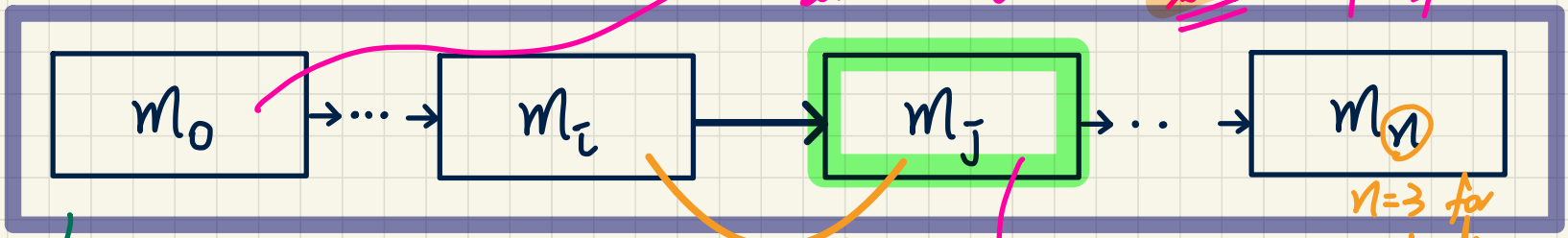
Lecture 2

Part A

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

Correct by Construction



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

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

PO of refinement:
1. m_j refines m_i (by intro. extra state variable and/or events)
m_j behaves consistently w.r.t. m_i.

$n=3$ for bridge controller

RD (requirements document)

- E-descriptions
- R-descriptions

State Space of a Model

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

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

typing, properties
actions, theorems

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

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

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

pos. train.

theorem proving

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

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

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

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

⇒ manipulation of symbols & predicate logic at the abstract level → combinatorial explosion (vs. concrete valuations for individual JUnit tests)

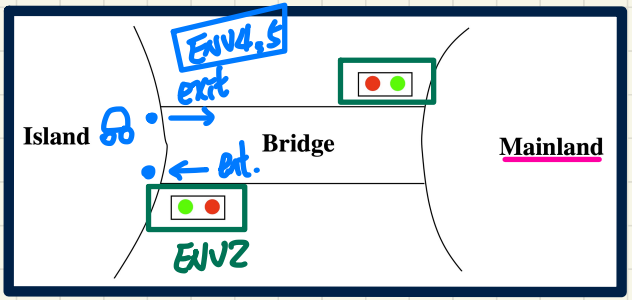
$|\mathbb{N}1| \times |\mathbb{N}1| \times |String \pfun \mathbb{Z}|$ values

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

Bridge Controller:

Requirements Document

ENV1	The system is equipped with two traffic lights with two colors: green and red.
ENV2	The traffic lights control the entrance to the bridge at both ends of it.
ENV3	Cars are not supposed to pass on a red traffic light, only on a green one.
ENV4	The system is equipped with four sensors with two states: on or off.
ENV5	The sensors are used to detect the presence of a car entering or leaving the bridge: "on" means that a car is willing to enter the bridge or to leave it.



E-descriptions (working environment)

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.

R-descriptions (functionalities, properties)

Lecture 2

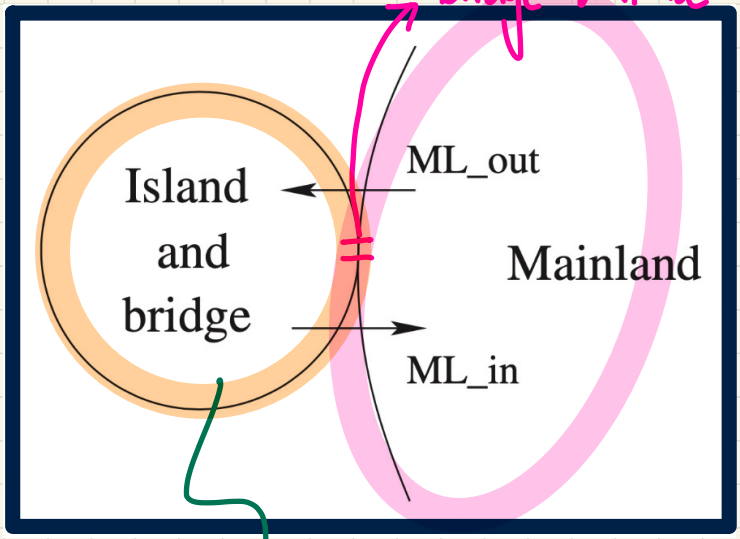
Part B

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

Bridge Controller: Abstraction in the Initial Model

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

bridge will be added at a later refinement.



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

Bridge Controller: State Space of the Initial Model

thm: theorem

REQ2	The number of cars n on bridge and island is limited. $n \leq d$

Static Part of Model

Context in Rodin

constants: d

axioms: (assumed to be true in initial model M_0)

axm0_1 : $d \in \mathbb{N}$ (1st axiom of M_0)

