# COL:750

## Foundations of Automatic Verification

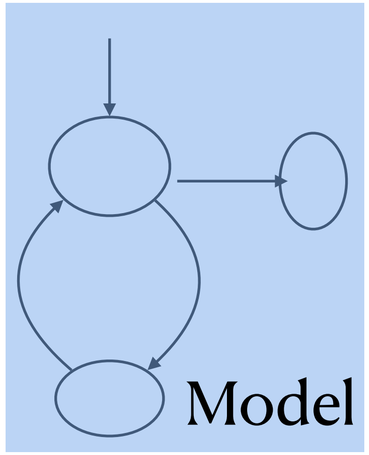
### Instructor: Priyanka Golia

Course Webpage 

https://priyanka-golia.github.io/teaching/COL-750/index.html

Acknowledgment: content of today's lecture was inspired/taken from Prof. B. Srivathsan lectures

Does Code Satisfy Requirements ?

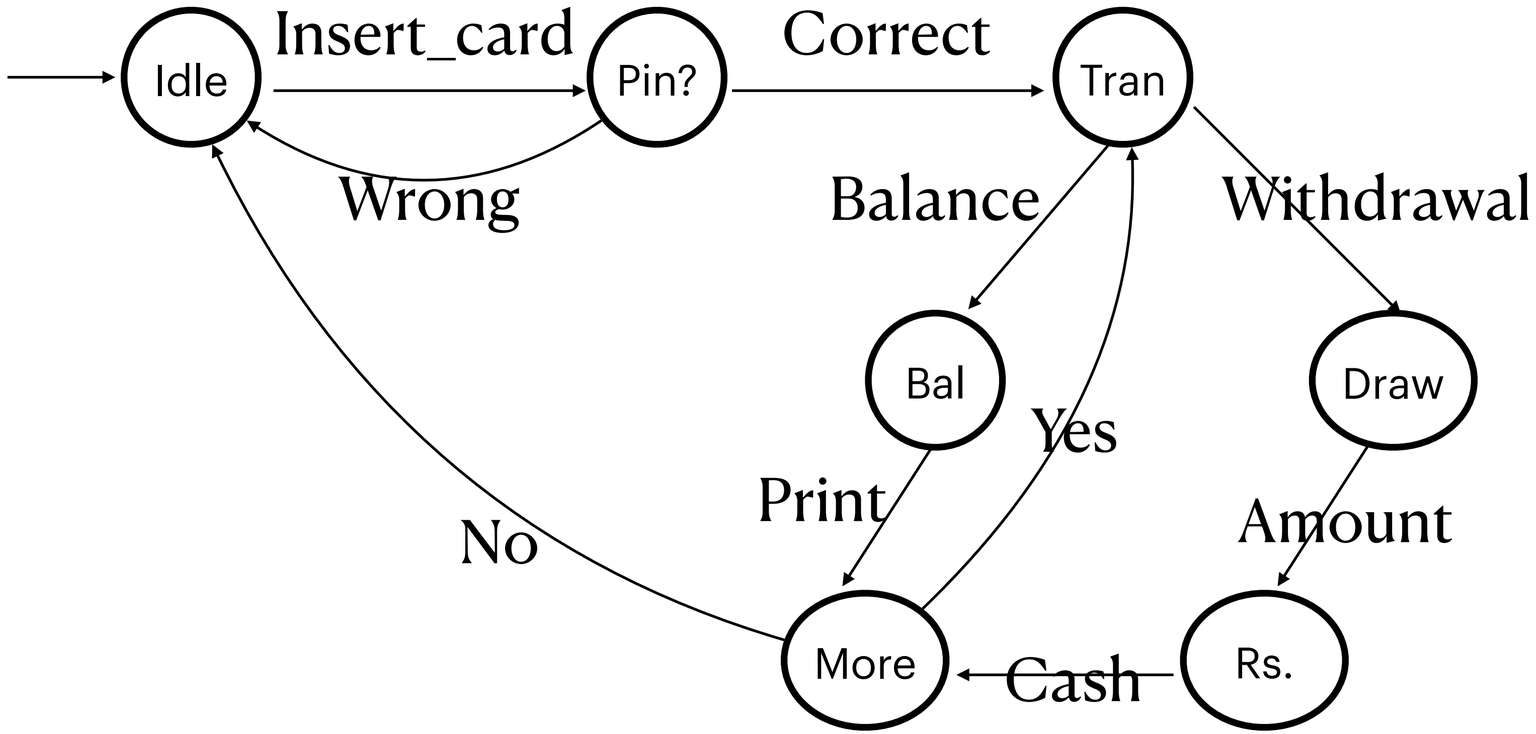


Does Model Satisfy Logical formulation: LTL/CTL Formula ?

Model Checking

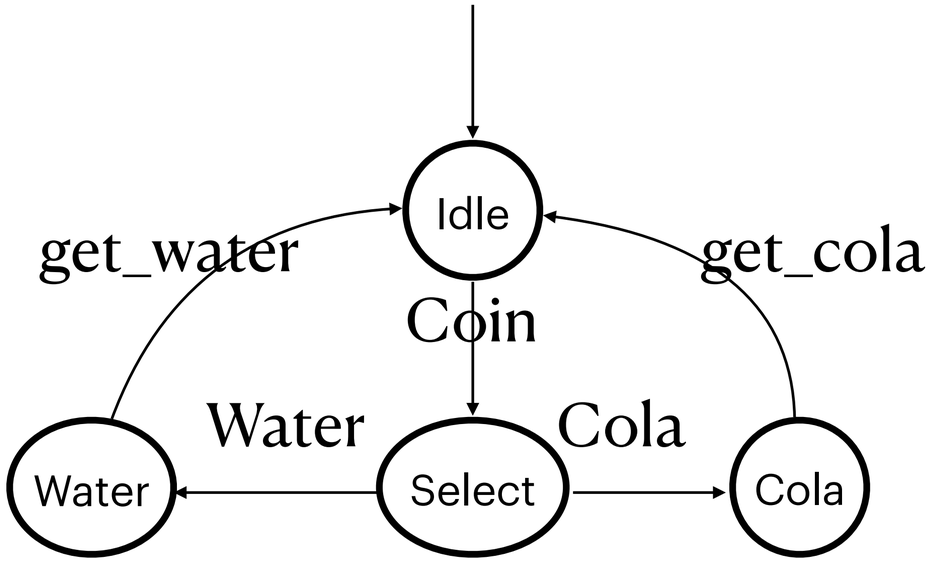
# Modeling code behavior

How to model ATM behavior?



# Modeling code behavior

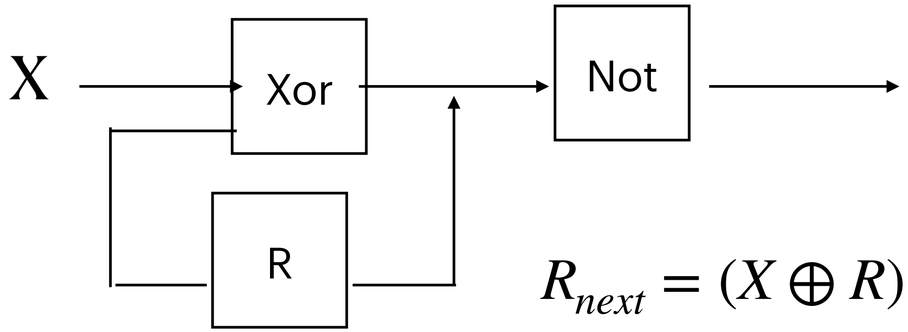
How to model a vending machine behavior?

