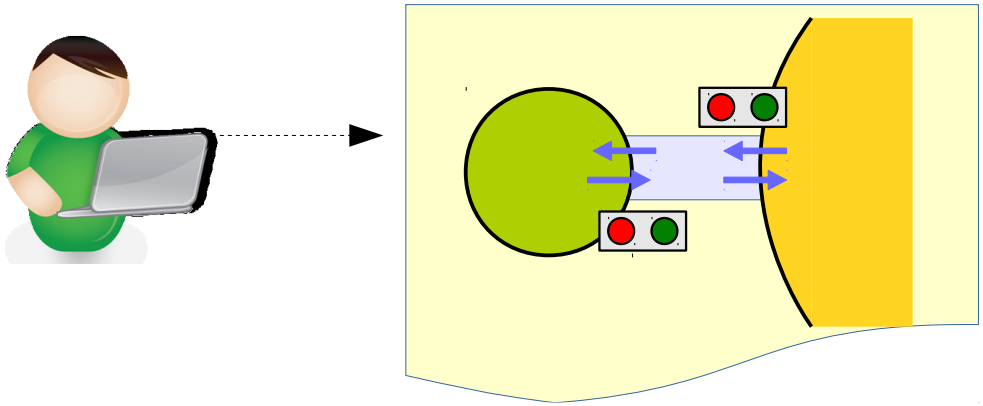
SliceAndMerge: A Rodin Plug-in for Refactoring Refinement Structure of Event-B Machines

Tsutomu Kobayashi (University of Tokyo),
Aivar Kripsaar (RWTH Aachen University),
Fuyuki Ishikawa (NII, Japan),
and Shinichi Honiden (NII, Japan)

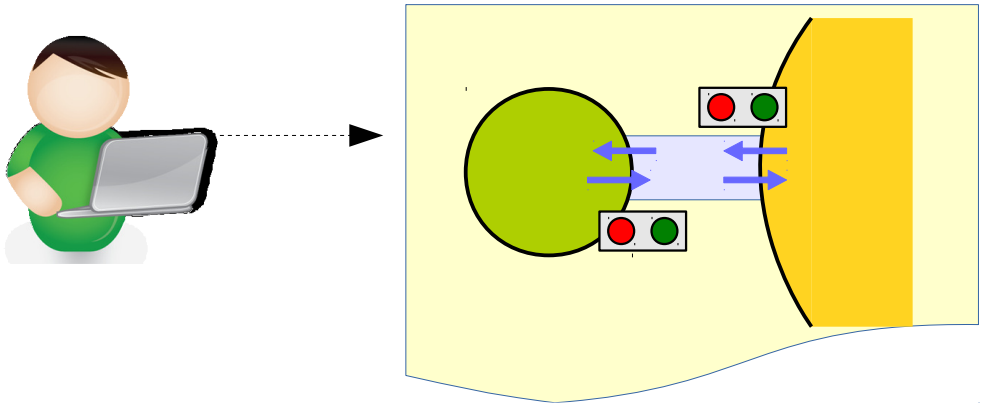
Rodin Workshop 2016
May 23, 2016

Modeling in Event-B

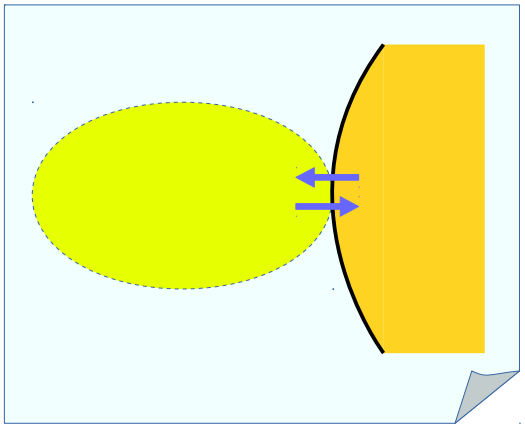


Informal document of traffic control system

Modeling in Event-B

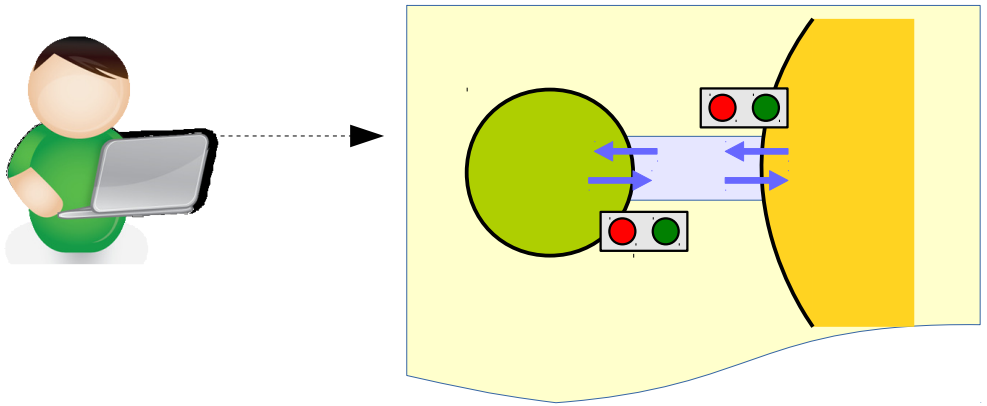


Informal document of traffic control system

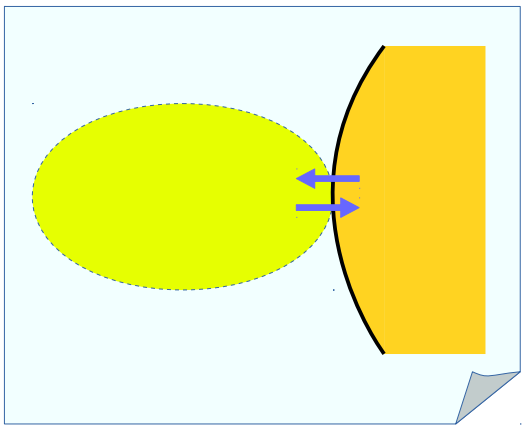


Abstract machine

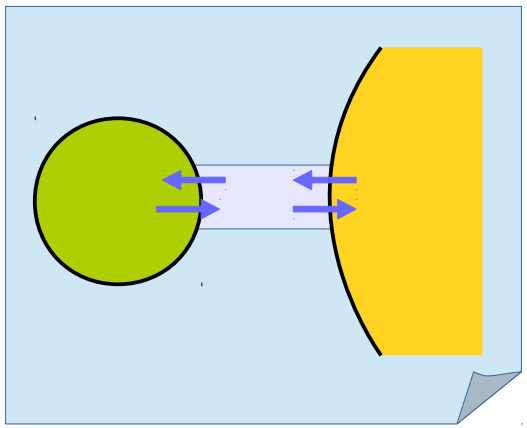
Modeling in Event-B



Informal document of traffic control system

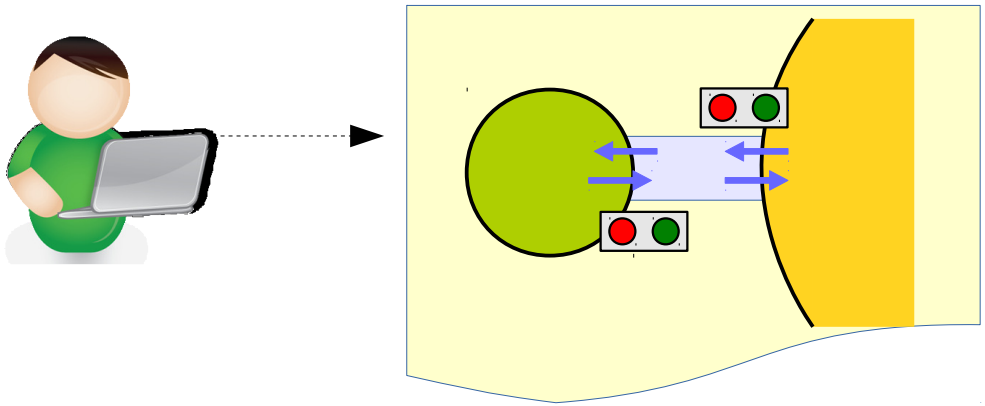


Abstract machine



Medium machine

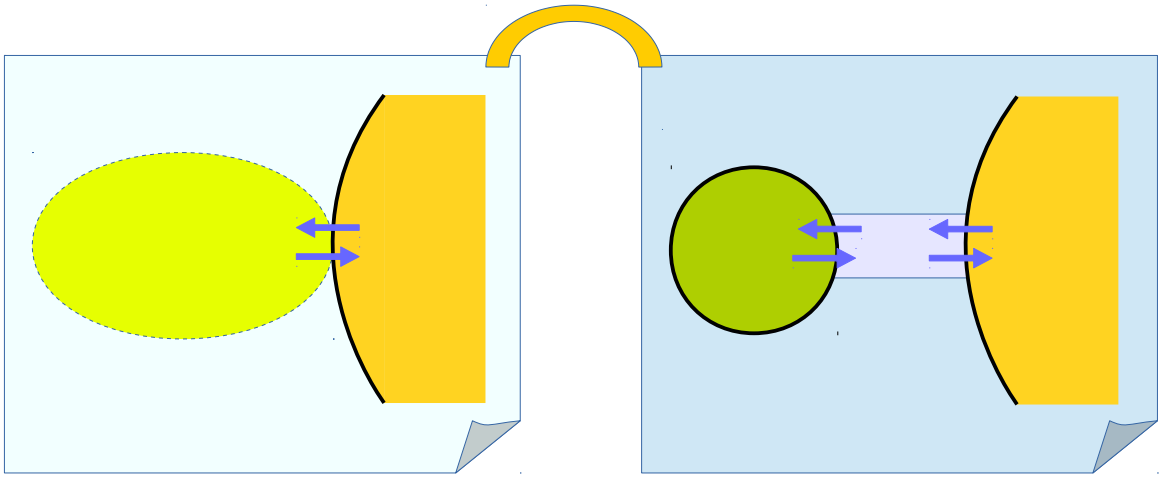
Modeling in Event-B



Informal document of traffic control system

prove

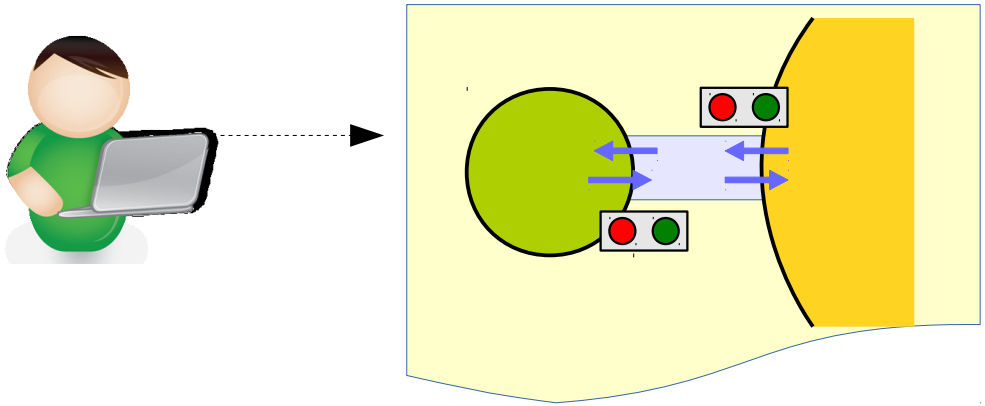
Consistent?