Dynamic Part of Model

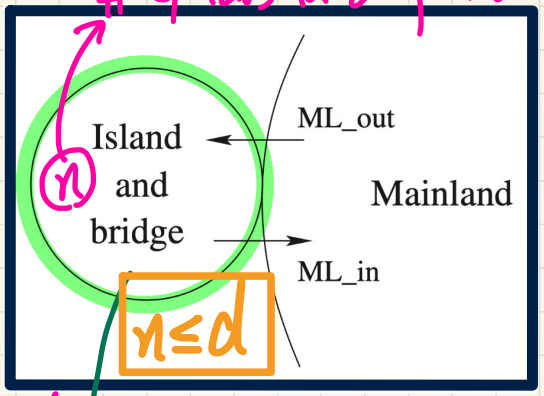
variables: n

invariants: I

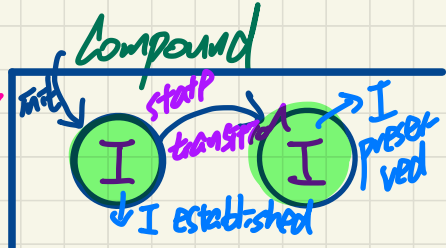
inv0_1 : $n \in \mathbb{N}$ (typing constraint)

inv0_2 : $n \leq d$ (property)

$I \triangleq n \in \mathbb{N} \wedge n \leq d$ in M_0



Interactions between system and users continue "forever"



Bridge Controller: State Transitions of the Initial Model

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

abstraction w.r.t. REQ2.

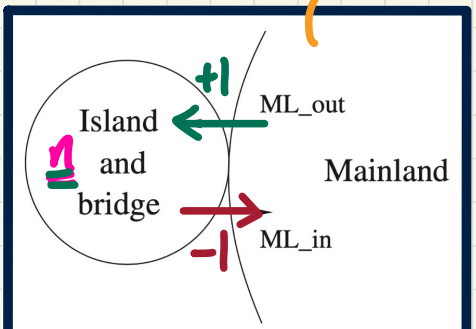
state space

constants: d

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

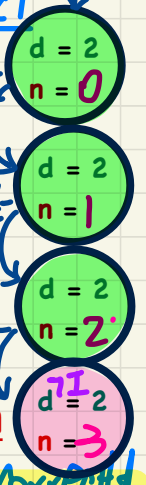
variables: n

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



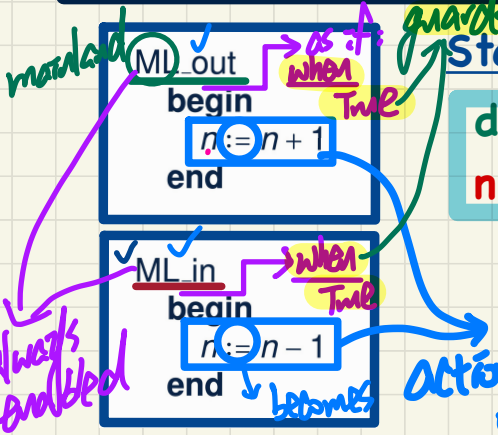
Trace: ⟨ML_out, ML_out, ML_out⟩

Ex1 fail



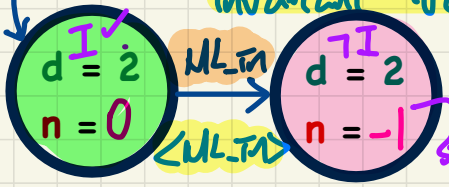
State Transition Diagram on an Example Configuration

guards → events enabled if they evaluate to true.

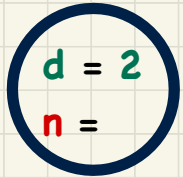


d = 2
n initialized to 0

Ex1 fail

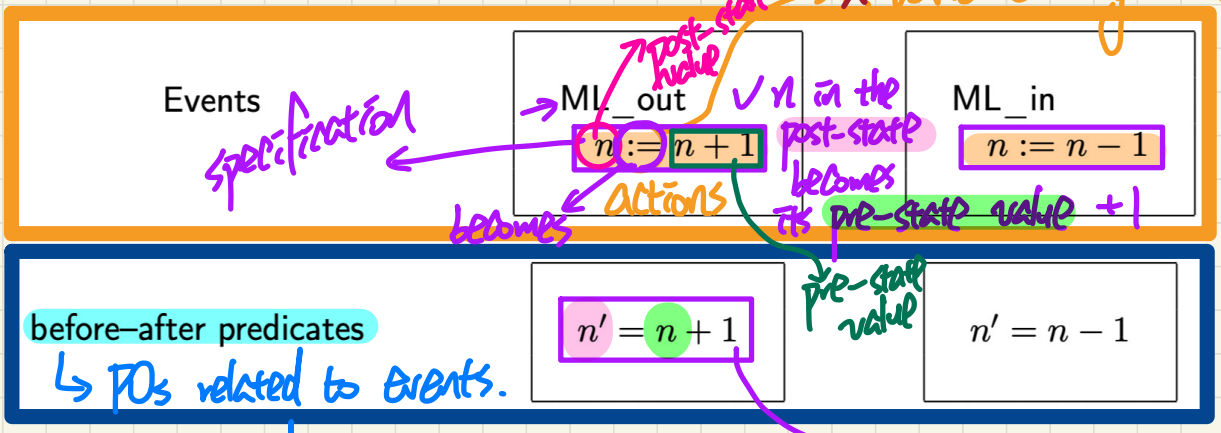


Are ML_in and ML_out specified correctly s.t. there's not a trace leading to invariant violation.



Before-After Predicates of Event Actions

- Pre-State
- Post-State
- State Transition



post-state → x' variable assignment

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

pre-state value

↳ POs related to events.

e's guard evaluates to true

event e occurs

Pre-State $\in$ State Space

Before-State

state changed from pre-state according to e's action

Post-State $\in$ State Space

After-State

BAP

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

post-state value $n' = n + 1$ pre-state

Variable x:

Unprimed version x denotes its value in pre-state.

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