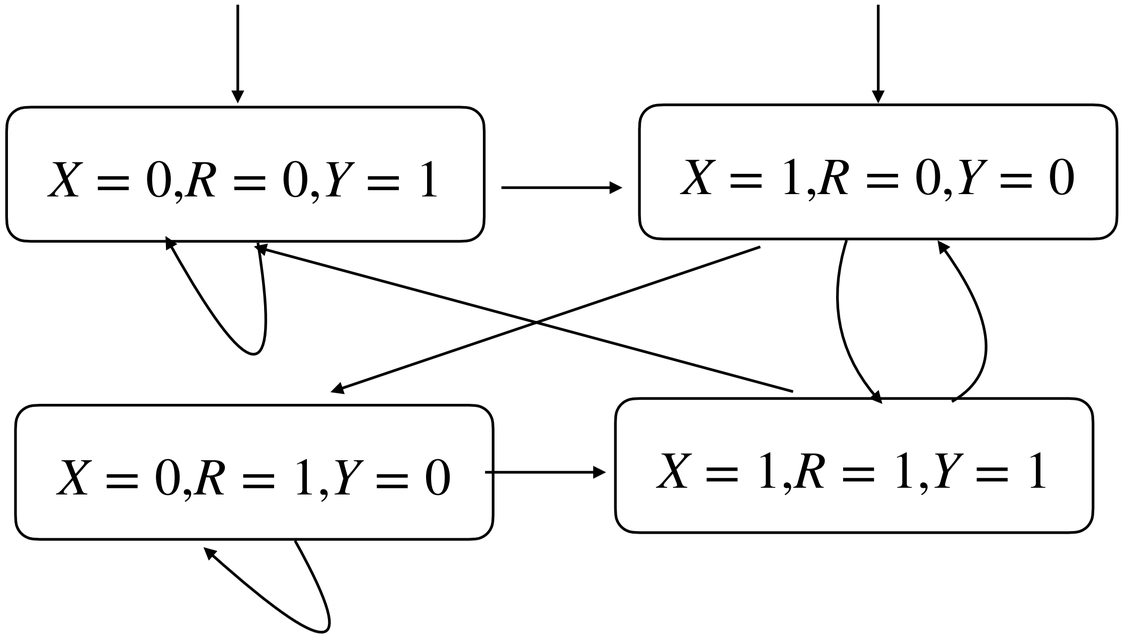



Transition systems:  states, actions, transitions

# Modeling code behavior

Hardware $Y = \neg(X \oplus R)$

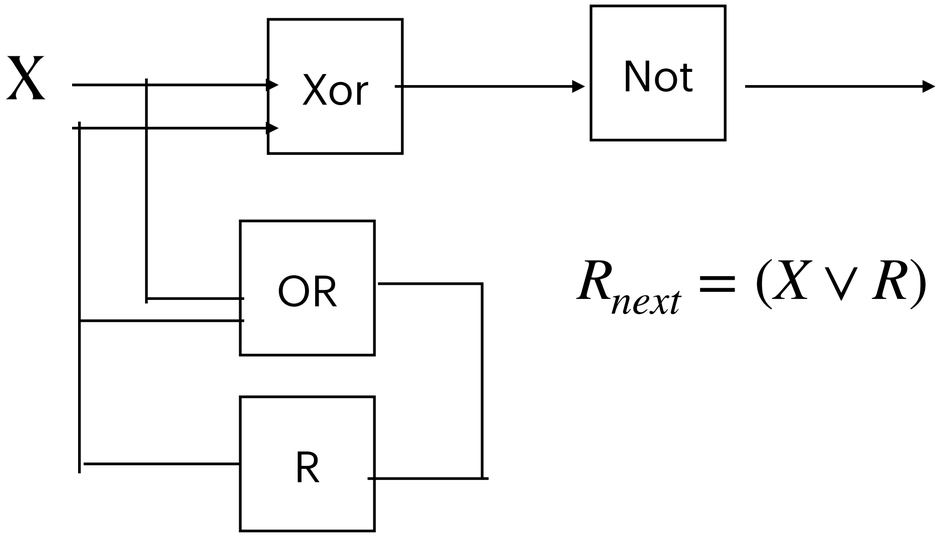



| | | | | | | |
|---|---|---|---|---|---|---|
| X | 1 | 1 | 0 | 1 | 1 | 1 |
| R | 0 | 1 | 0 | 0 | 1 | 0 |
| Y | 0 | 1 | 1 | 0 | 1 | 0 |

# Modeling code behavior

Hardware $Y = \neg(X \oplus R)$




$R_{next} = (X \vee R)$

Register

| | | | | | | |
|---|---|---|---|---|---|---|
| X | 1 | 1 | 0 | 1 | 0 | 1 |
| R | 0 | 1 | 1 | 1 | 1 | 1 |
| Y | 0 | 1 | 0 | 1 | 0 | 1 |

