Lecture 9 - February 7

Reactive System: Bridge Controller

Announcements

- Lab2 released
- WrittenTest1 coming

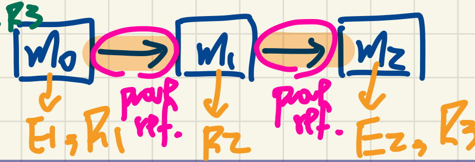
Lecture

Reactive System: Bridge Controller

Correct by Construction
State Space
Req. Doc.

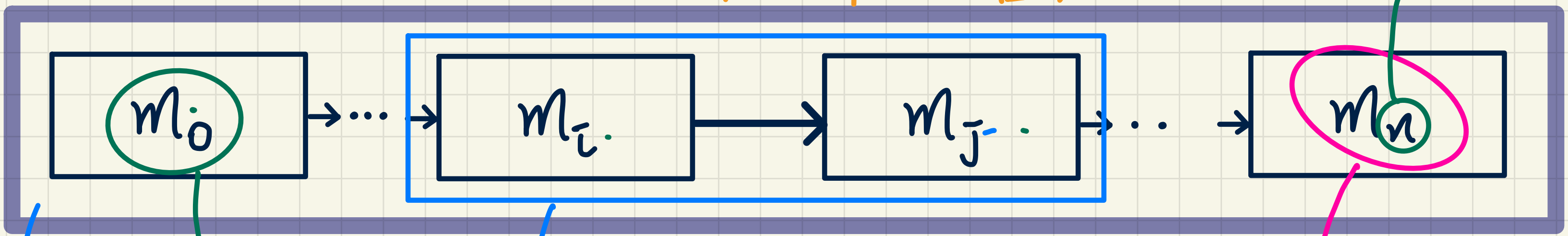
Correct by Construction

RD: E1, E2, R1 ~ R3



bridge controller

n=3



models: descriptions of SUD by filtering out irrelevant details

most abstract

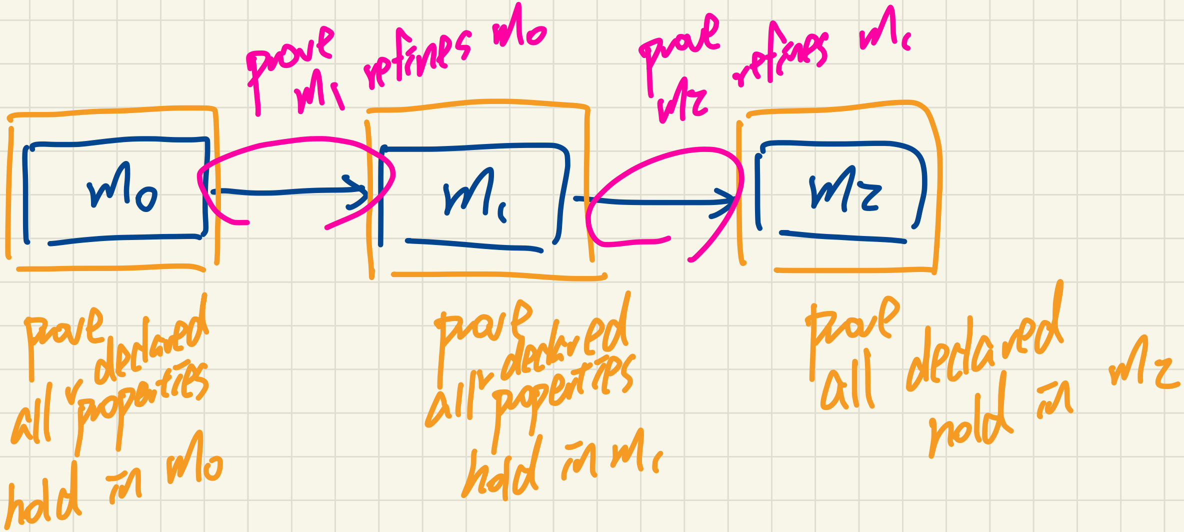
m_j refines m_i

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



Is it necessary to <u>also</u> prove m2 refines m0?

↳ No. Refinement relations are transitive.

