel{ml\_pass'}} + \overset{il\_pass}{\cancel{il\_pass'}} < ml\_pass + il\_pass$

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

BAP: $\cancel{ml\_pass'} = 0 \land \cancel{il\_pass'} = il\_pass \land \ldots$

ML_tl_green/VAR

$d \in \mathbb{N}$	$d > 0$	
$COLOUR = \{green, red\}$	$green \neq red$	
$n \in \mathbb{N}$	$n \leq d$	m0
$a \in \mathbb{N}$	$b \in \mathbb{N}$	$c \in \mathbb{N}$ m1
$a + b + c = n$	$a = 0 \lor c = 0$	m1
$ml\_tl \in COLOUR$	$il\_tl \in COLOUR$	m2
$ml\_tl = green \Rightarrow a + b < d \land c = 0$	$il\_tl = green \Rightarrow b > 0 \land a = 0$	m2
$ml\_tl = red \lor il\_tl = red$		m2
$ml\_pass \in \{0, 1\}$	$il\_pass \in \{0, 1\}$	m2
$ml\_tl = red \Rightarrow ml\_pass = 1$	$il\_tl = red \Rightarrow il\_pass = 1$	m2
$ml\_tl = red$	$a + b < d$	$c = 0$
$il\_pass = 1$		

Concrete guards of ML_tl_green

$\vdash$

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

Lecture

Reactive System: Bridge Controller

2nd Refinement:
Relative Deadlock Freedom

PO of Relative Deadlock Freedom

axm0_1 $d \in \mathbb{N}$
axm0_2 $d > 0$
axm2_1 $COLOUR = \{green, red\}$
axm2_2 $green \neq red$
inv0_1 $n \in \mathbb{N}$
inv0_2 $n \leq d$
inv1_1 $a \in \mathbb{N}$
inv1_2 $b \in \mathbb{N}$
inv1_3 $c \in \mathbb{N}$
inv1_4 $a + b + c = n$
inv1_5 $a = 0 \lor c = 0$
inv2_1 $ml\_tl \in COLOUR$
inv2_2 $il\_tl \in COLOUR$
inv2_3 $ml\_tl = green \Rightarrow a + b < d \land c = 0$
inv2_4 $il\_tl = green \Rightarrow b > 0 \land a = 0$
inv2_5 $ml\_tl = red \lor il\_tl = red$
inv2_6 $ml\_pass \in \{0, 1\}$
inv2_7 $il\_pass \in \{0, 1\}$
inv2_8 $ml\_tl = red \Rightarrow ml\_pass = 1$
inv2_9 $il\_tl = red \Rightarrow il\_pass = 1$

Disjunction of ***abstract*** guards

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

$\vdash$

Disjunction of ***concrete*** guards

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

Abstract m1

variables: a, b, c

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

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

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

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

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

Concrete m2

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

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

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

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

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

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

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

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

Discharging POs of m2: Relative Deadlock Freedom

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

Ex.1

⋮

Study

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

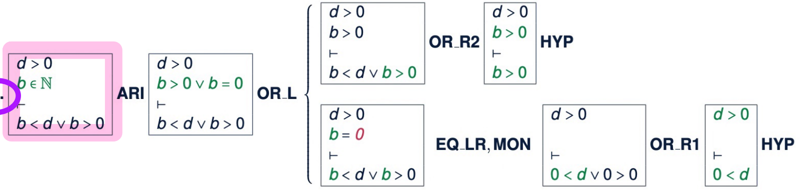
Ex.2

...

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

Ex3

...