$X = 0, R = 0, Y = 1$

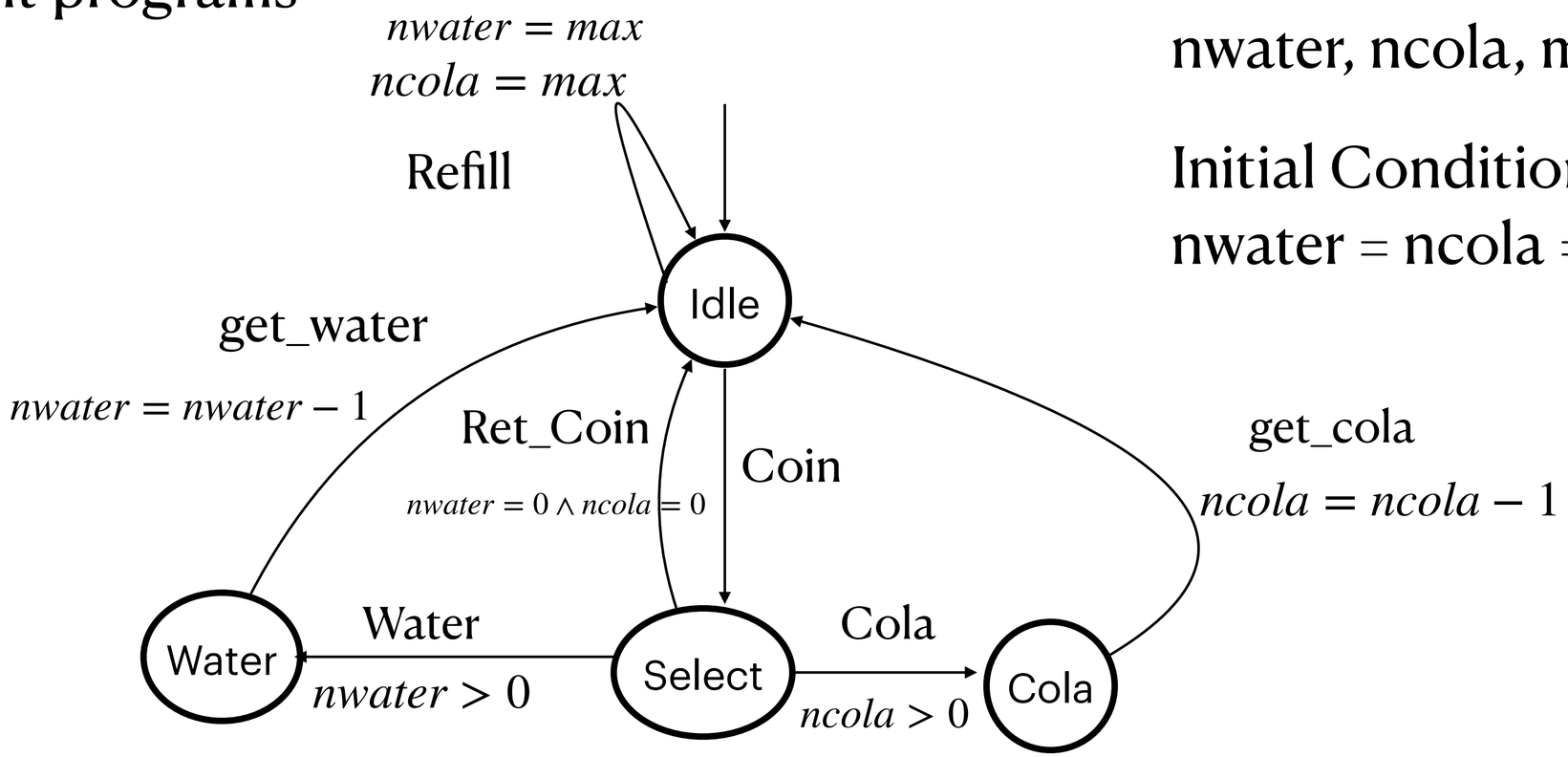
$X = 1, R = 0, Y = 0$

$X = 0, R = 1, Y = 0$

$X = 1, R = 1, Y = 1$

# Modeling code behavior

Data-dependent programs



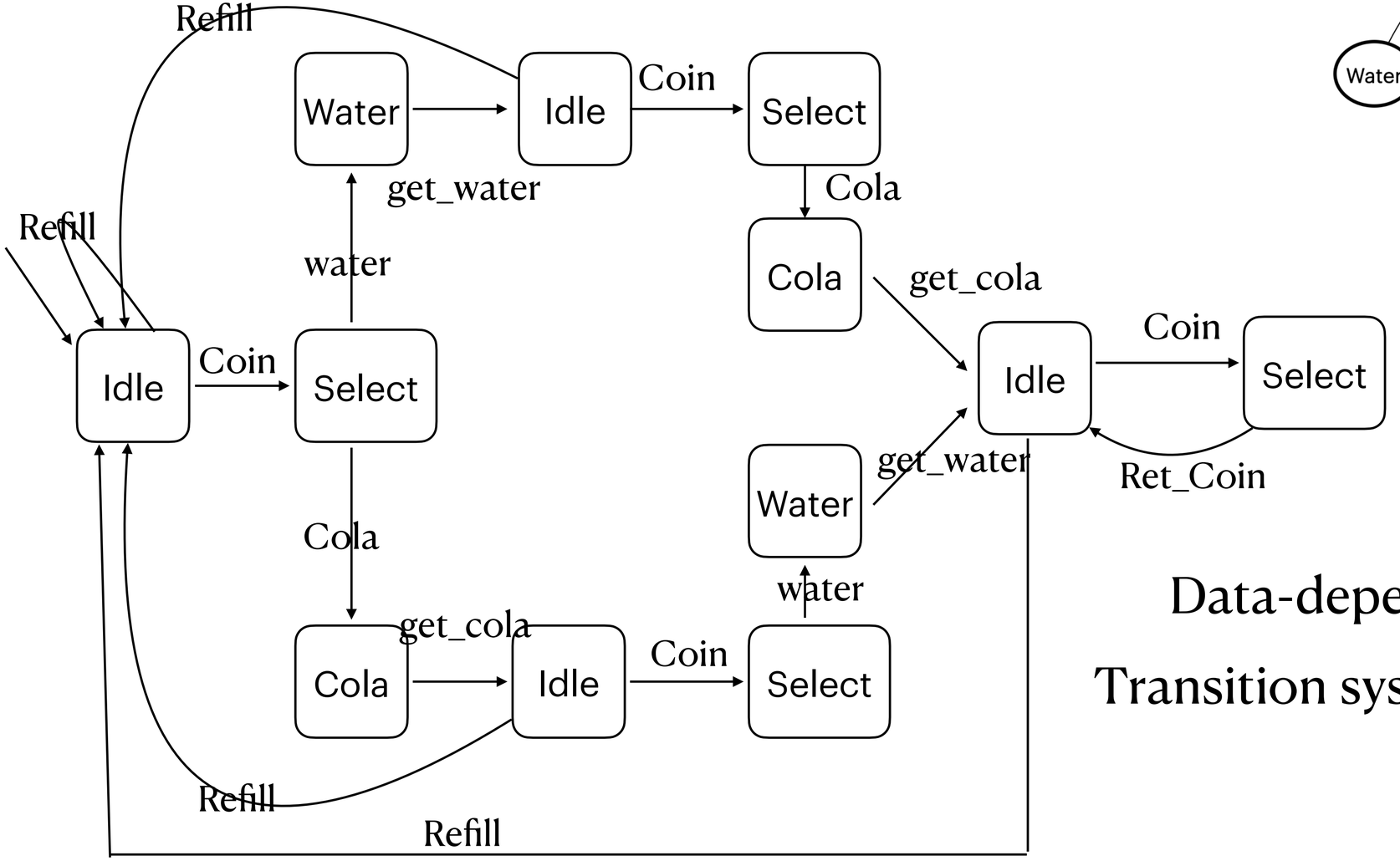

Program Graph

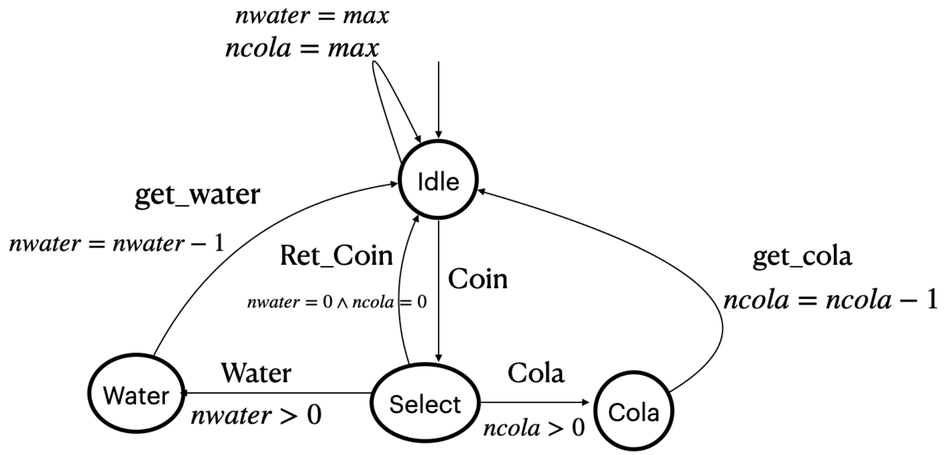
Variables:
nwater, ncola, max

Initial Condition:
nwater = ncola = max

# Modeling code behavior

Data-dependent programs




Program Graph

Data-dependent programs

Transition system corresponding to Max=1

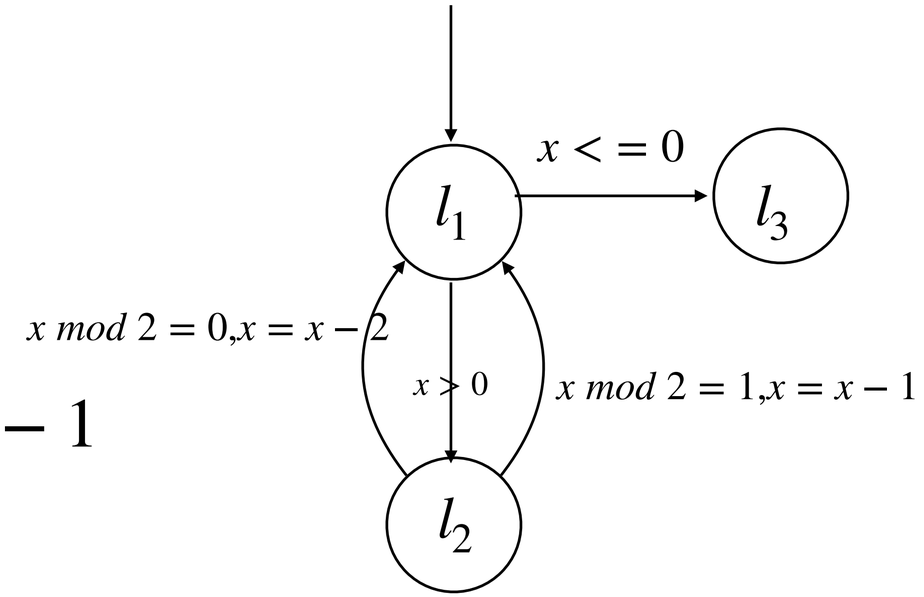
# Modeling code behavior

Data-dependent programs

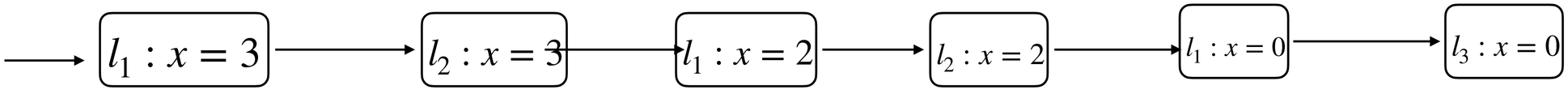
$l_1$: while $(x > 0)$

$l_2$ : $If(x \ mod \ 2 == 0)$ then $x = x - 2$, else $x = x - 1$

$l_3$ : …

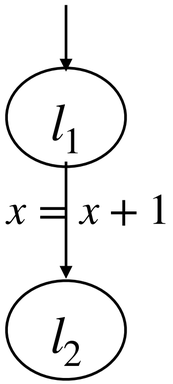



Transition system with initial condition $x = 3$



# Modeling Concurrent Systems

Independent