Abstract machine

Medium machine

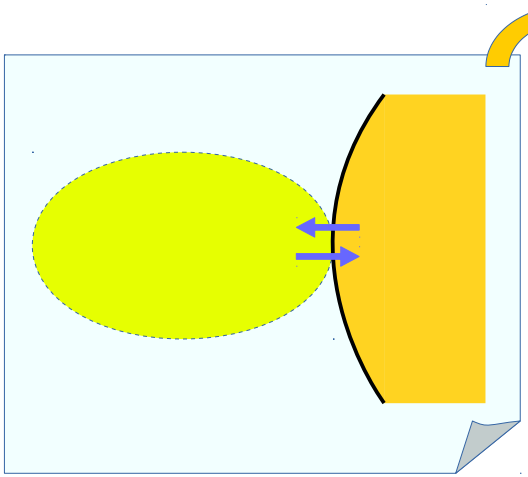
Modeling in Event-B



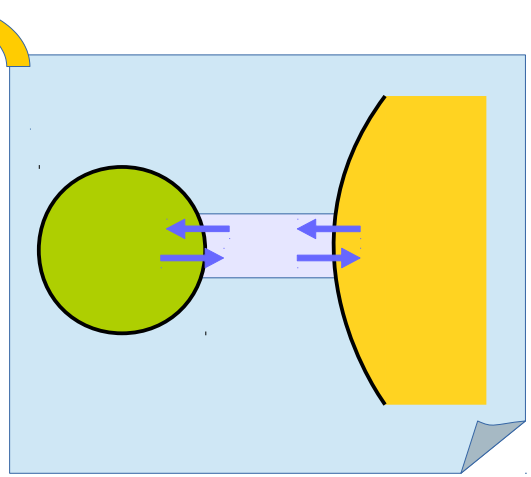
Informal document of traffic control system

prove

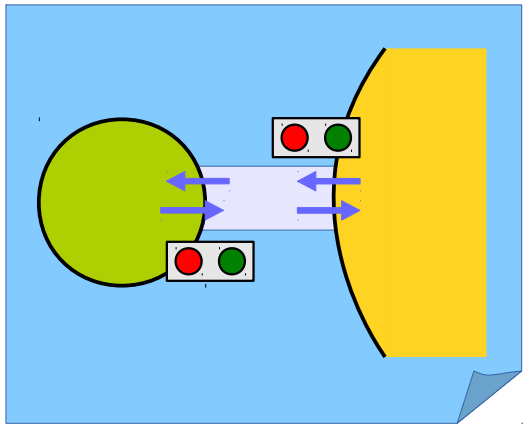
Consistent?



Abstract machine

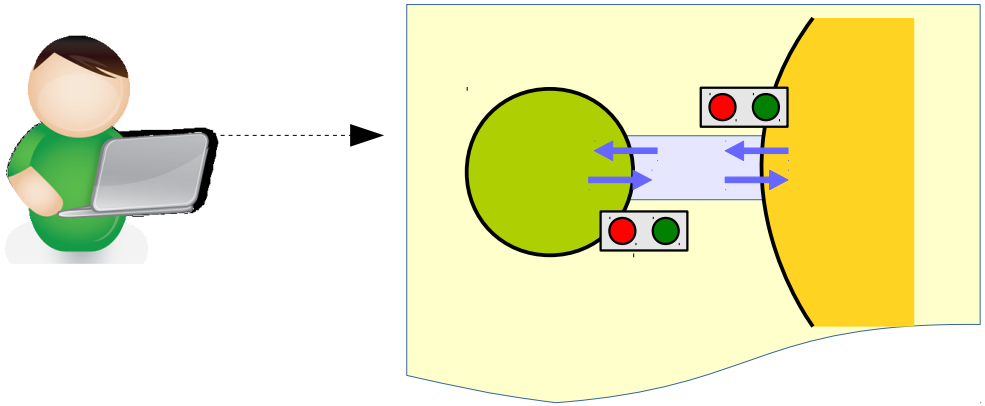


Medium machine



Concrete machine

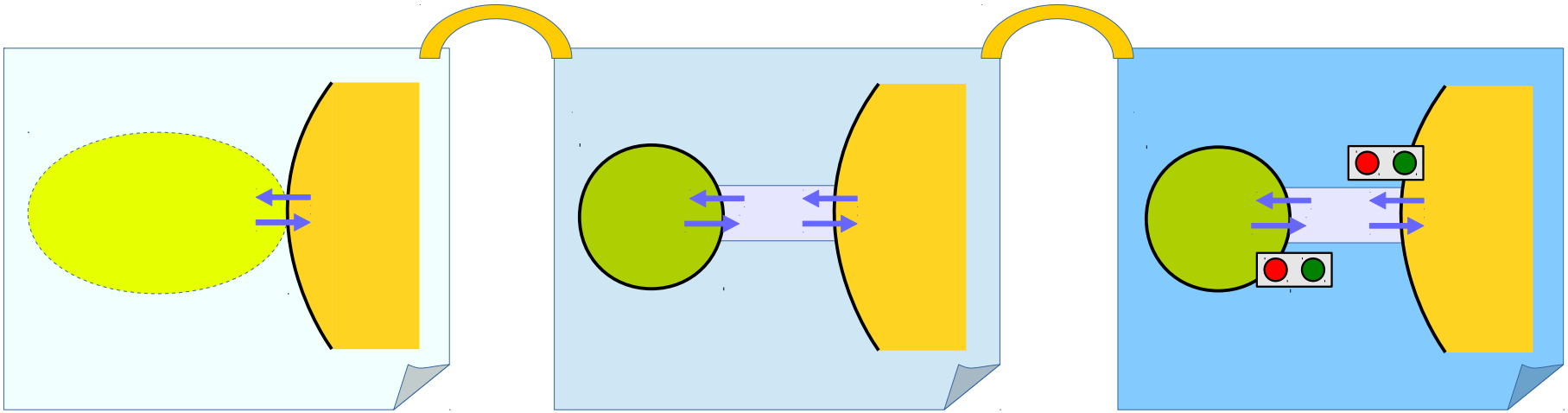
Modeling in Event-B



Informal document of traffic control system

prove ✓ Consistent?

prove ✓ Consistent?



Abstract machine

Medium machine

Concrete machine

Motivation

Refinement design and specifications

Designing refinement = designing target system's aspects of interest

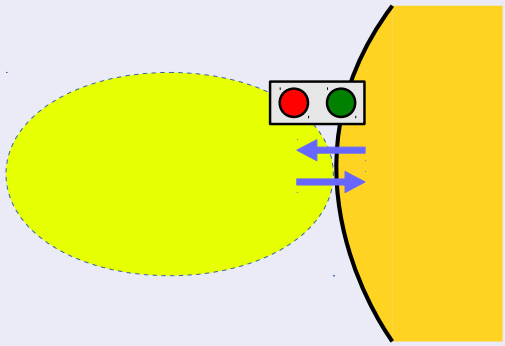
e.g., Focus on

- the outside
- traffic lights on the mainland

↓

Specify/verify properties of them

- *If traffic light is green,*
  # cars outside $\leq$ *capacity*



Our goal

Modify refinement design of existing specification
→ improve understandability, maintainability, extensibility

Example of Motivation

Problem

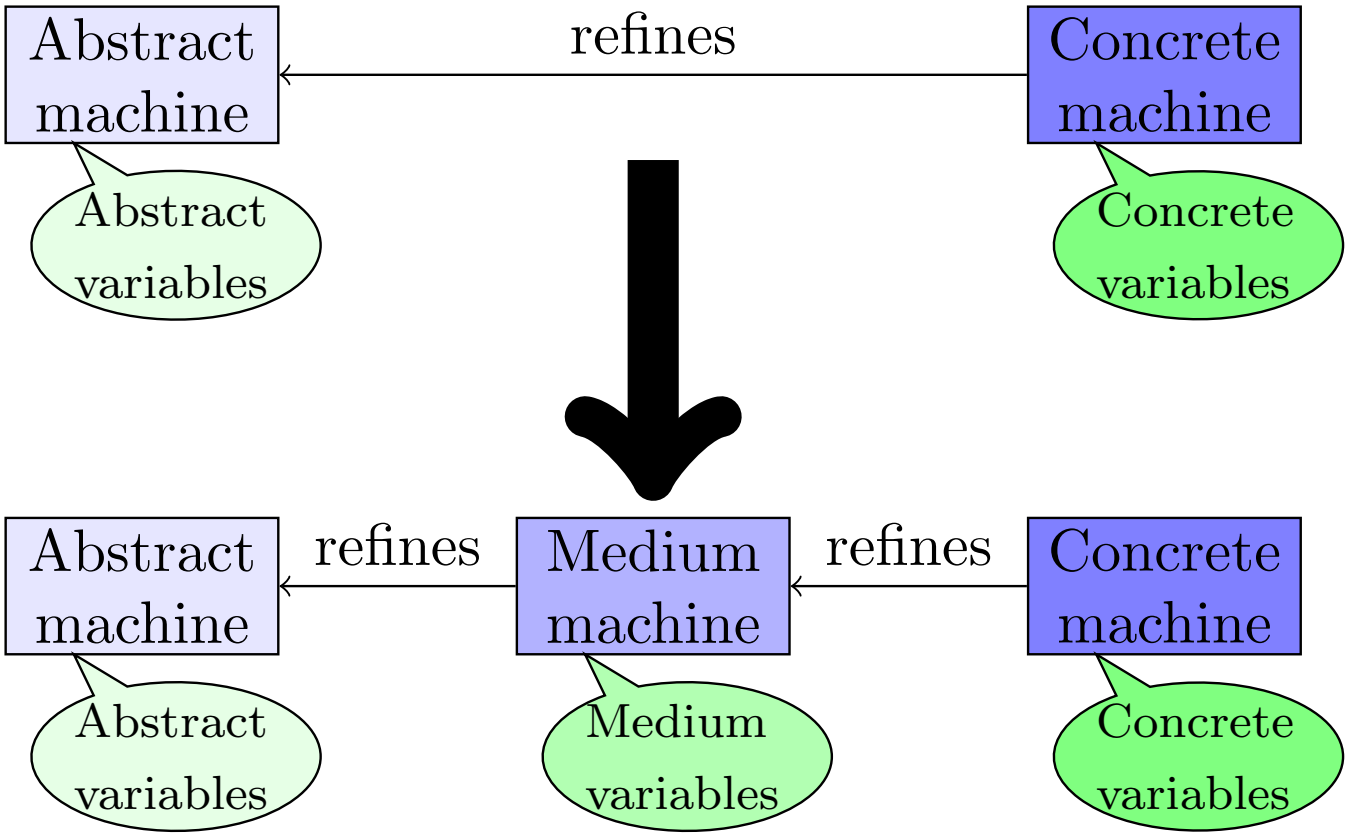
Somtimes we make refinements with many additional variables/invariants