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

ARI

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

OR_L

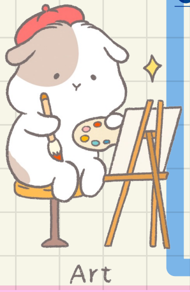
$d > 0$, $b > 0$ $\vdash$ $b < d \lor b > 0$ **OR_R2** $d > 0$, $b > 0$ $\vdash$ $b > 0$ **HYP**

$d > 0$, $b = 0$ $\vdash$ $b < d \lor b > 0$ **EQ_LR, MON** $d > 0$ $\vdash$ $0 < d \lor 0 > 0$ **OR_R1** $d > 0$ $\vdash$ $0 < d$ **HYP**

1st Refinement and 2nd Refinement: Provably Correct

```
variables:   a, b, c

constants:   d

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

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

variants:
    2 · a + b

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

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

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

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

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

Abstract m1



Art

Correctness Criteria:

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

Superposition

```
variables:
    a
    b
    c
    ml_tl
    il_tl
    ml_pass
    il_pass

constants:   d

sets:   COLOR

axioms:
    axm0_1 : d ∈ ℕ
    axm0_2 : d > 0
    axm2_1 : COLOR = {green, red}
    axm2_2 : green ≠ red

invariants:
    inv2_1 : ml_tl ∈ COLOUR
    inv2_2 : il_tl ∈ COLOUR
    inv2_3 : ml_tl = green ⇒ a + b < d ∧ c = 0
    inv2_4 : il_tl = green ⇒ b > 0 ∧ a = 0
    inv2_5 : ml_tl = red ∨ il_tl = red
    inv2_6 : ml_pass ∈ {0, 1}
    inv2_7 : il_pass ∈ {0, 1}
    inv2_8 : ml_tl = red ⇒ ml_pass = 1
    inv2_9 : il_tl = red ⇒ 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 ≠ 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

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

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

divergence freedom

Concrete m2

Lecture

Distributed System: File Transfer Protocol

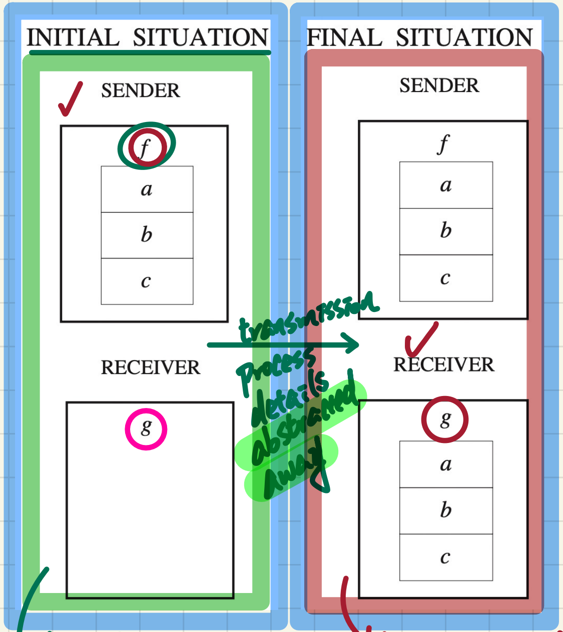
Initial Model: State and Events

FTP: Abstraction and State Space in the Initial Model

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

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

Synchronous Transmission



$b = FALSE \Rightarrow g = \emptyset$ $b = TRUE \Rightarrow g = f$

Static Part of Model

carrier sets: membership abstracted away

sets: D $BOOLEAN$

data item

constants: n f

f → file on sender

n → max size of file

axioms:

axm0_1 : $n > 0$

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

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

Dynamic Part of Model

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

variables: g, b

b → whether or not the transmission has been completed

invariants:

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

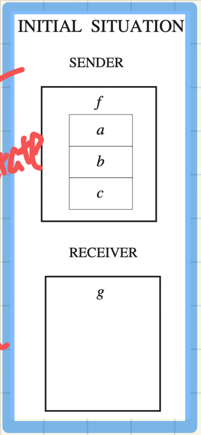
inv0_1b : $b \in BOOLEAN$

inv0_2 : * ??

inv0_3 : ** ??

conditional invariants

FTP: Events of Initial Model



post-state of init event

sets: $D, BOOLEAN$

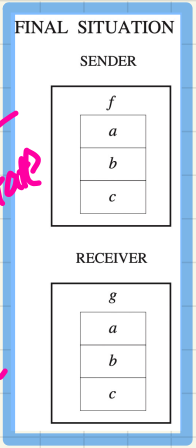
constants: n, f

axioms:

axm0_1 : $n > 0$

axm0_2 : $f \in 1 \,..\, n \to D$

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



post-state of final event

variables: g, b

invariants:

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

inv0_1b : $b \in BOOLEAN$

inv0_2 : $b = FALSE \Rightarrow g = \varnothing$

inv0_3 : $b = TRUE \Rightarrow g = f$

Testing

init:

sender's file ready for transmission

```
init
  begin
    ??
  end
```

enables

$g := \varnothing$

$b := FALSE$

final:

sender's file transmitted to receiver

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

$b = FALSE$

$g := f$

$b := TRUE$

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

PO of Invariant Establishment

sets: $D, BOOLEAN$

constants: n, f

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

variables: g, b

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

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

BAP:
$g' = \emptyset \land b' = FALSE$

Rule of Invariant Establishment

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

Components

K(c): effect of init's actions

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

Exercise: Generate Sequents from the INV rule.

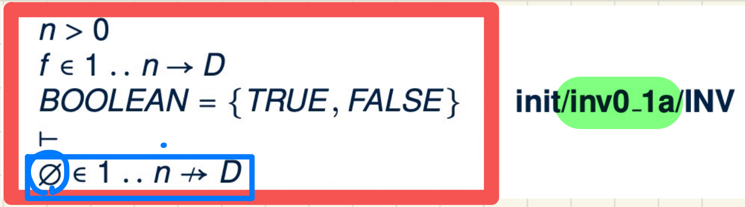
init/inv0_1a/INV

$n > 0$
$f \in 1..n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$\vdash$
$\underset{\emptyset}{g'} \in 1..n \nrightarrow D$

init/inv0_2/INV

$n > 0$
$f \in 1..n \to D$
$BOOLEAN = \{TRUE, FALSE\}$
$\vdash$
$\underset{FALSE}{b'} = FALSE \Rightarrow \underset{\emptyset}{g'} = \emptyset$

Discharging PO of Invariant Establishment



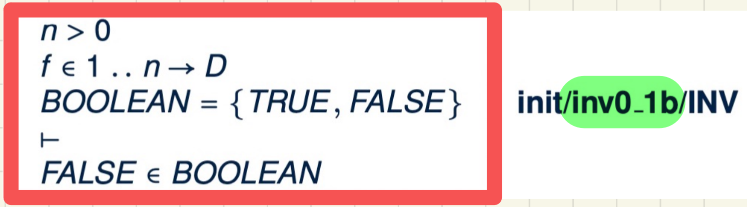
$n > 0$
$f \in 1\,..\,n \rightarrow D$
$BOOLEAN = \{TRUE, FALSE\}$
$\vdash$
$\boxed{\varnothing \in 1\,..\,n \nrightarrow D}$

init/inv0_1a/INV

ARI

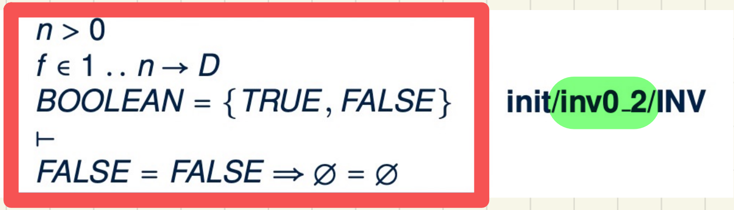
$\boxed{\begin{array}{l} n>0 \\ f \in 1\,..\,n \rightarrow D \\ BOOLEAN = \{TRUE, FALSE\} \\ \vdash T \ \cancel{TRUE} \end{array}}$ TRUE_R