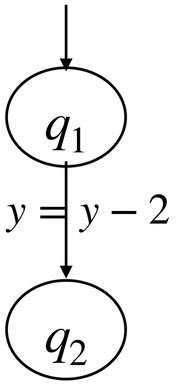



..
X: X+1
...

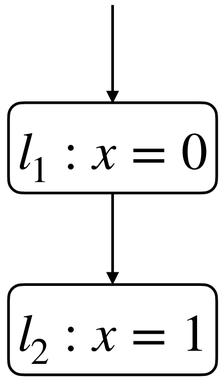
Shared variables

..
Y: Y-2
...

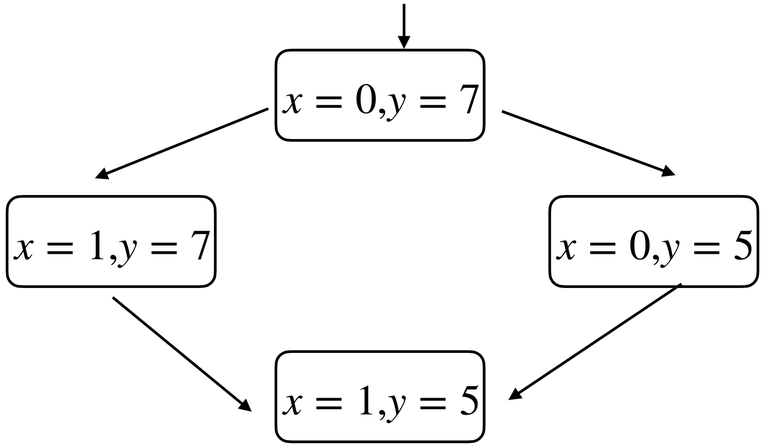
Shared actions



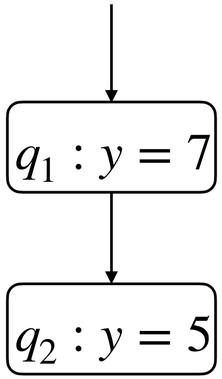

What is the transition system for the joint behaviour?




Initially x = 0

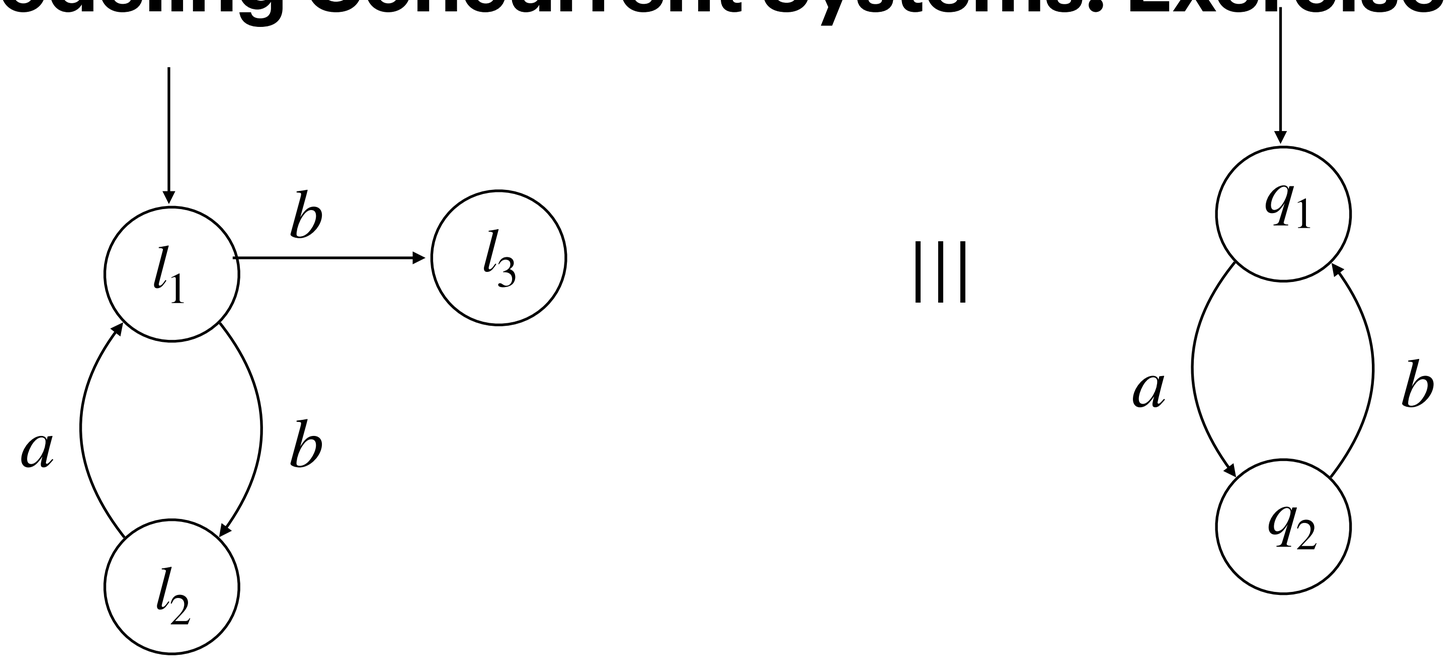



$TS_1 \;|||\; TS_2$




Initially y = 5

# Modeling Concurrent Systems: Exercise



# Modeling Concurrent Systems

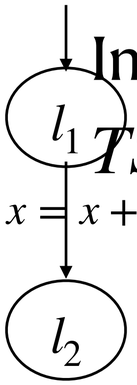
Independent

Shared variables

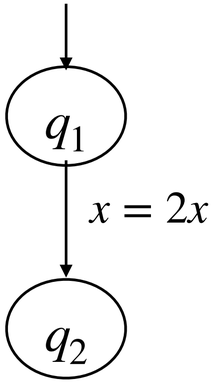
Shared actions

Interleaving:

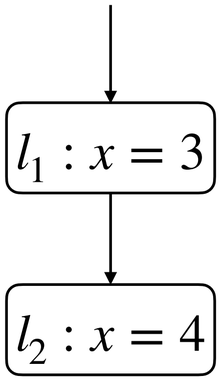
$TS_1 \;|||\; TS_2 \;|||\; T_3 \ldots$



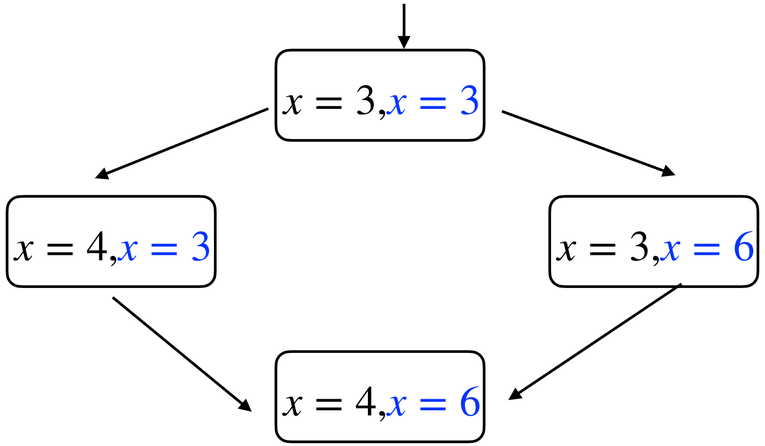


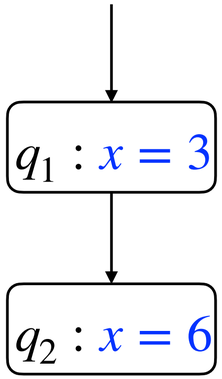

What is the transition system for the joint behaviour?