- To specify several aspects in a step

Solution

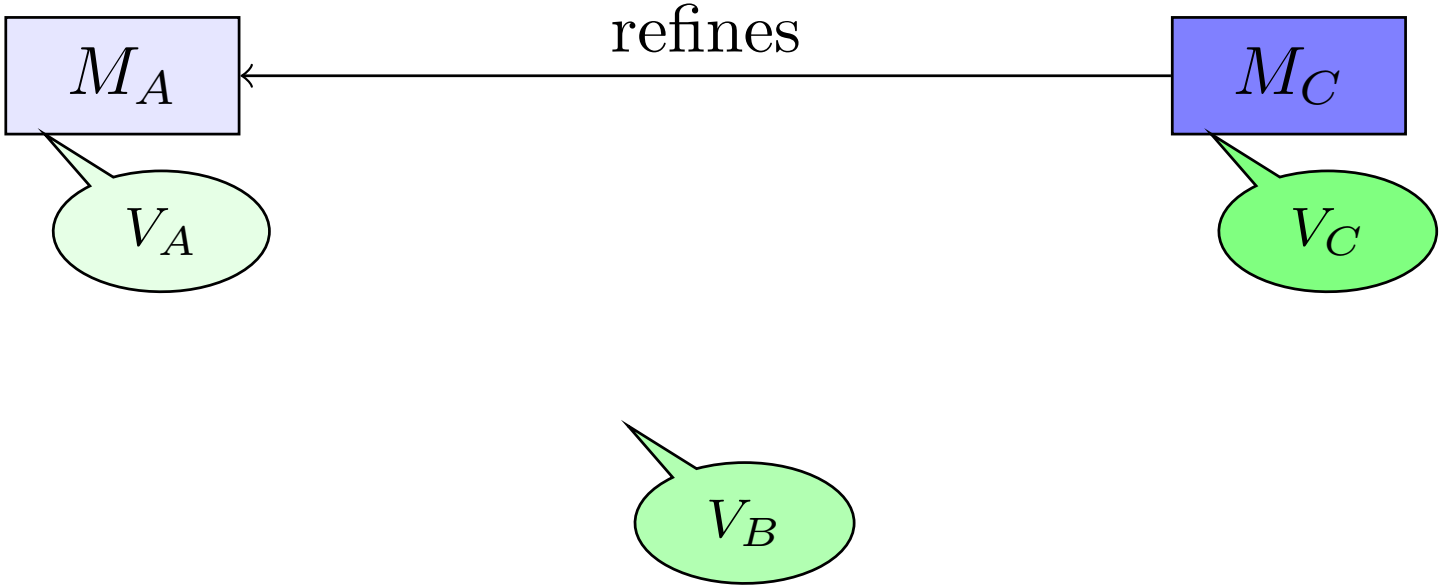
Refinement decomposition

Refactoring of Refinement – Decomposition



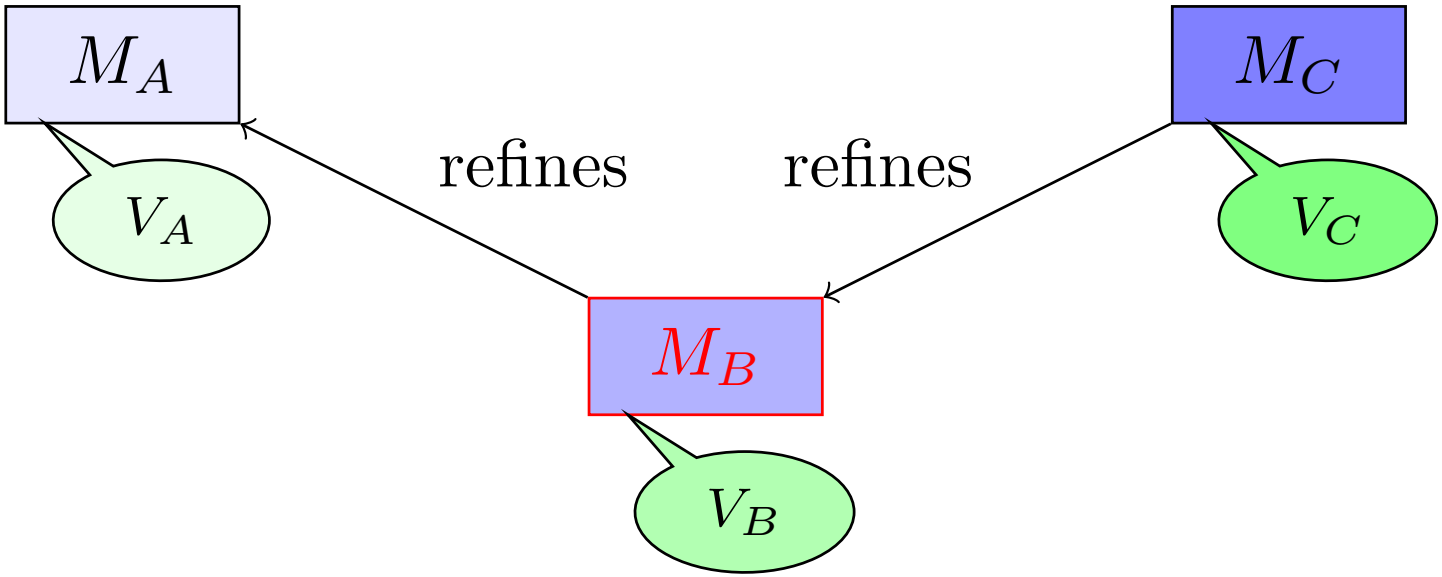
- Decomposition of refinement

Goal of Refinement Decomposition



- Input
  - Proved machines M_A and M_C
  - A set of variables V_B (subset of V_C, slicing criteria)

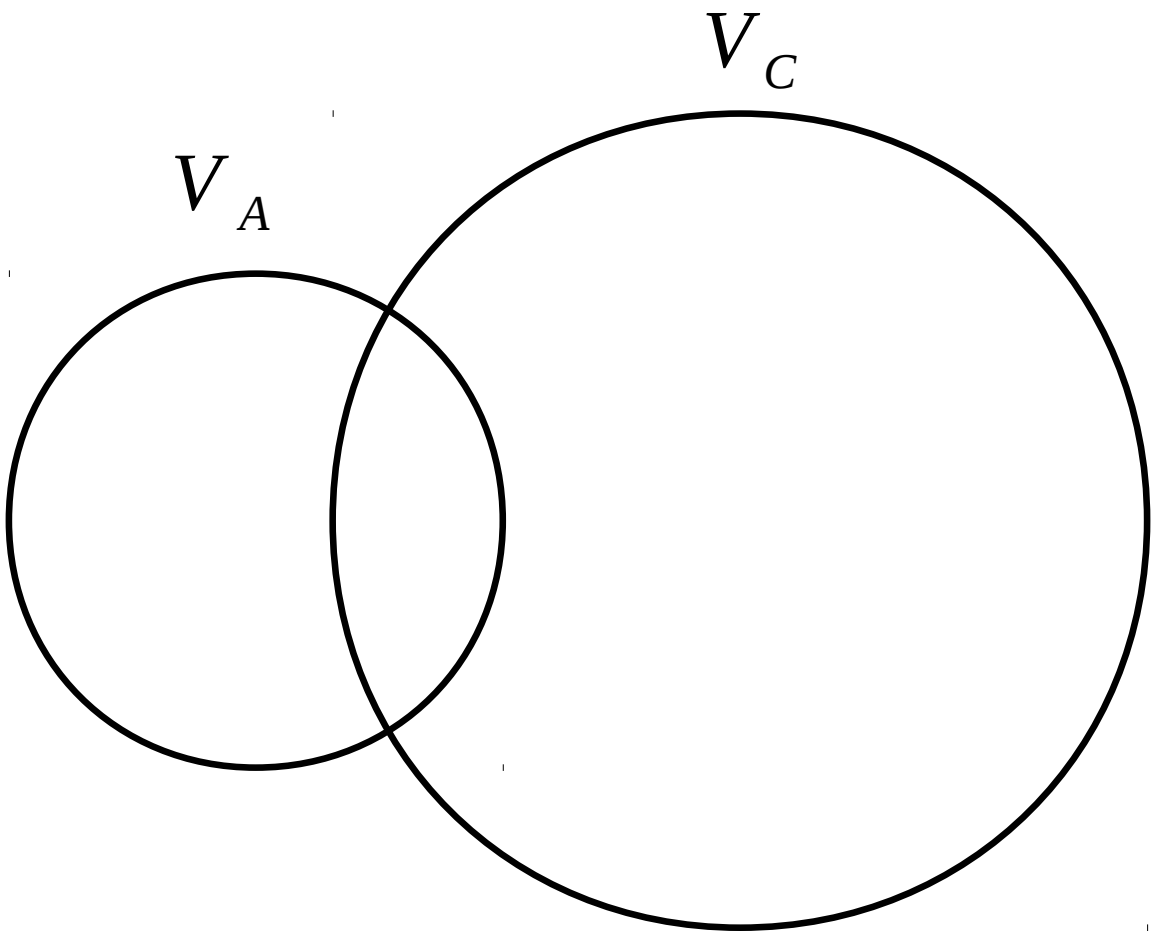
Goal of Refinement Decomposition



- Input
  - Proved machines M_A and M_C
  - A set of variables V_B (subset of V_C, slicing criteria)
- Output: Intermediate machine M_B such that
  - M_C refines M_B and M_B refines M_A
  - M_B is specified in V_B