∅ is always a partial function whose domain & range are ∅



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

init/inv0_1b/INV

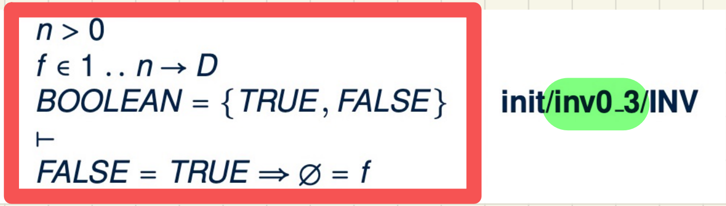


$n > 0$
$f \in 1\,..\,n \rightarrow D$
$BOOLEAN = \{TRUE, FALSE\}$
$\vdash$
$FALSE = FALSE \Rightarrow \varnothing = \varnothing$

init/inv0_2/INV

MON $\boxed{\begin{array}{l} \vdash \\ FALSE = FALSE \Rightarrow \varnothing = \varnothing \end{array}}$ ARI $\boxed{\begin{array}{l} \vdash \\ T \end{array}}$ TRUE_R

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



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

init/inv0_3/INV

PO of Invariant Preservation

sets: $D, BOOLEAN$

constants: n, f

variables: g, b

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

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

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

BAP: $g' = f \land b' = FALSE$

Rule of Invariant Preservation

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

Exercise: Generate Sequents from the INV rule.

final/inv0_1a/INV ✓

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

$* \quad \underset{f}{\cancel{g}} \in 1..n \nrightarrow D$

final/inv0_2/INV

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

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

Discharging POs of m0: Invariant Preservation

final/inv0_1a/INV

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

✓

MON

$$\boxed{\begin{array}{l} f \in 1..n \rightarrow D \\ \vdash \\ f \in 1..n \nrightarrow D \end{array}}$$

ARI

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

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

final/inv0_1b/INV

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

final/inv0_2/INV

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

MON

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

ARI

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

TRUE_R

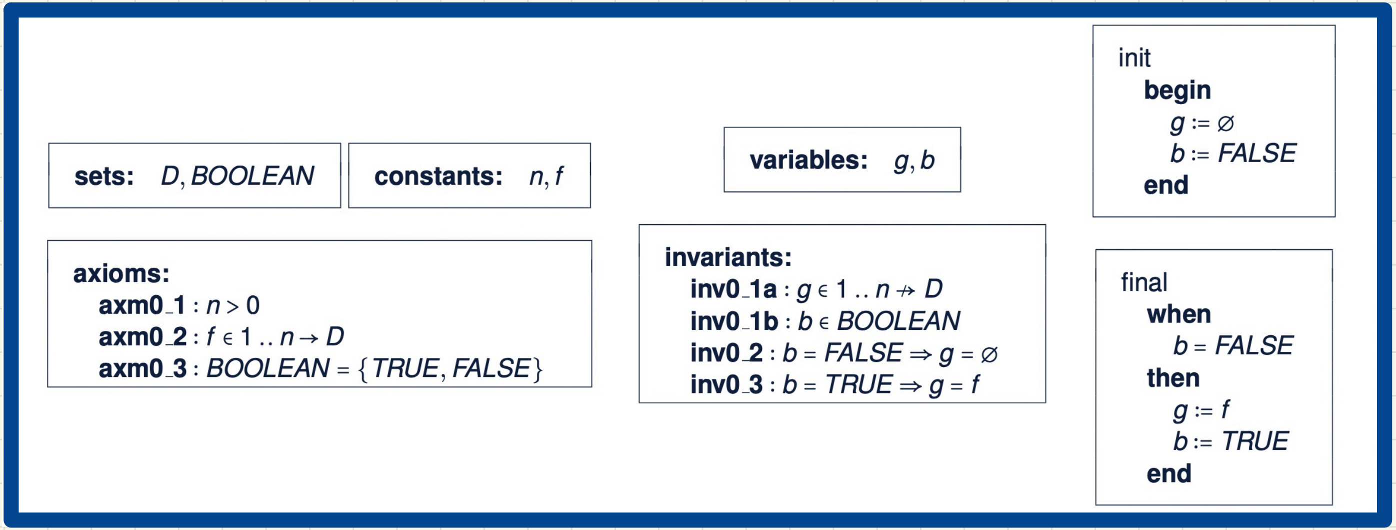
① $TRUE = FALSE \equiv \bot$

② $\bot \Rightarrow P \equiv \top$

final/inv0_3/INV

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

Summary of the Initial Model: Provably Correct



sets: $D, BOOLEAN$

constants: n, f

axioms:

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

variables: g, b

invariants:

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

```
init
  begin
    g := ∅
    b := FALSE
  end
```

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

Correctness Criteria:

+ Invariant Establishment
+ Invariant Preservation
+ Deadlock Freedom

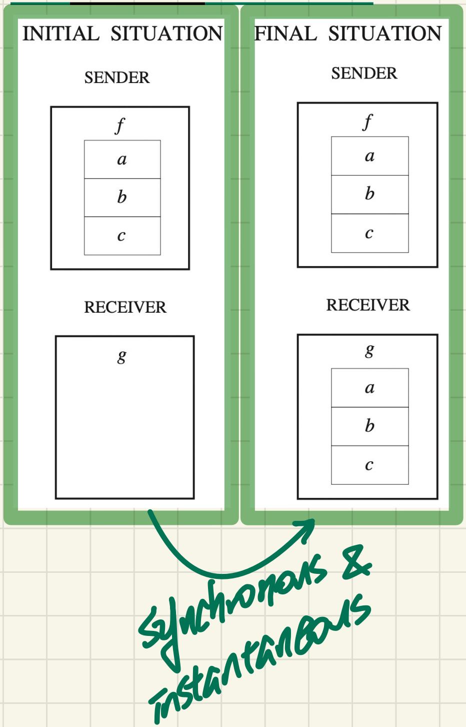
Lecture

Distributed System: File Transfer Protocol

1st Refinement: State, Events, Proofs

FTP: Abstraction in the 1st Refinement

m0: most abstract

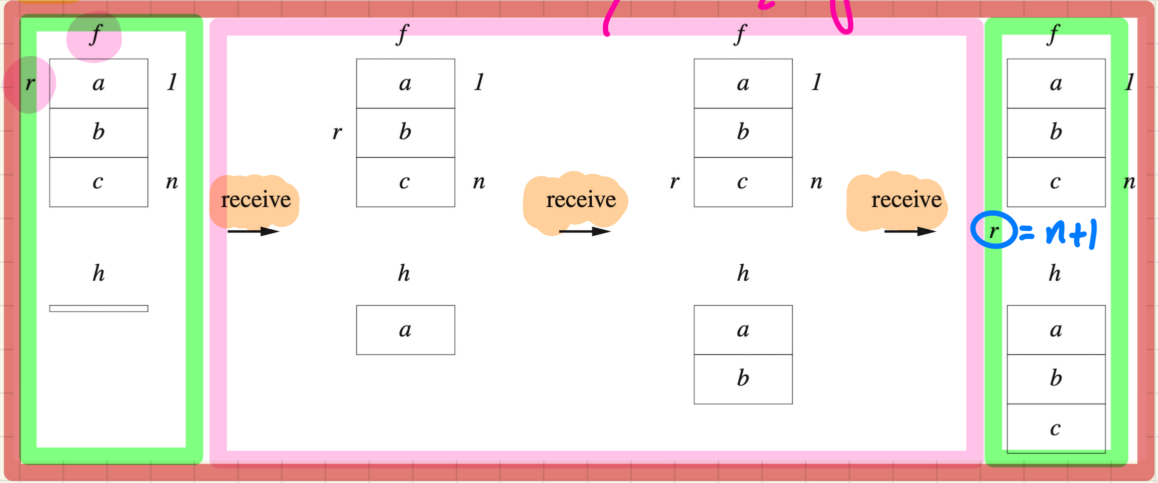


synchronous & instantaneous

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

m1: more concrete than m0

refinement:
1. asynchronous
2. gradual



r = n+1

FTP: State Space of the 1st Refinement

Static Part of Model

sets: $D, BOOLEAN$

constants: n, f

axioms:

axm0_1 : $n > 0$

axm0_2 : $f \in 1 \,..\, n \to D$

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

Dynamic Part of Model

variables: b, h, r

invariants:

inv1_1 : $r \in 1 \,..\, n+1$

inv1_2 : ?? *

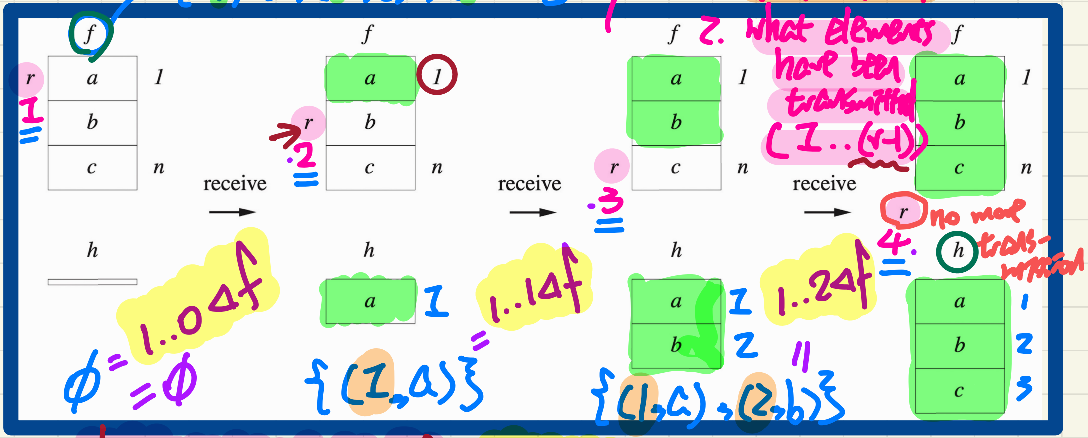
inv1_3 : ?? **

thm1_1 : ?? ***

to be proved for establishment & preservation

1. need not be proved for establishment & preservation

2. to be proved as derivable from invariants



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

* $h = (1 \,..\, (r-1)) \lhd f$

$\{1, 2, \dots, r-1\}$

$1..0 = \emptyset$

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

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

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

$= 1..4 \lhd f$

dom(f)

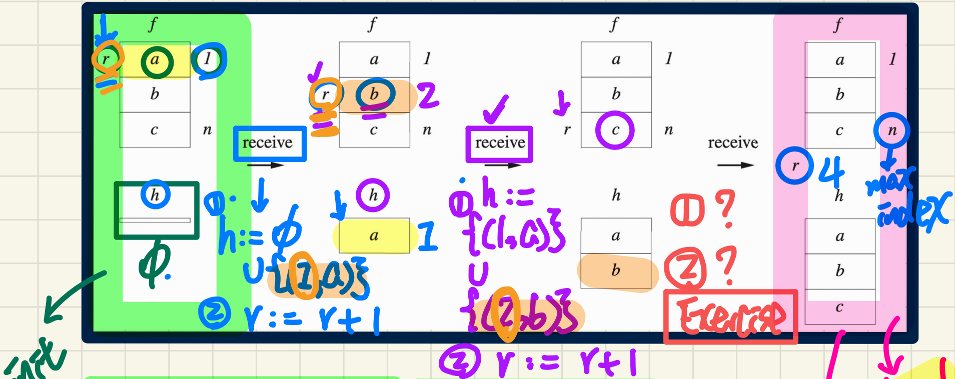
Exercises

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

inv1_3: transmission completed means no more elements to be transmitted

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

FTP: Concrete Events in 2nd Refinement



sets: $D, BOOLEAN$

constants: n, f

axioms:

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

variables: b, h, r

invariants:

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

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

init: getting the transmission ready

```
init
  begin
    ??
  end
```

$b := FALSE$

$h := \emptyset$

$r := 1$

receive: transmitting element by element

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

$r \leq n$

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

sender's private info should be hidden

occurrence of final is restricted to 1

final: finalizing the transmission

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

$\boxed{b = FALSE}$

$r = n+1$

$b := TRUE$

I hope you enjoyed learning with me

All the best to you !