Week-1: Introduction to model checking

B. Srivathsan

Chennai Mathematical Institute

NPTEL-course

July - November 2015

Module 4:

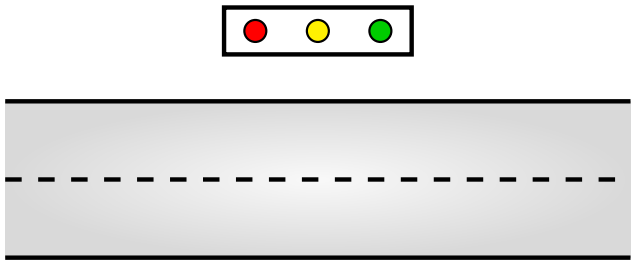
Modeling concurrent systems

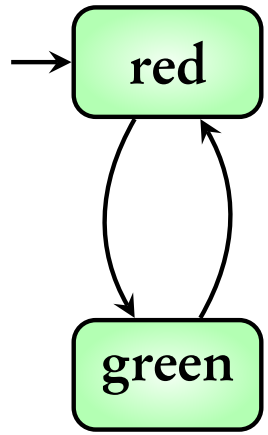
Concurrent systems

Independent

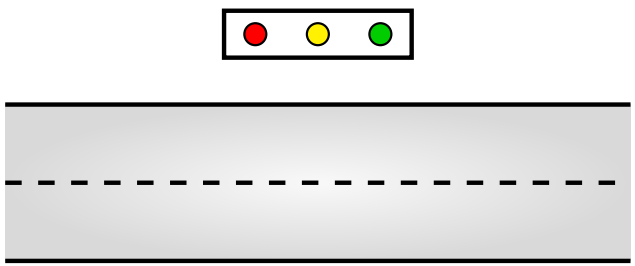
Shared variables

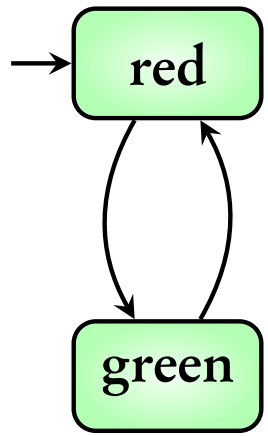
Shared actions



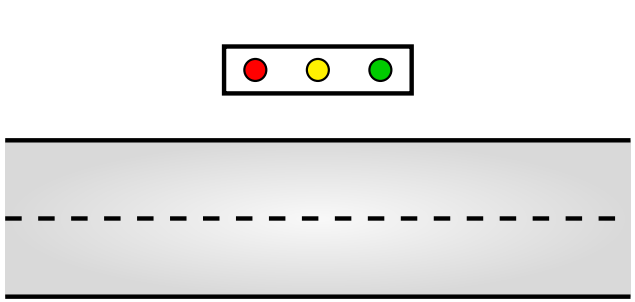


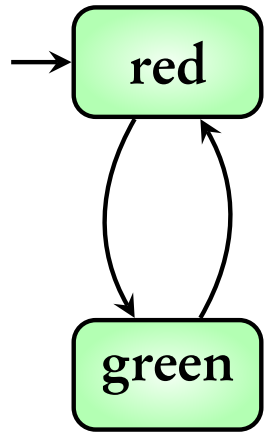
$TrLight_1$



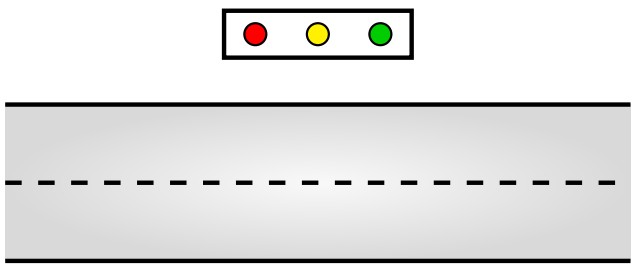


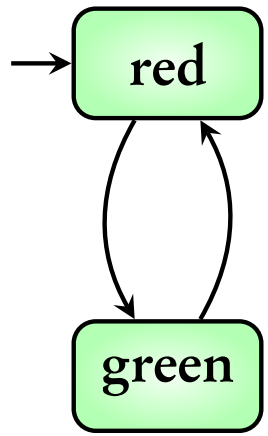
$TrLight_2$





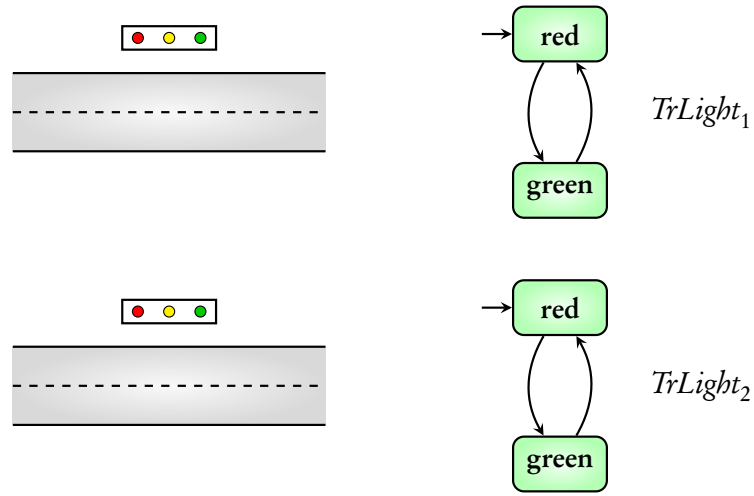
$TrLight_1$

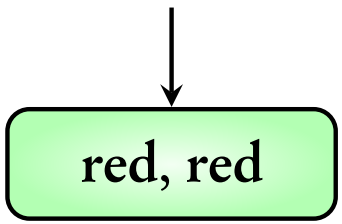


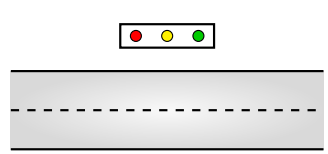


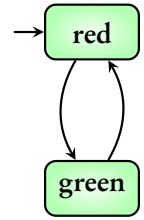
$TrLight_2$

What is the transition system for the **joint behaviour**?