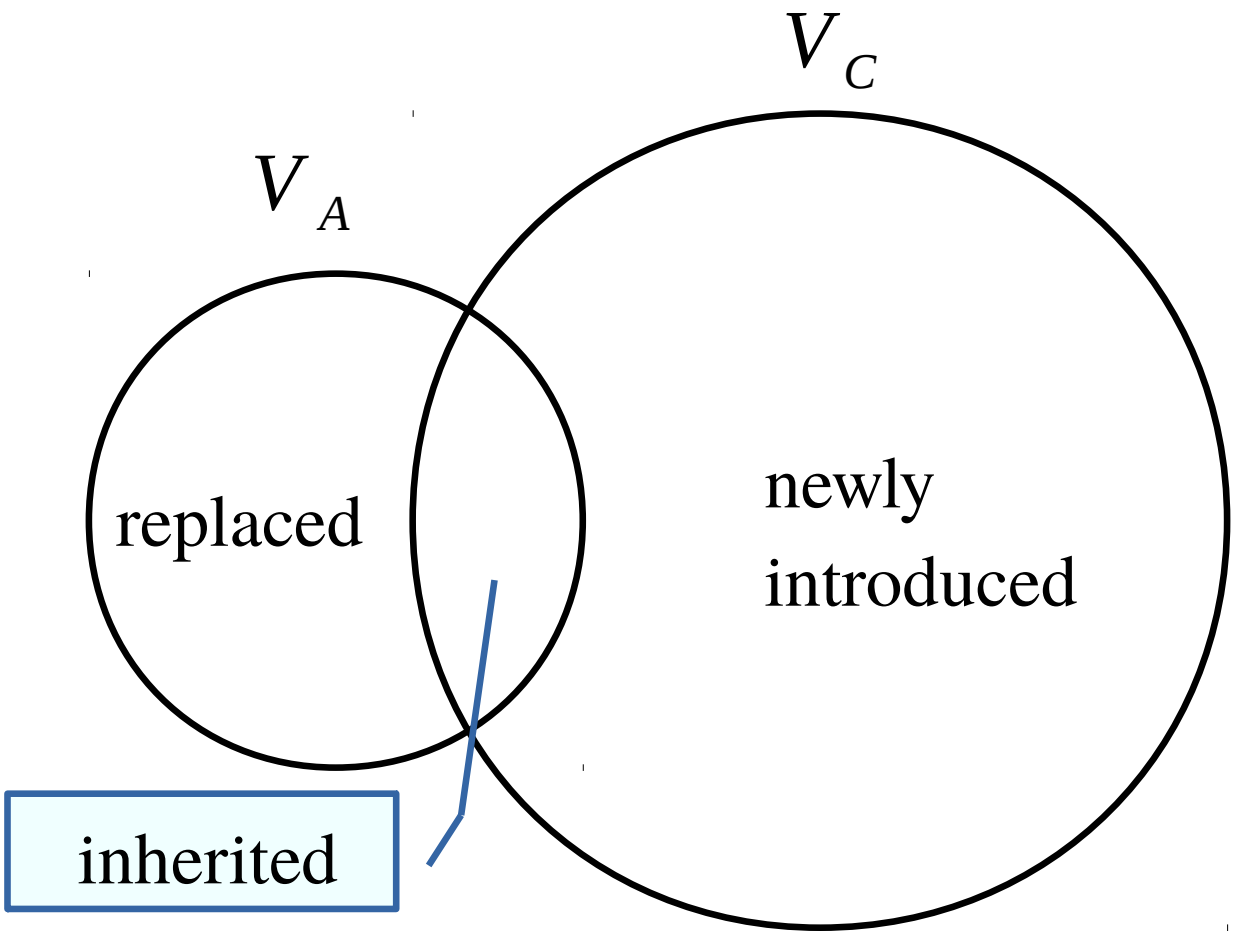
Restriction on V_B

Variables in the machines:



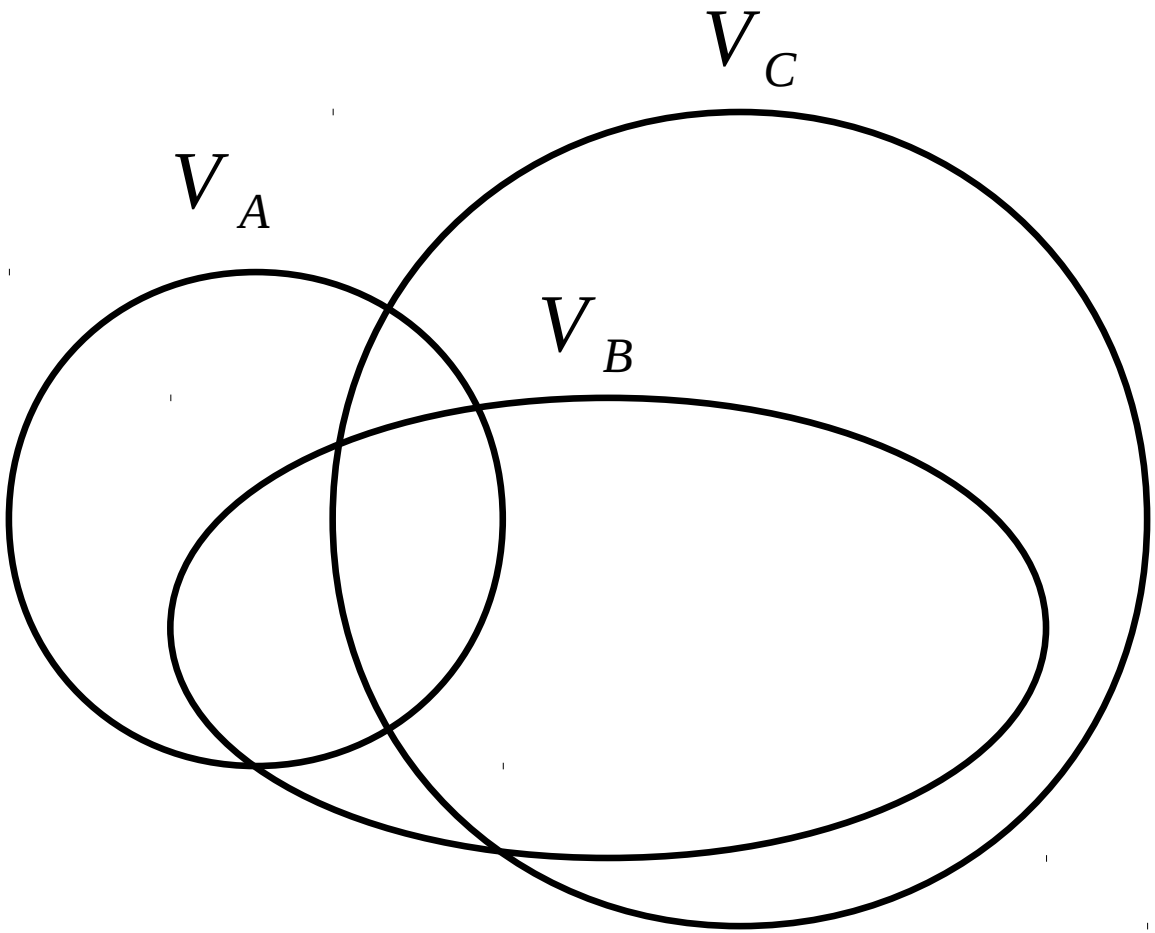
Restriction on V_B

Variables in the machines:



Restriction on V_B

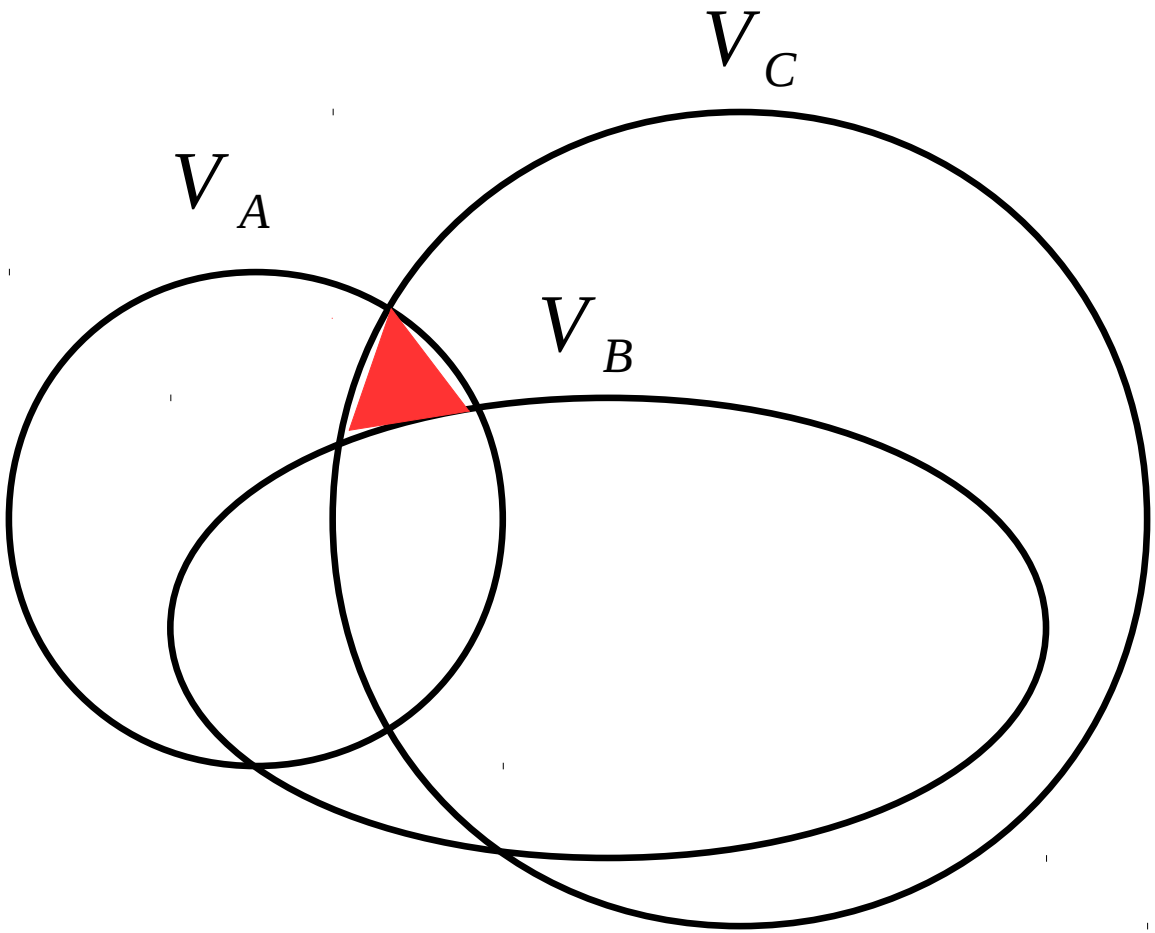
Variables in the machines:



Arbitrary V_B:

Restriction on V_B

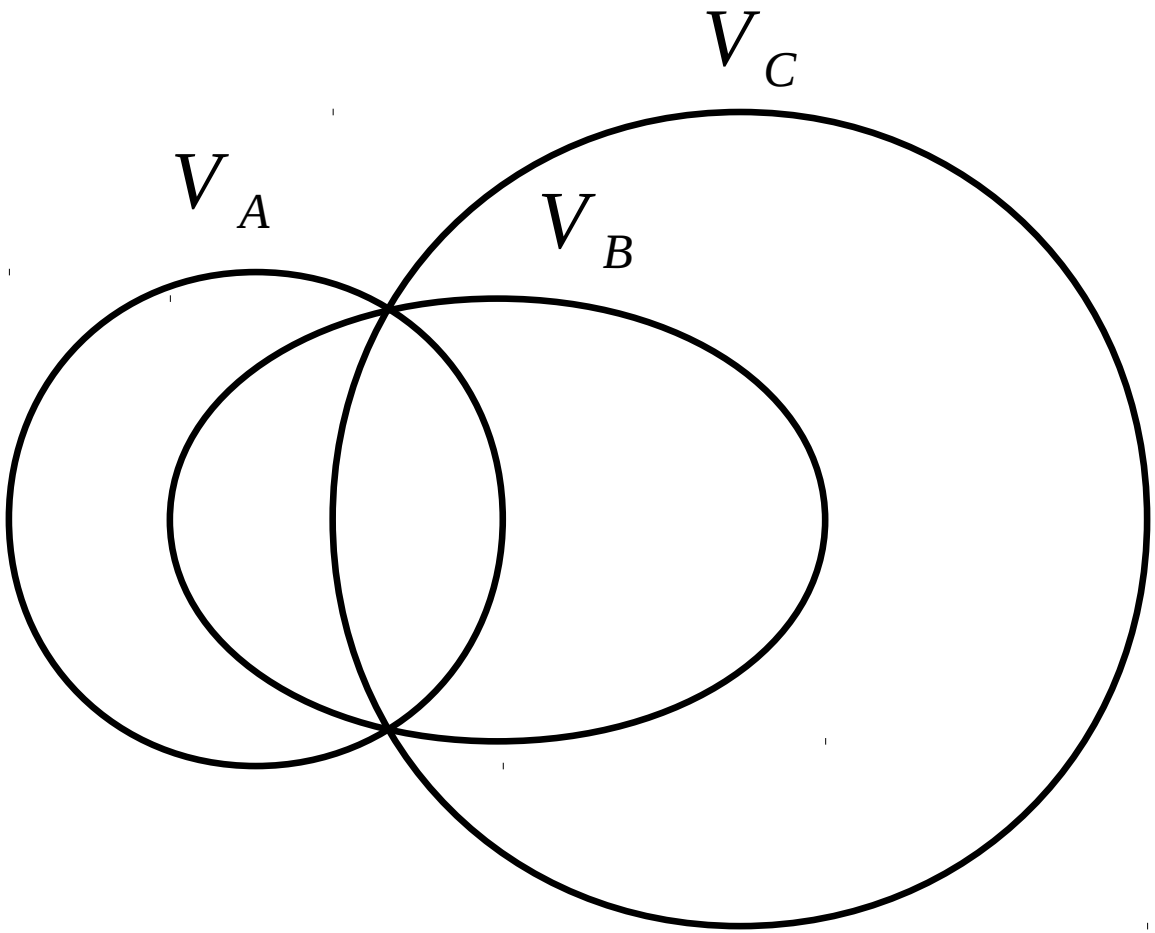
Variables in the machines:



Arbitrary V_B: Some variables are in V_A and V_C but not V_B !

Restriction on V_B

Variables in the machines:



$=>$ V_B should be a superset of $V_A \cap V_C$

Approach to Decomposing Refinement

1. Slicing from original machines M_C and M_A

by finding specifications that can be expressed by V_B

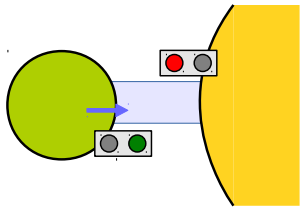
2. Mending for consistency

by providing complementary predicates
to fill the gap originated from slicing

Slicing

Find predicates that can be expressed by V_B
e.g.,

M_C $(TL_{Island} \in V_C)$



Invariants

$TL_{Mainland} = red \lor TL_{Island} = red$

$TL_{Mainland} = green \Rightarrow n_{\rightarrow} = 0$

Event `leave_island`

when

$TL_{Island} = green$

$1 \leq n_{Island}$

then

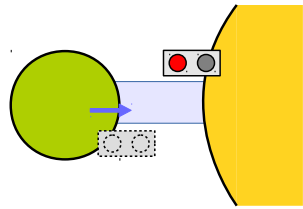
$n'_{Island} = n_{Island} - 1$

$n'_{\rightarrow} = n_{\rightarrow} + 1$

end

$\xrightarrow{slice}$

M_B $(TL_{Island} \notin V_B)$



Invariants

