Lecture 10 - February 9

Reactive System: Bridge Controller

Announcements

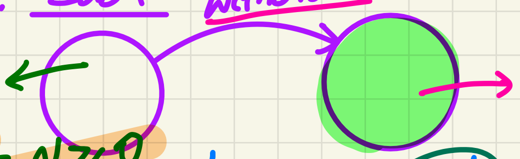
- Lab2 released
- WrittenTest1 guide released
  + Verify EECS account on a WSC machine
  + Verify PPY account and Duo Mobile on eClass

Invariants (I)

= Conditions that must hold true all the time

1. accounts balance is ≥ -C
2. v ≥ 0

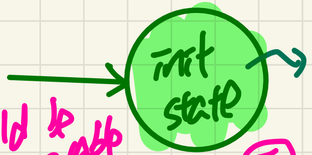
e.g. Lab1



action: decrement balance by v.

→ account's balance remains ≥ -C

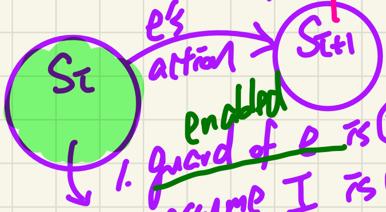
(1) I established after initializing the system.



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.

it should be provable that I remains (T)



1. ~~guard of e~~ enabled is (T)
2. assume I is (T)

Bridge Controller: State Transitions of the Initial Model

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

context

constants: d

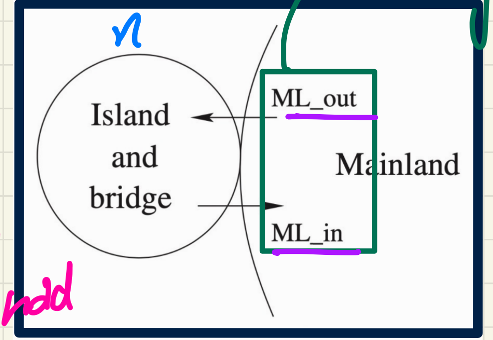
axioms:
axm0_1 : $d \in \mathbb{N}$

machine.

variables: n

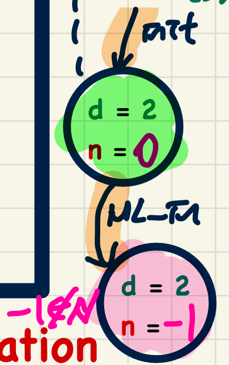
invariants: I
inv0_1 : $n \in \mathbb{N}$
inv0_2 : $n \leq d$

should always hold



occurrences of events change values of variable(s)

<init, ML_in>



enabled

ML_out when True
begin
$n := n + 1$
end

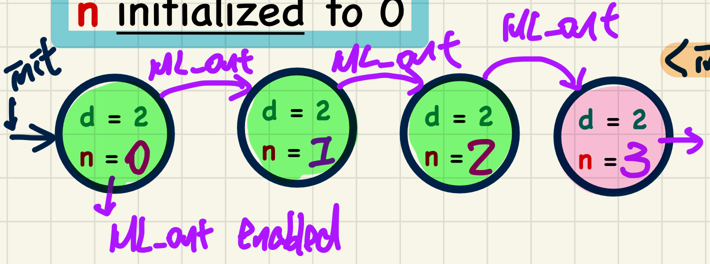
ML_in when True
begin
$n := n - 1$
end

State Transition Diagram on an Example Configuration

Is ML_out going to maintain I always?

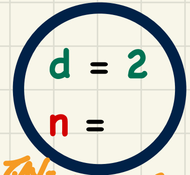
d = 2

n initialized to 0



$3 \in \mathbb{N}$ ✓
$3 \leq 2$ ✗
($\neg I$)

<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

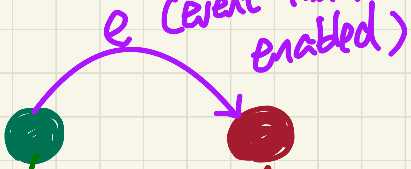
actions

after ML_out's action takes effect, the

post-state value

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

- Pre-State
- Post-State
- Sate Transition



pre-state
before-state

post-state
after-state → variables are "primed"

e.g. ML_out

$n=2$ $n'=3$ → $n' = n + 1$ before-after predicate characterising the effect of event.

event actions

$v := v + 1$

1. becomes
2. not variable assignment!!

swap $x, y, temp$

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

X

∵ := not variable assign.

Just:

$x := y$

$y := x$

BAP:

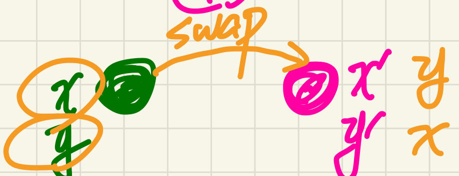
$x' = y$

$y' = x$

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

cannot have the same variable as LHS multiple times!

$$x' = x + 1 \wedge x' = x - 1 \equiv F.$$



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** : $n \leq d$

$\Big]\ I$

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

$$\downarrow$$

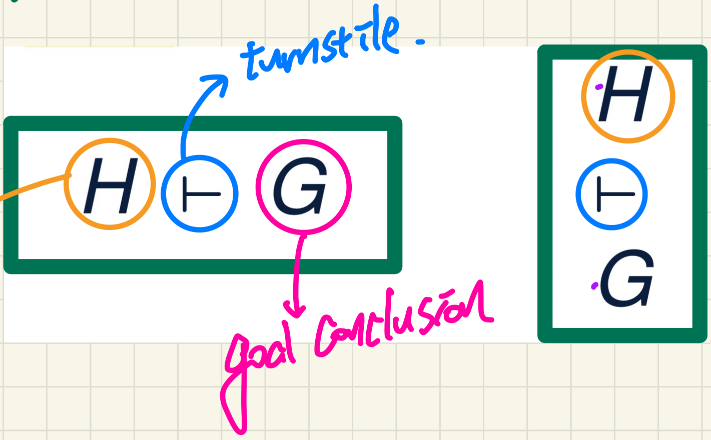
$$\neg \exists\, state \cdot state \in StateSpace \land \boxed{\neg I(state)}$$

witness of violation

Sequents: Syntax and Semantics

Both H and G are sets of predicates.

Syntax



assumptions/hypotheses

Semantics

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

T or F

T → G is provable, given H

F → G is not provable given H

↳ assuming H, G should be provable.

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

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

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

$\equiv \text{False} \Rightarrow G \equiv \text{True.}$ ✗ 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
  begin
    n := n - 1
  end
```

True

BAP: $n' = n-1$

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

ML_in

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

assumed to be true

should be provable

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

ML_out