Initially x =3



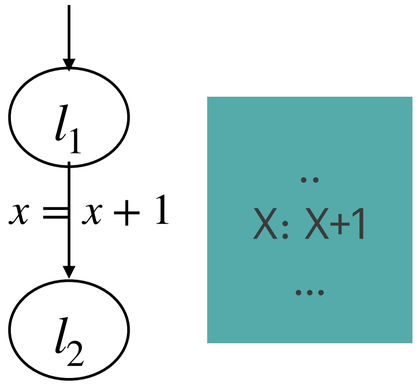

$TS_1 \;|||\; TS_2$




Initially x = 3

# Modeling Concurrent Systems

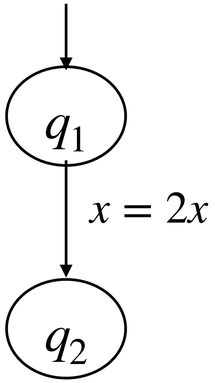
Shared variables

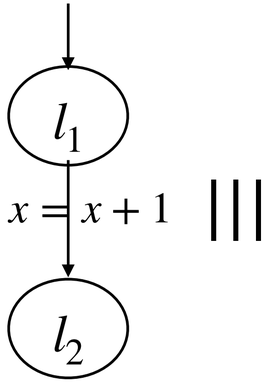




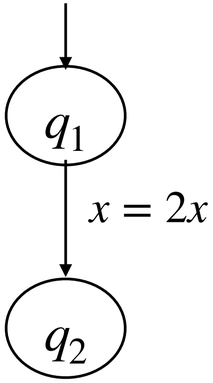


Program Graph Interleaving

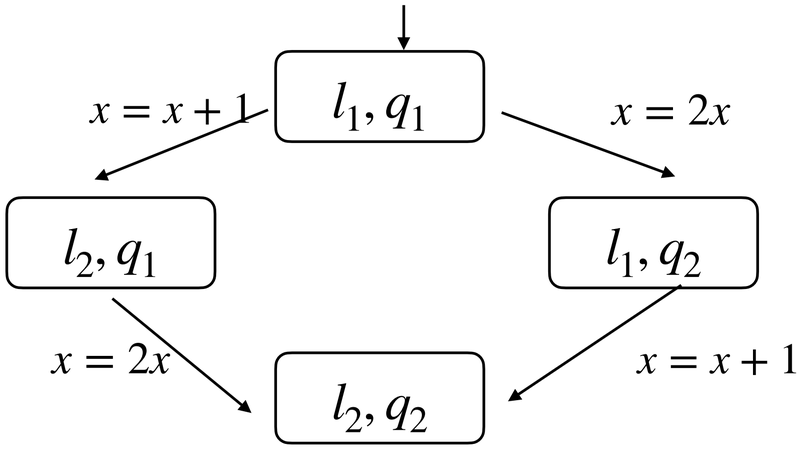



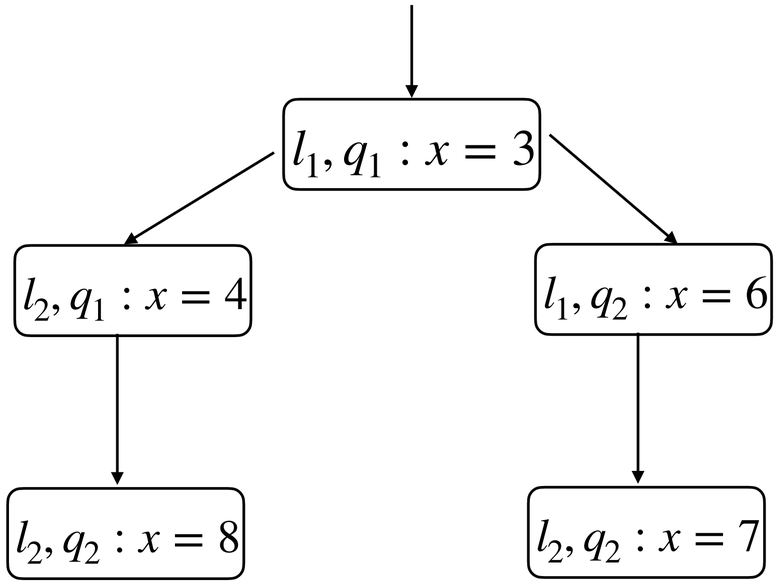

|||




=