$TL_{Mainland} = green \Rightarrow n_{\rightarrow} = 0$

Event `leave_island`

when

$1 \leq n_{Island}$

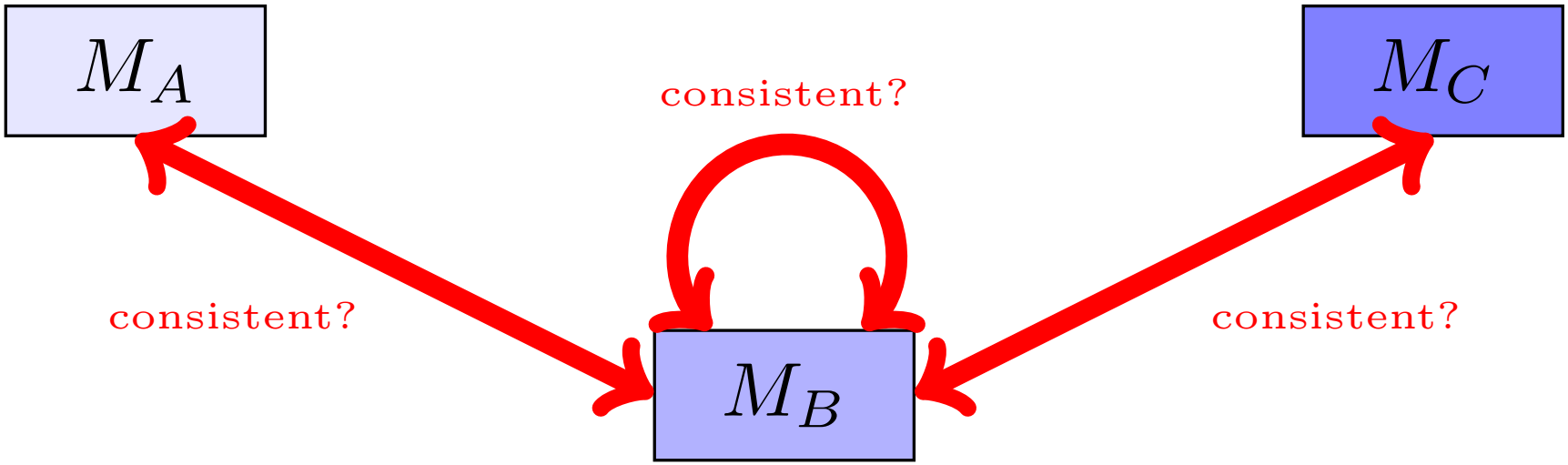
then

$n'_{Island} = n_{Island} - 1$

$n'_{\rightarrow} = n_{\rightarrow} + 1$

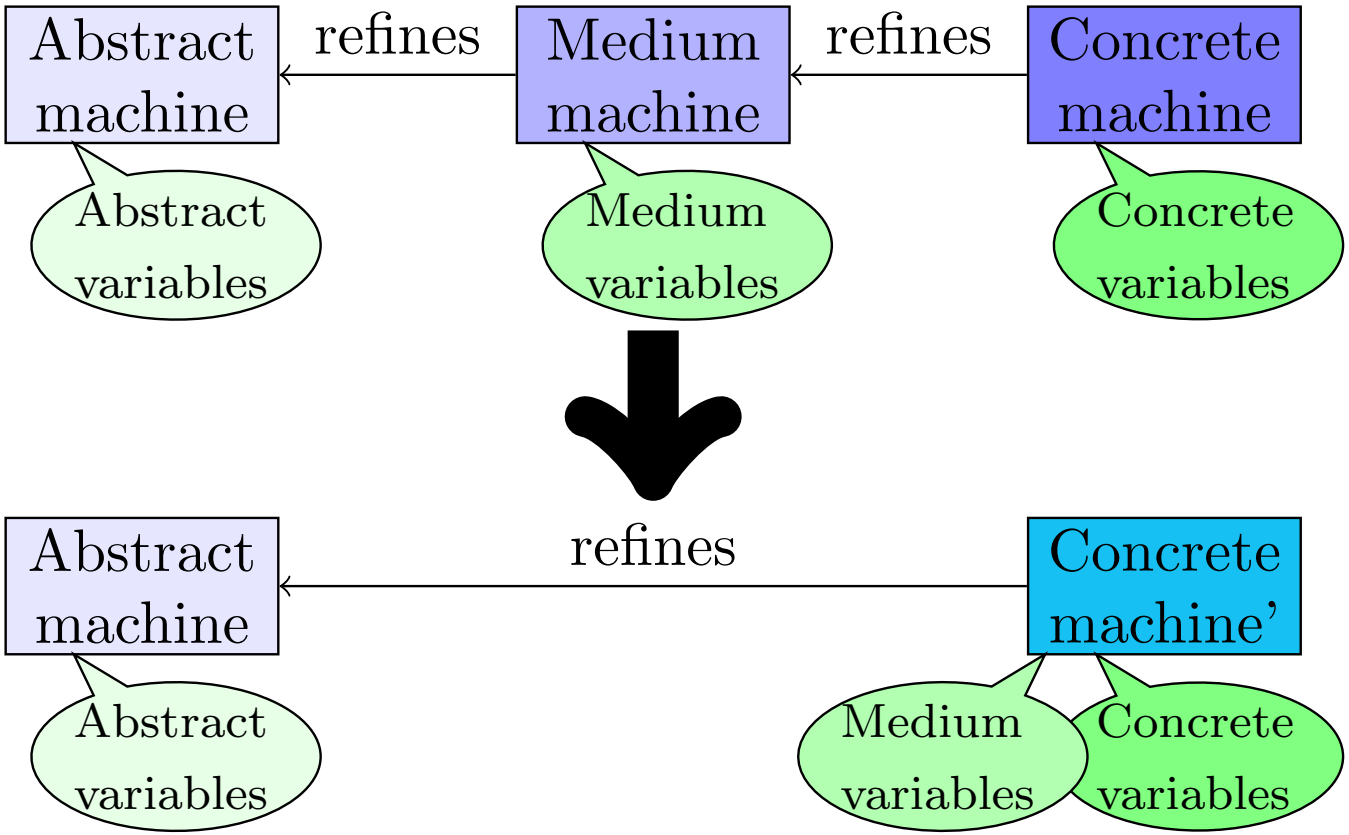
end

Intermediate Machine is Not Always Consistent



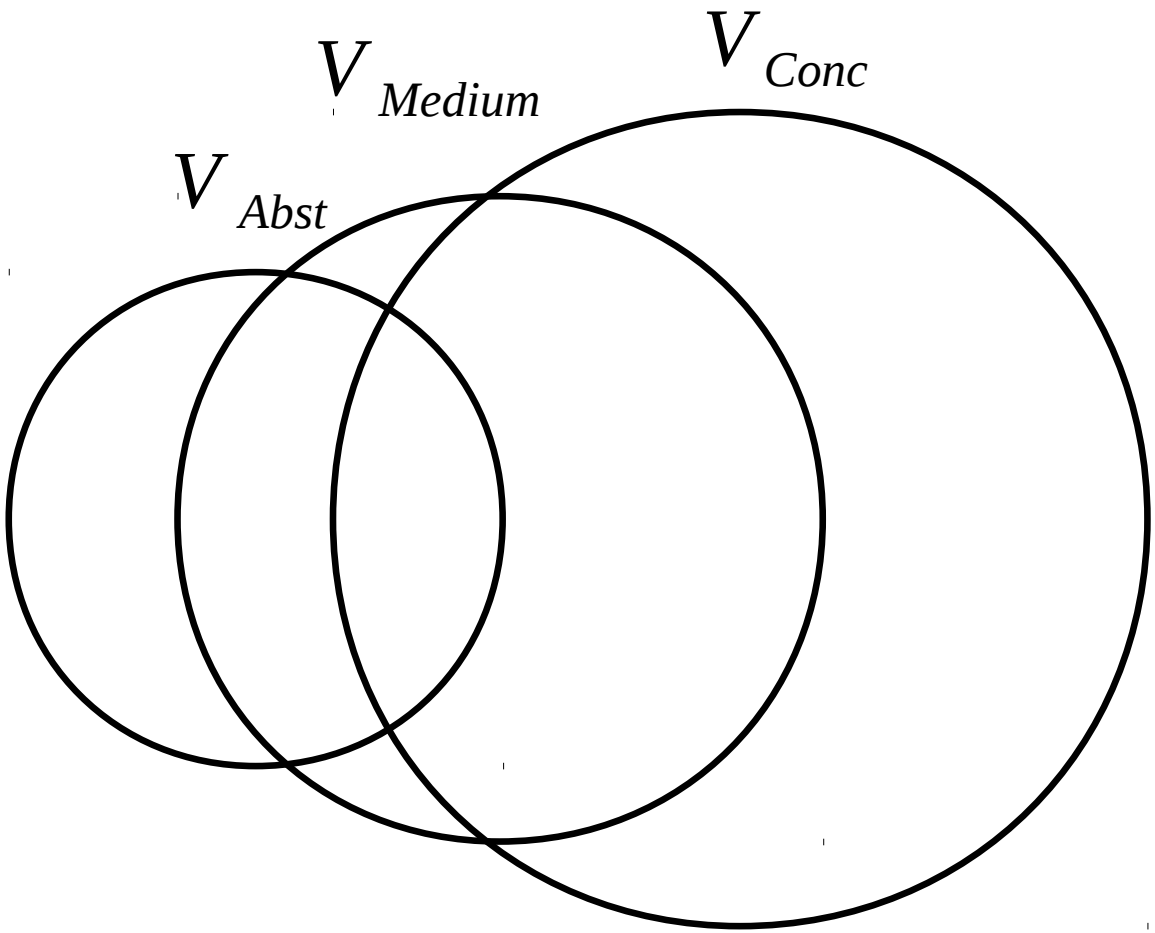
Sometimes we need to guarantee these consistencies

Refactoring of Refinement – Composition

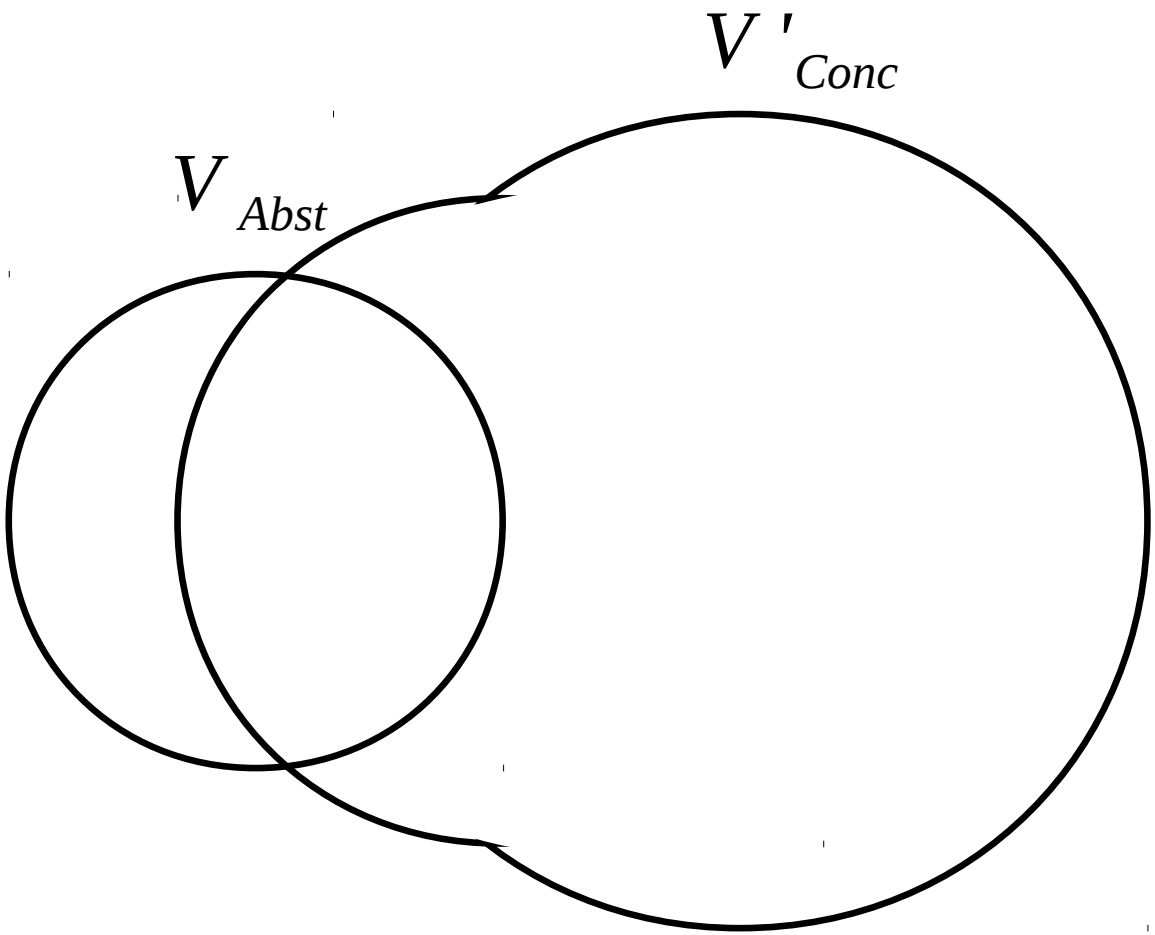


- Composition of refinement

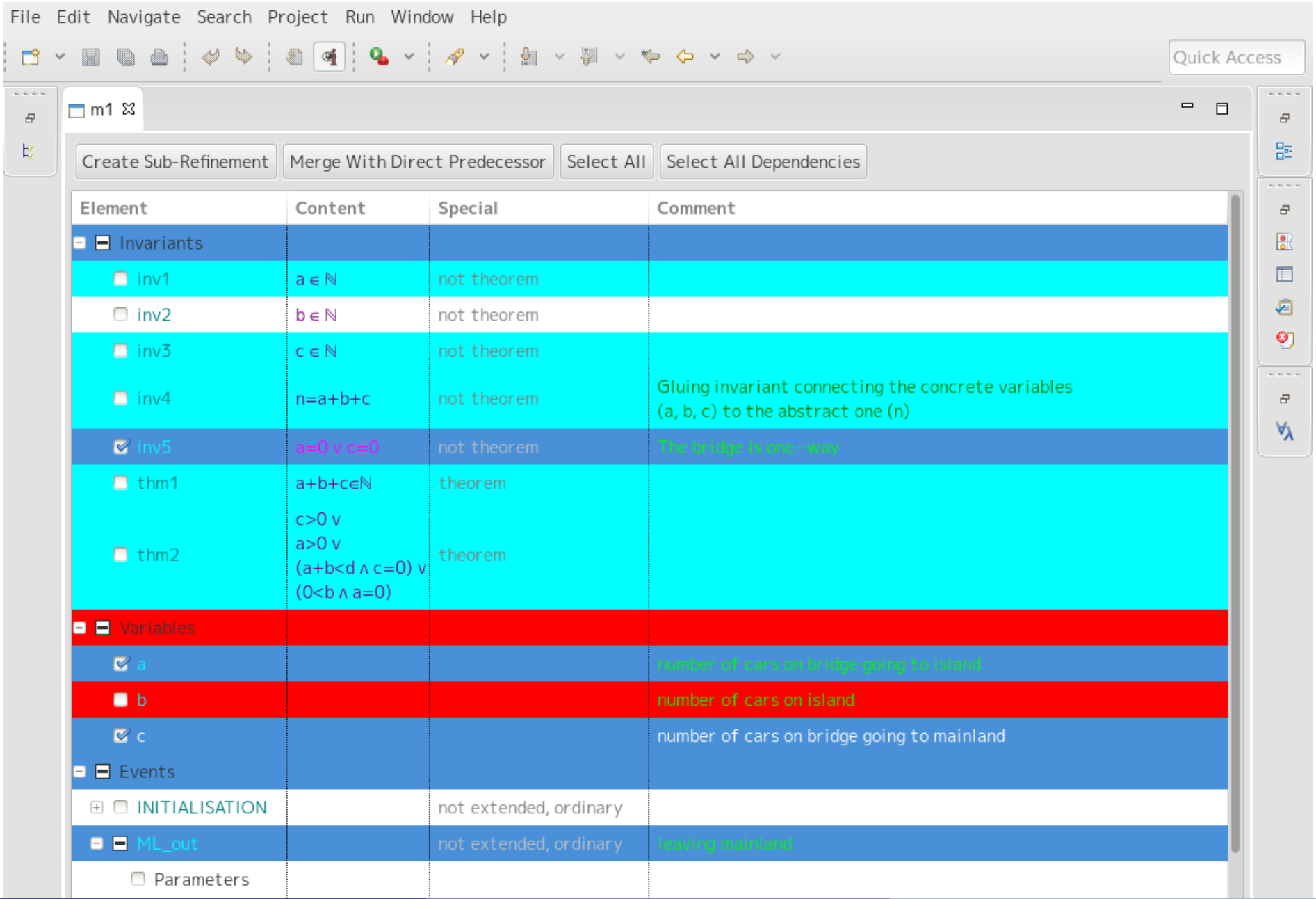
Refactoring of Refinement – Composition



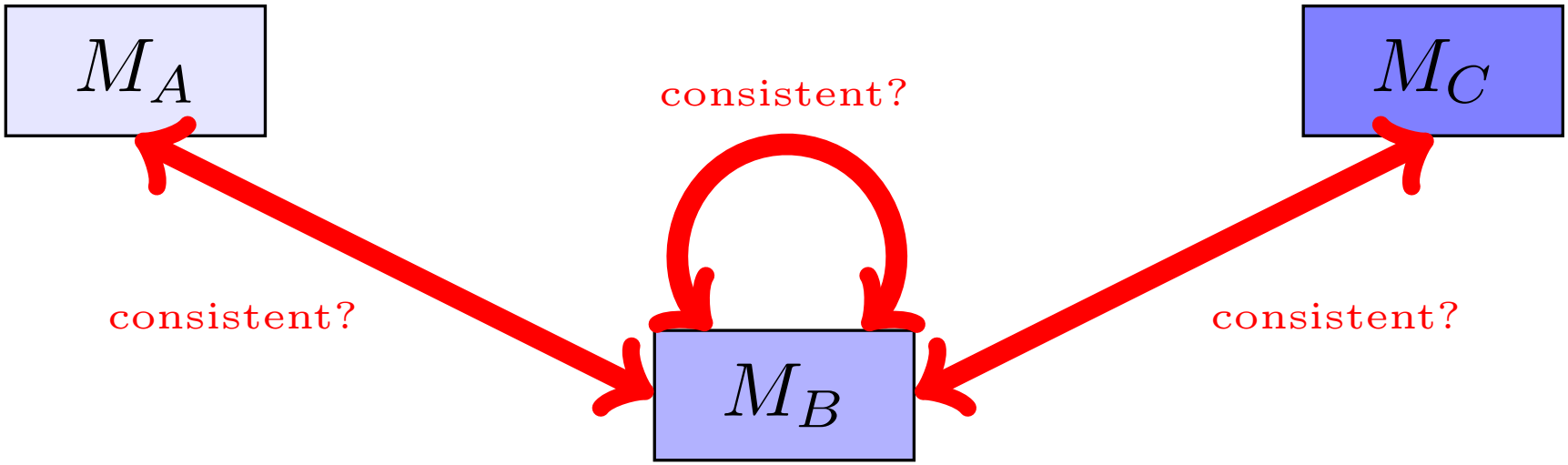
Refactoring of Refinement – Composition



SliceAndMerge: Implementation



Intermediate Machine is Not Always Consistent



Sometimes we need to guarantee these consistencies

Variables and Provability of Proof Obligations

Preservation of $TL_{Mainland}$'s property by "leaving from island"