State Space of a Model

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

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:

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

should not be even considered! does not satisfy axiom

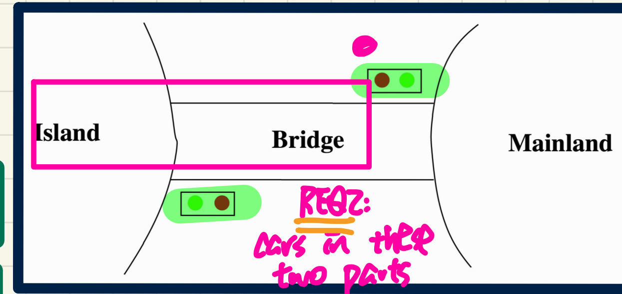
(C1) (C = 100, L = 300, accounts {("alan", 150), ("mark", 199)}) {C=100, L=200, accounts = {"alan" ↦ -300}}

satisfies the axiom, that is no state can violate 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.
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.



REQ2: cars on these two parts should be limited!

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

without this, verification results would be unrealistic.

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

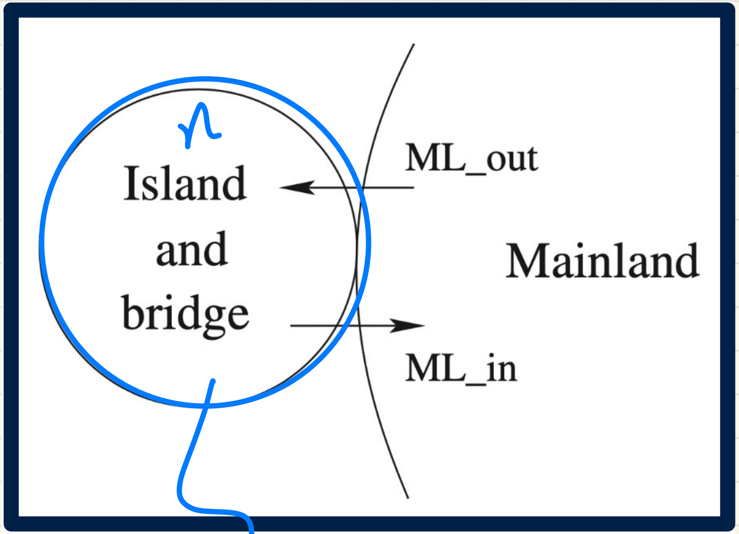
Lecture

Reactive System: Bridge Controller

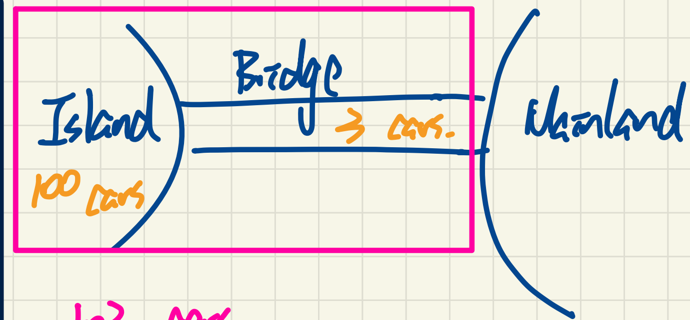
Initial Model: State and Events

Bridge Controller: Abstraction in the Initial Model

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



the notion of bridge is abstracted away.



103 cars on the Island-Bridge compound

Bridge Controller: State Space of the Initial Model

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

Static Part of Model

constants: d

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

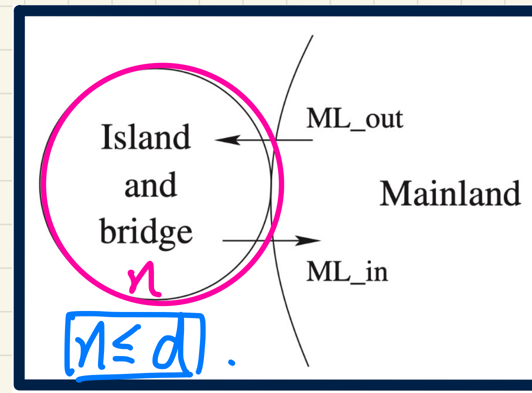
current # of cars in island and bridge

invariants:

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

inv0_2 : $n \leq d$

REQ2



Bridge Controller: State Transitions of the Initial Model

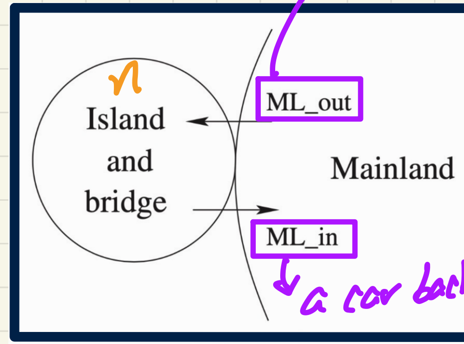
REQ2	The number of cars on bridge and island is limited.

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

ML_in
  begin
    $n := n - 1$
  end

actions

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 =