$PG_1 \,|||\, PG_2$



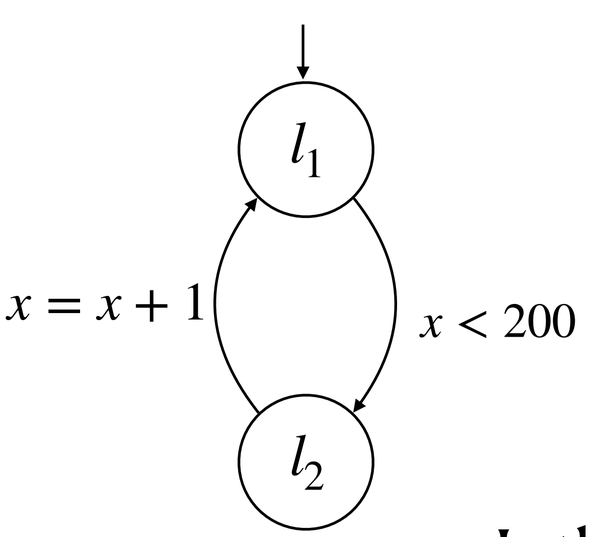

$TS(PG_1 \,|||\, PG_2)$

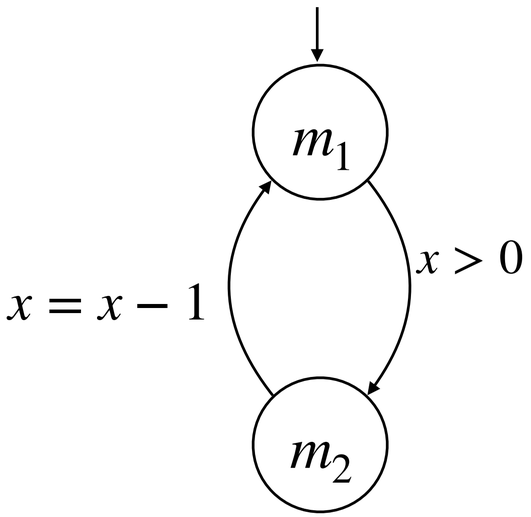
# Modeling Concurrent Systems

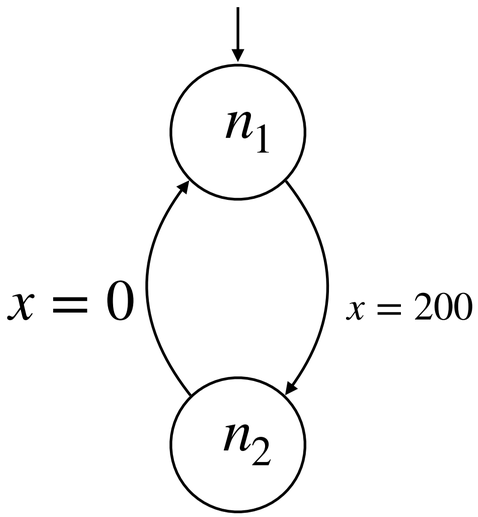
While (X <200)
{x = x+1}

While (X >0)
{x = x-1}

While (X = 200)
{x = 0}

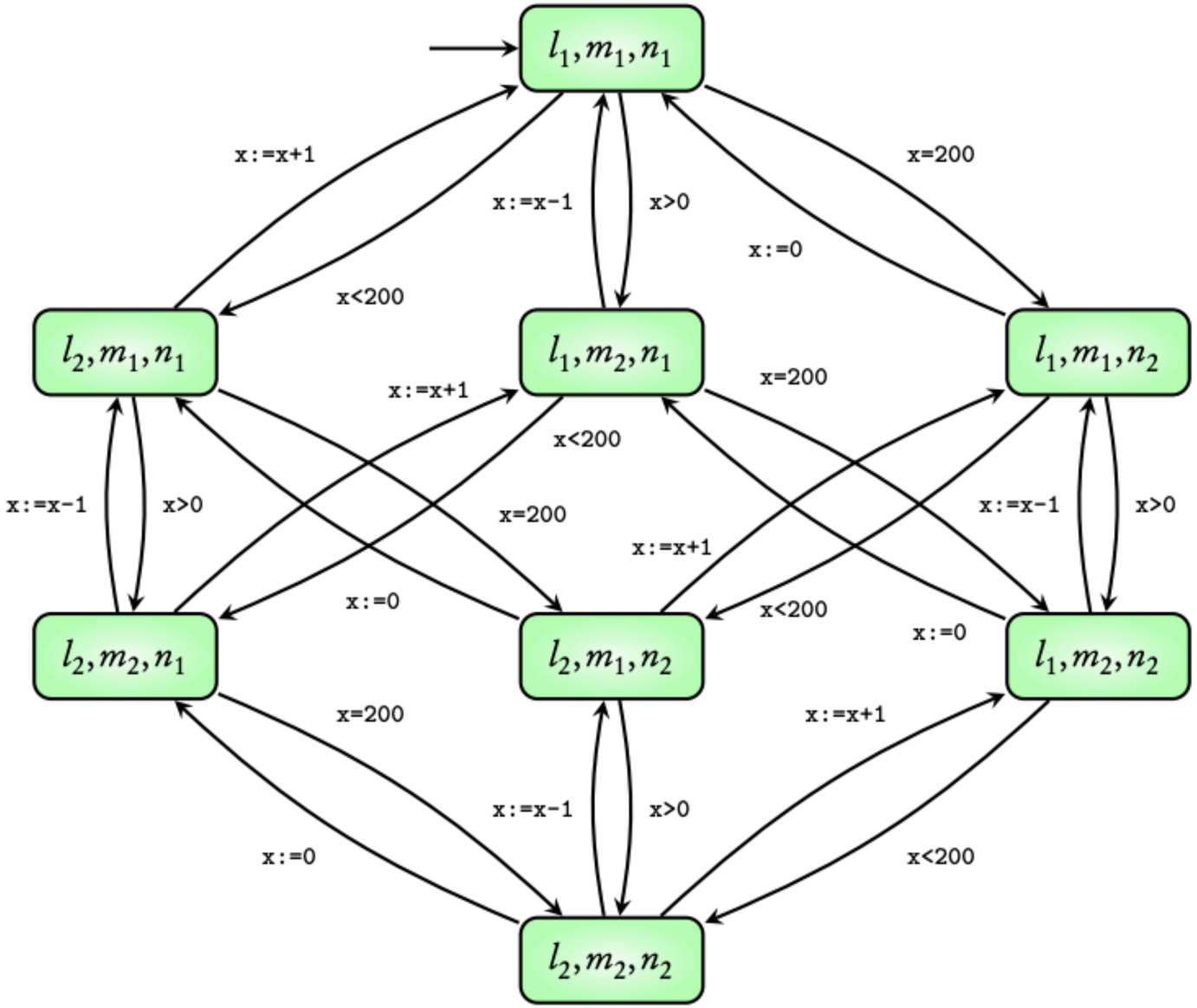


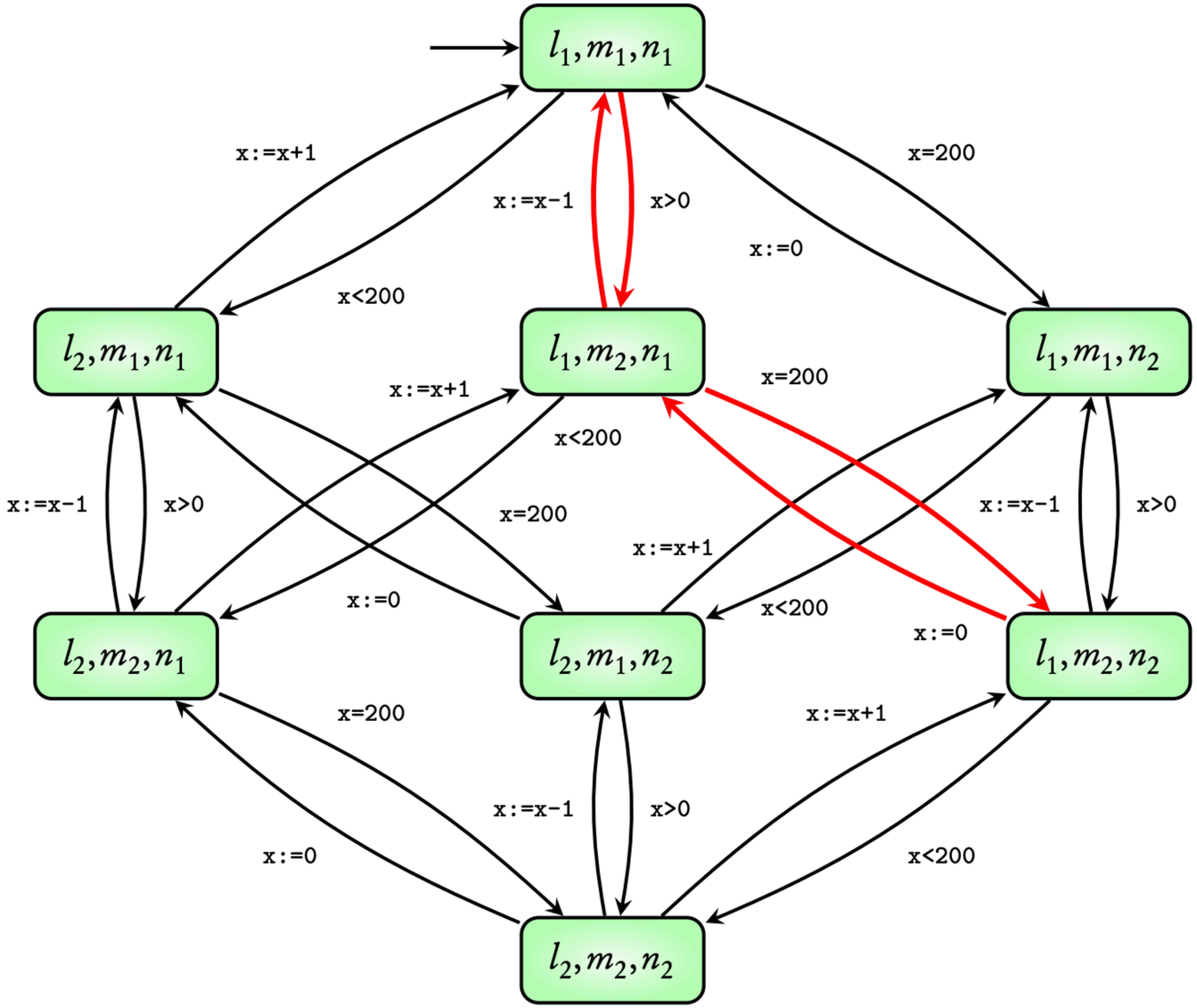




Is the value of x always between 0 and 200?