Invariants		$TL_{Mainland} = red \lor TL_{Island} = red$
Guards		$TL_{Island} = green$
Before-after predicates	$\cdots$	$n'_{\rightarrow} = n_{\rightarrow} + 1$
$\vdash$		$\vdash$
Invariant after the event		$TL_{Mainland} = green \Rightarrow n'_{\rightarrow} = 0$

Provability in {concrete, medium} machine

Concrete machine Provable. $\because TL_{Mainland} \neq green$ from hypotheses

- Because of TL_{Island}-related predicates
  - either $TL_{Mainland}$ or TL_{Island} is red
  - TL_{Island} is green

Medium machine Not provable

- Because of lack of TL_{Island}-related predicates

Mending by Adding Complementary Predicates

Idea

- Original machines are consistent
- M_B lacks essential predicates for consistencies because of vocabulary limitations

$$\downarrow$$

- Find essential predicates
- Express them in vocabulary of V_B and mend M_B with them

Ways of mending include:

- Heuristics such as extracting a predicate P from $P \wedge Q$
- Analyzing the proof (as described in previous slide)
  1. Trace the proof of original machine's consistency
  2. Infer the complementary predicates from that
- Using *Craig's interpolation theorem*?

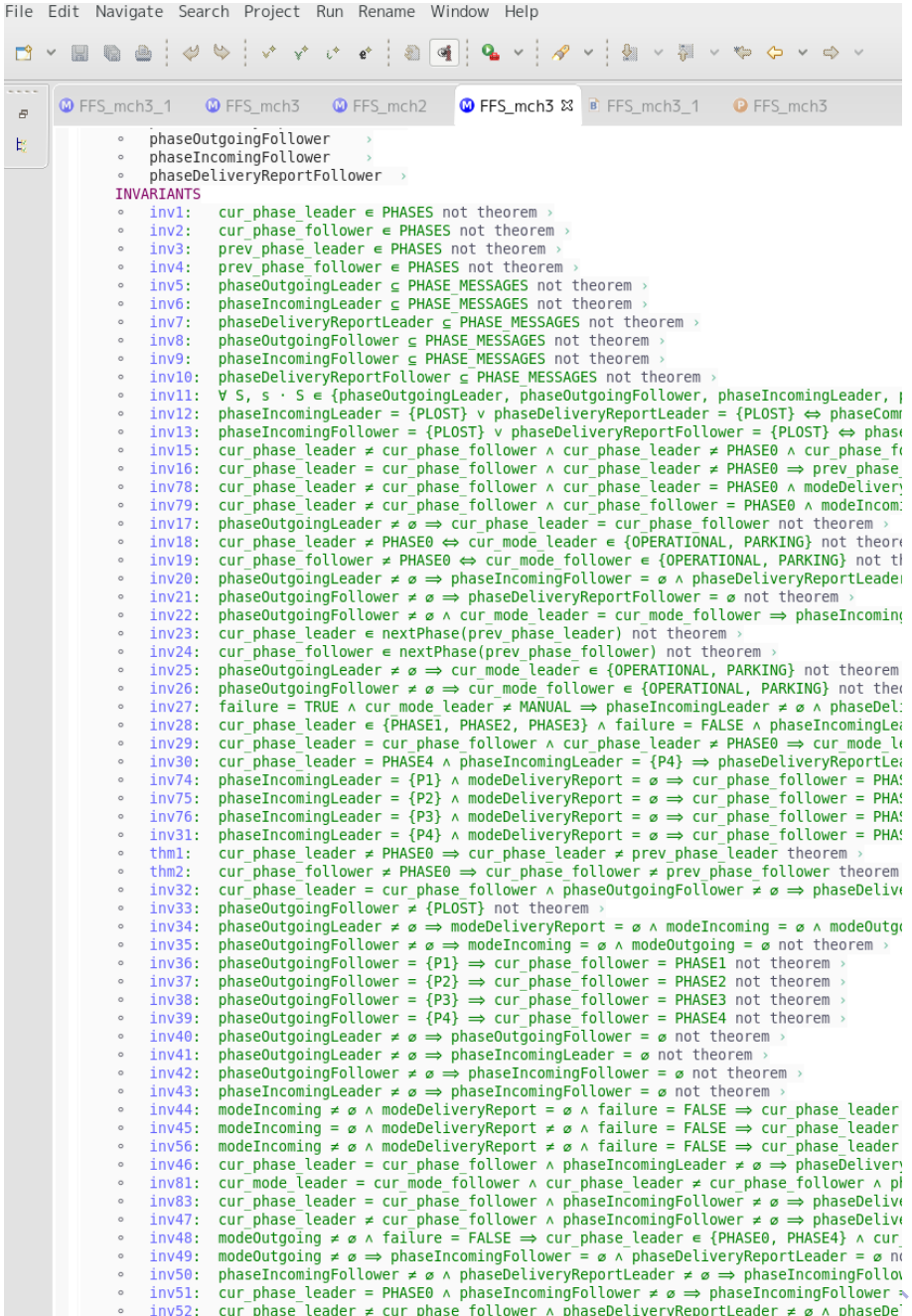
Case Study: Decomposing Refinement

Example

- Model of flight formation of satellites
  - by an experienced modeler
  - high-quality, but include large refinement steps
    - 72 invariants in one refinement

Result

- Decomposed refinements into multiple steps



Results

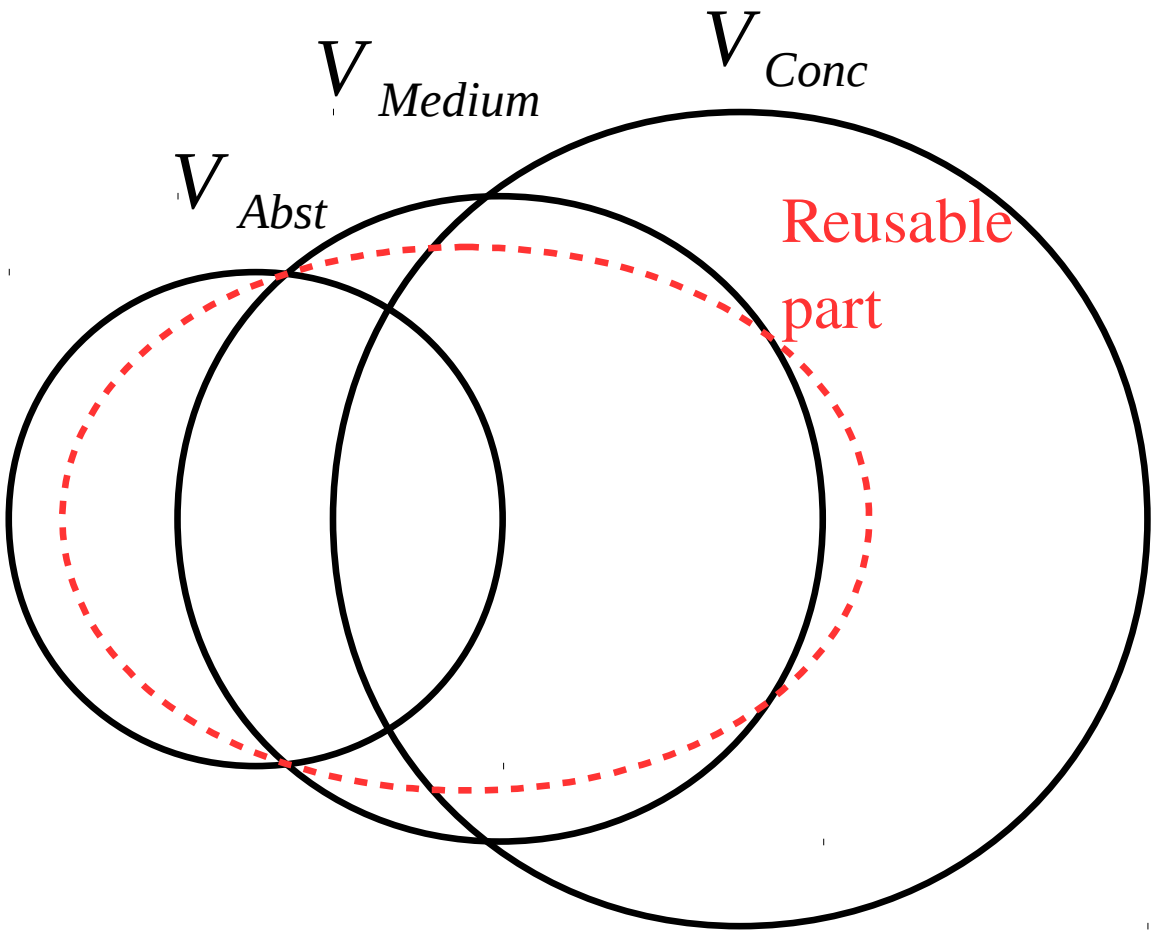
original machines / intermediate machines	machine2	1st	2nd	3rd	4th
# variables	+7	+3	+2	+1	+1
# invariants	46	11	10	13	12
# complement preds		12	13	10	0
(# unique comp preds)		(5)	(7)	(5)	(0)