# Modeling Concurrent Systems



# Modeling Concurrent Systems




Is the value of x always between 0 and 200? No!!!

# Modeling Concurrent Systems

Shared variables — Mutual exclusion

Mutual Exclusion: No two processes can access the resource (variables, printers, ..) simultaneously

How do we model the protocol for mutual exclusion?

Exercise!!

# Modeling Concurrent Systems

Independent: Book-keeping system in a supermarket

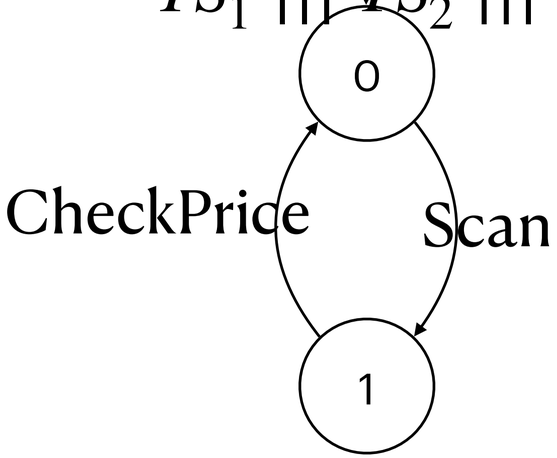
Shared variables

Shared actions

Interleaving:

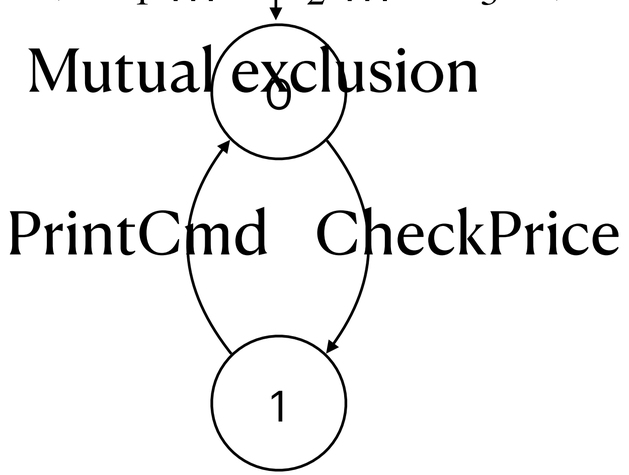
$TS_1 \mid\mid\mid TS_2 \mid\mid\mid T_3 \ldots$

$TS(PG_1 \mid\mid\mid PG_2 \mid\mid\mid PG_3 \ldots)$

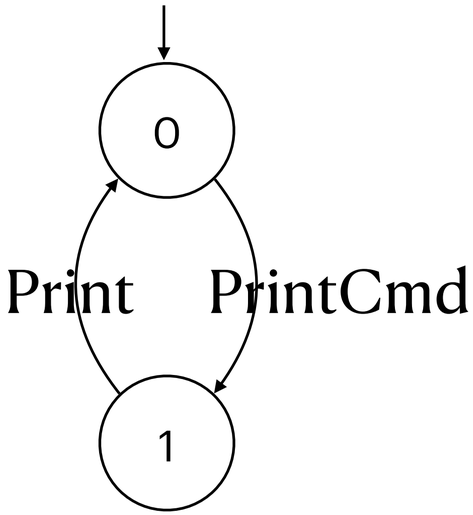



Bar Code Reading (BCR)

Mutual exclusion




Booking Program (BP)

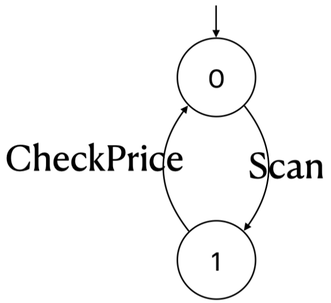


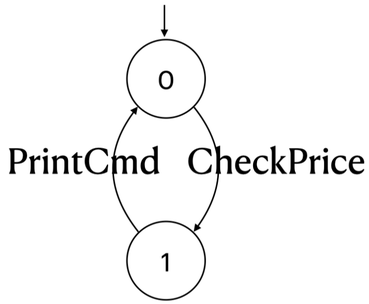

Printer

check_price, print_cmd: Shared actions (also called handshaking actions)

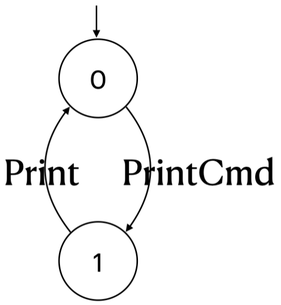
# Modeling Concurrent Systems

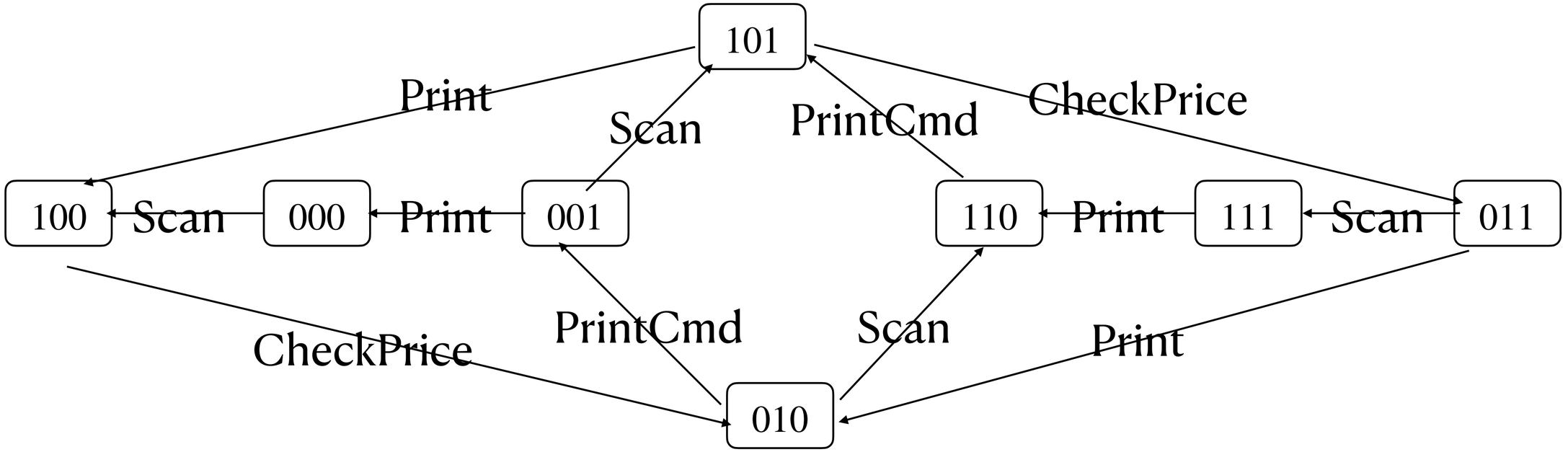
Book-keeping system in a supermarket




Shared actions



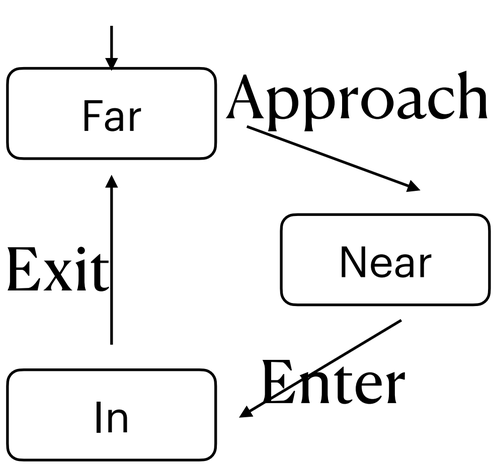




$$BPR \;||\; BP \;||\; P$$

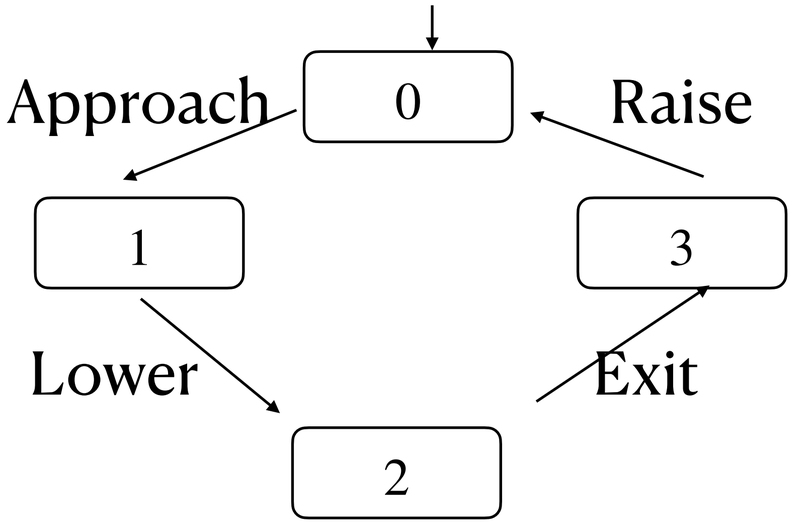
# Modeling Concurrent Systems

Shared actions

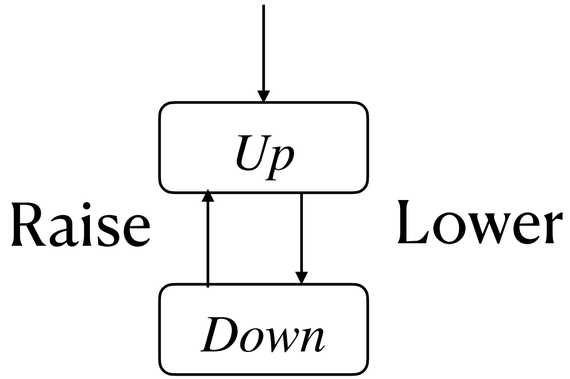
Train Crossing: Automatic Gate Closing




Train




Controller




Gate

# Modeling Concurrent Systems

Independent

Interleaving:

$TS_1 \;|||\; TS_2 \;|||\; T_3 \ldots$

Shared variables

Mutual exclusion

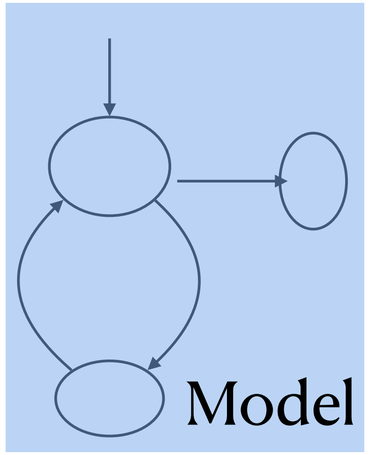
$TS(PG_1 \;|||\; PG_2 \;|||\; PG_3 \ldots)$

Shared actions

Handshake:

$TS_1 \;||\; TS_2 \;||\; T_3 \ldots$

Does Code Satisfy Requirements ?



Does Model Satisfy Logical formulation: LTL/CTL Formula ?

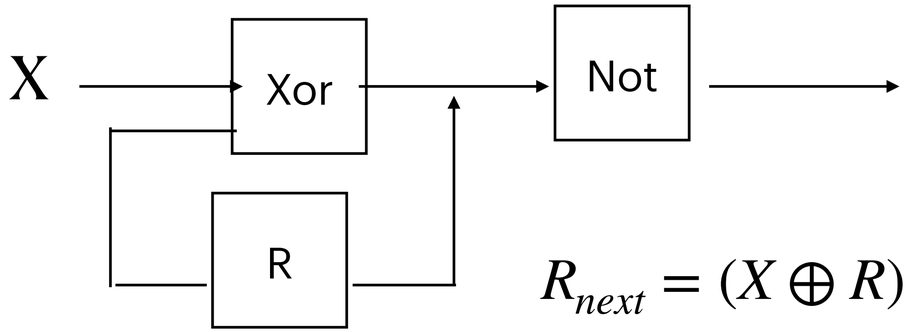
Model Checking

Model-checker will automatically check if system satisfies requirements

**NuSMV** : New Symbolic Model Verifier https://nusmv.fbk.eu/

**NuSMV** : New Symbolic Model Verifier https://nusmv.fbk.eu/

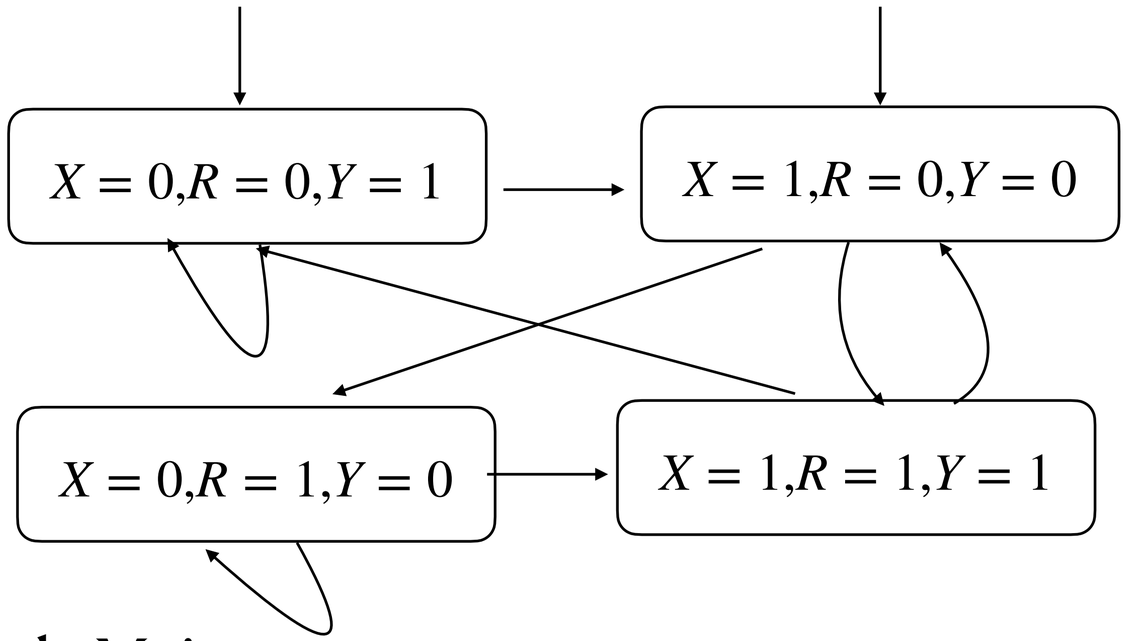
Hardware $Y = \neg(X \oplus R)$




$R_{next} = (X \oplus R)$

Register

| | | | | | | |
|---|---|---|---|---|---|---|
| X | 1 | 1 | 0 | 1 | 1 | 1 |
| R | 0 | 1 | 0 | 0 | 1 | 0 |
| Y | 0 | 1 | 1 | 0 | 1 | 0 |




Module Main

Var

X: Boolean, R: Boolean

Assign

Init(R):= False, Next(R):= x xor r