Distributed introduction of variables/invariants

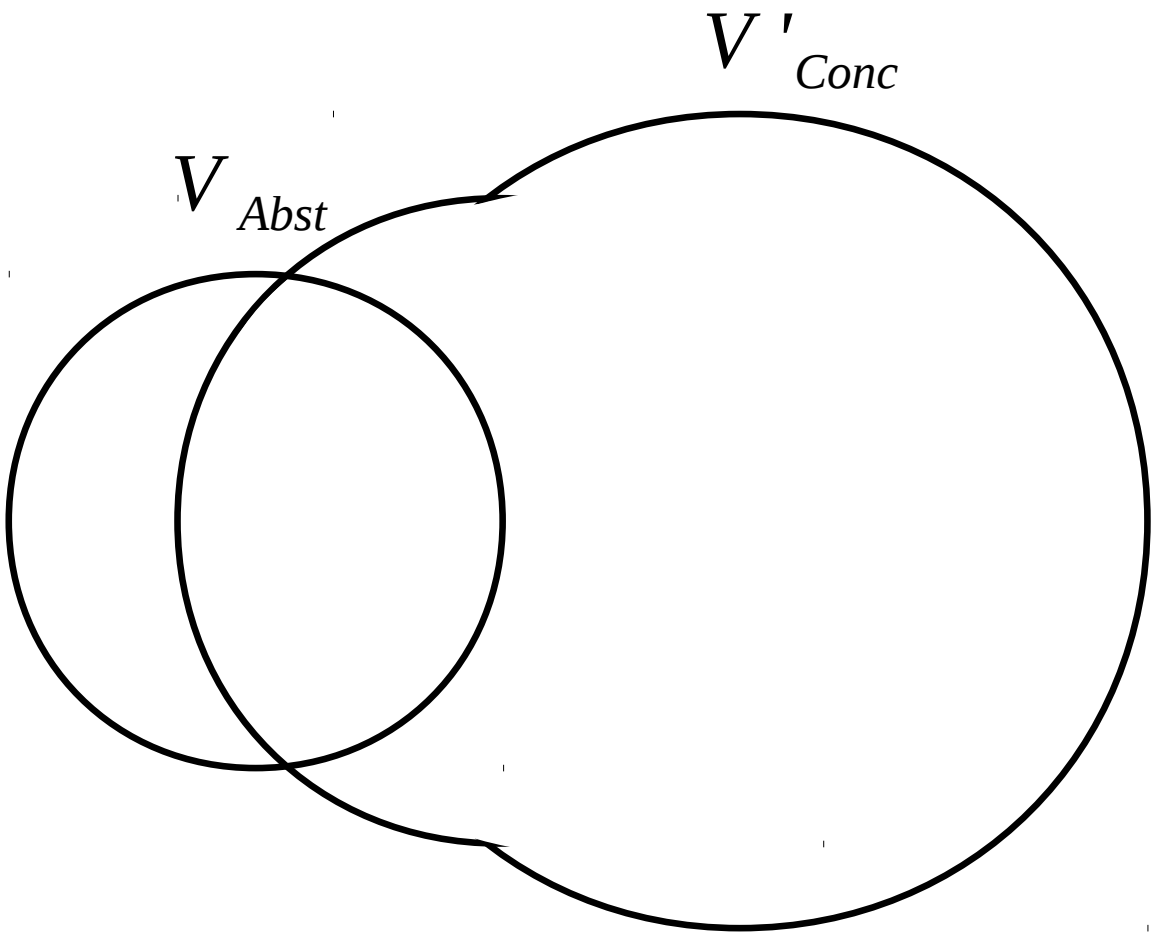
found Complement preds

original machines / intermediate machines	machine3	1st	2nd	3rd	4th
# variables	-2+10	-1+3	-1+3	+2	+2
# invariants	72	7	17	14	34
# complement preds		20	20	15	0
(# unique comp preds)		(8)	(9)	(8)	(0)

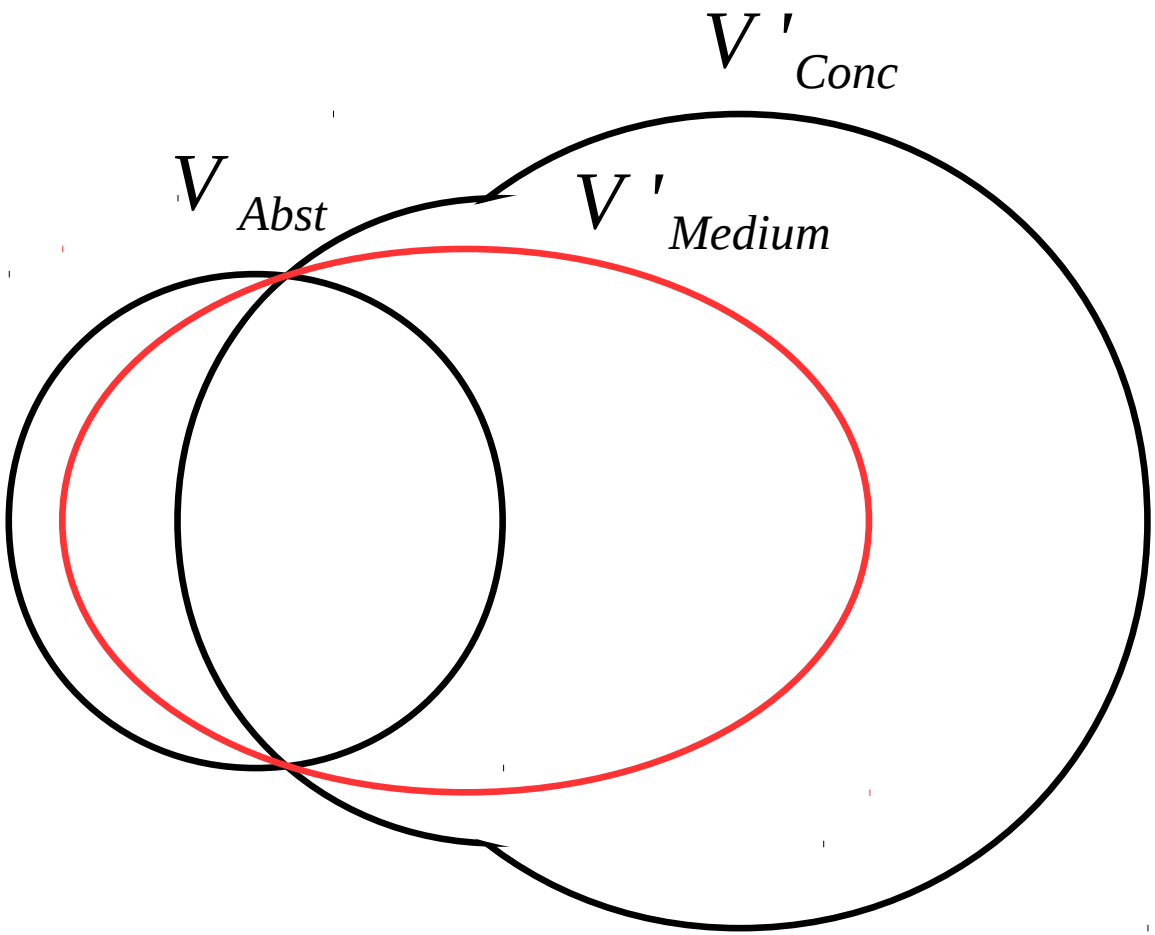
Extracting Parts of Machines for Reuse



Extracting Parts of Machines for Reuse



Extracting Parts of Machines for Reuse



Case Study: Extracting Authentication Parts of LAC

Original model: location access controller

- Persons move between locations connected with turnstiles
- Persons are authorized to enter certain locations
- Persons insert their ID card in card readers on turnstiles
- Turnstiles communicate with a controller via messages
  - authentication, movement of a person, indicator light, ...

New model: consoles, servers, and monitors

- Location have consoles with card readers and monitors.
- Authorized persons can login server by inserting ID card to the reader
- Controller try to find an unoccupied monitor in a room
- Consoles communicate with a controller via messages

Concluding Remarks

Summary

- Aiming at modify refinement structure of existing machines
- Slicing before mending / merging
- SliceAndMerge v1.0 coming soon?

Future work

- Constructing systematic methods for mending
- *Planning refinement* for various use cases
- Finding other use cases
  - Planting specifications in other methods (e.g., VDM)?
  - ...

Interpolation

Theorem (Craig interpolation)

If a sequent **Hyp** $\vdash$ **Goal** *is provable,*
there exists an interpolant I *such that*

- *All symbols of* I *occur in both* **Hyp** *and* **Goal**
- **Hyp** $\vdash I$ *and* $I \vdash$ **Goal** *are provable*

Example of interpolant

$TL_{Mainland} = red \lor TL_{Island} = red$
$TL_{Island} = green$

$n'_{\rightarrow} = n_{\rightarrow} + 1$ $\xleftarrow{interpolates}$ $TL_{Mainland} \neq green$
$\vdash$
$TL_{Mainland} = green \Rightarrow n'_{\rightarrow} = 0$

Complementary Predicates and Interpolation

Finding complementary predicates using interpolation

1. Original consistency sequent $\mathbf{Hyp} \vdash \mathbf{Goal}$ is given
2. Apply inference rules to get another sequent $\mathbf{Hyp}' \vdash \mathbf{Goal}'$ such that $\mathbf{Goal}'$ can be expressed using V_B
3. Obtain an interpolant I' of $\mathbf{Hyp}' \vdash \mathbf{Goal}'$
   - $\mathbf{Hyp}' \vdash I'$, $I' \vdash \mathbf{Goal}'$ // I' is an essence
   - I' can be expressed in V_B
4. Find complementary predicates from I'

Complementary Predicates and Interpolation

Finding complementary predicates using interpolation

1. Original consistency sequent $\mathbf{Hyp} \vdash \mathbf{Goal}$ is given
2. Apply inference rules to get another sequent $\mathbf{Hyp}' \vdash \mathbf{Goal}'$ such that $\mathbf{Goal}'$ can be expressed using V_B
3. Obtain an interpolant I' of $\mathbf{Hyp}' \vdash \mathbf{Goal}'$
   - $\mathbf{Hyp}' \vdash I'$, $I' \vdash \mathbf{Goal}'$ // I' is an essence
   - I' can be expressed in V_B
4. Find complementary predicates from I'

$$TL_{Mainland} = red \lor TL_{Island} = red$$
$$TL_{Island} = green$$
$$n'_{\rightarrow} = n_{\rightarrow} + 1$$
$$\vdash$$
$$TL_{Mainland} = green \Rightarrow n'_{\rightarrow} = 0$$

Complementary Predicates and Interpolation

Finding complementary predicates using interpolation

1. Original consistency sequent $\mathbf{Hyp} \vdash \mathbf{Goal}$ is given
2. Apply inference rules to get another sequent $\mathbf{Hyp}' \vdash \mathbf{Goal}'$ such that $\mathbf{Goal}'$ can be expressed using V_B
3. Obtain an interpolant I' of $\mathbf{Hyp}' \vdash \mathbf{Goal}'$
   - $\mathbf{Hyp}' \vdash I'$, $I' \vdash \mathbf{Goal}'$ // I' is an essence
   - I' can be expressed in V_B
4. Find complementary predicates from I'

$TL_{Mainland} = red \vee TL_{Island} = red$
$TL_{Island} = green$

$\vdash$
$TL_{Mainland} = green \Rightarrow n'_{\rightarrow} = 0$
$\neg n'_{\rightarrow} = n_{\rightarrow} + 1$

Complementary Predicates and Interpolation

Finding complementary predicates using interpolation

1. Original consistency sequent $\mathbf{Hyp} \vdash \mathbf{Goal}$ is given
2. Apply inference rules to get another sequent $\mathbf{Hyp}' \vdash \mathbf{Goal}'$ such that $\mathbf{Goal}'$ can be expressed using V_B
3. Obtain an interpolant I' of $\mathbf{Hyp}' \vdash \mathbf{Goal}'$
   - $\mathbf{Hyp}' \vdash I'$, $I' \vdash \mathbf{Goal}'$ // I' is an essence
   - I' can be expressed in V_B
4. Find complementary predicates from I'

$TL_{Mainland} = red \vee TL_{Island} = red$
$TL_{Island} = green$

$\vdash$ $\xleftarrow{interpolates}$ $TL_{Mainland} \neq green$

$TL_{Mainland} = green \Rightarrow n'_{\rightarrow} = 0$
$\neg n'_{\rightarrow} = n_{\rightarrow} + 1$