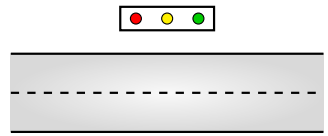

red
green
$TrLight_1$
red
green
$TrLight_2$

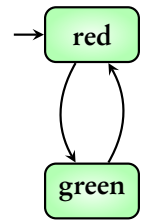




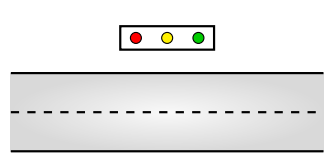


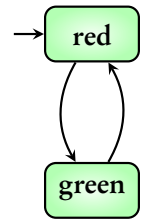
$TrLight_1$



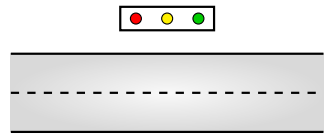


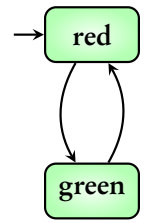
$TrLight_2$



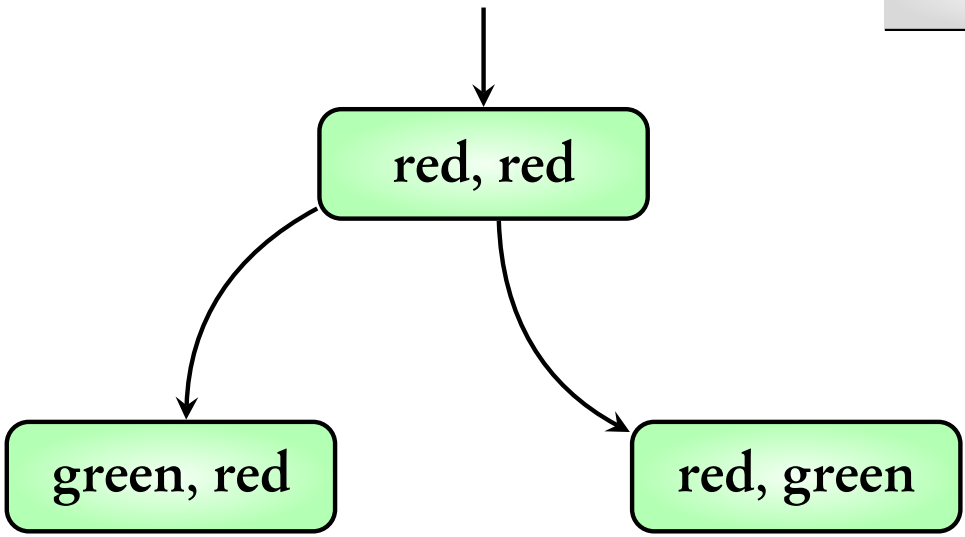


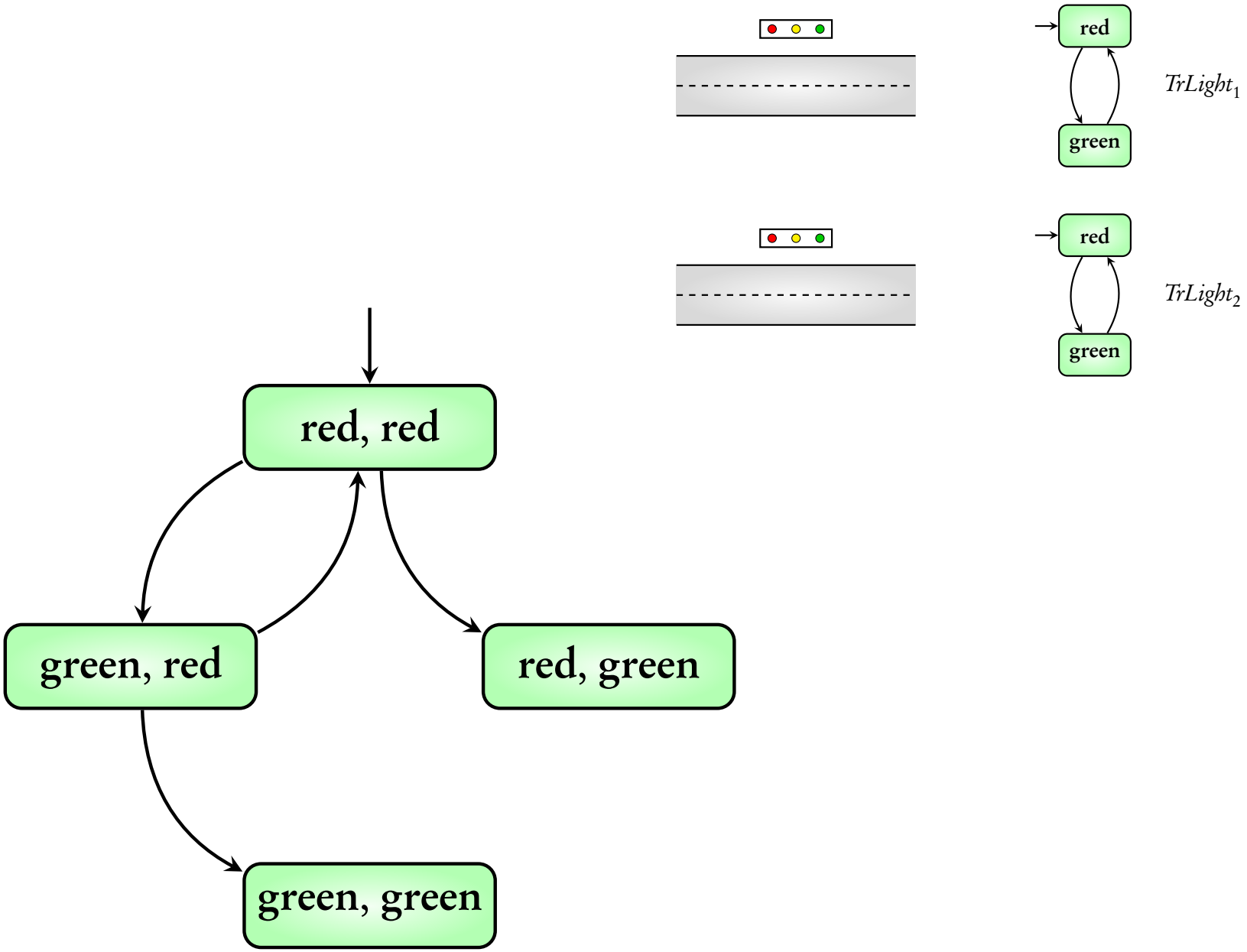
$TrLight_1$

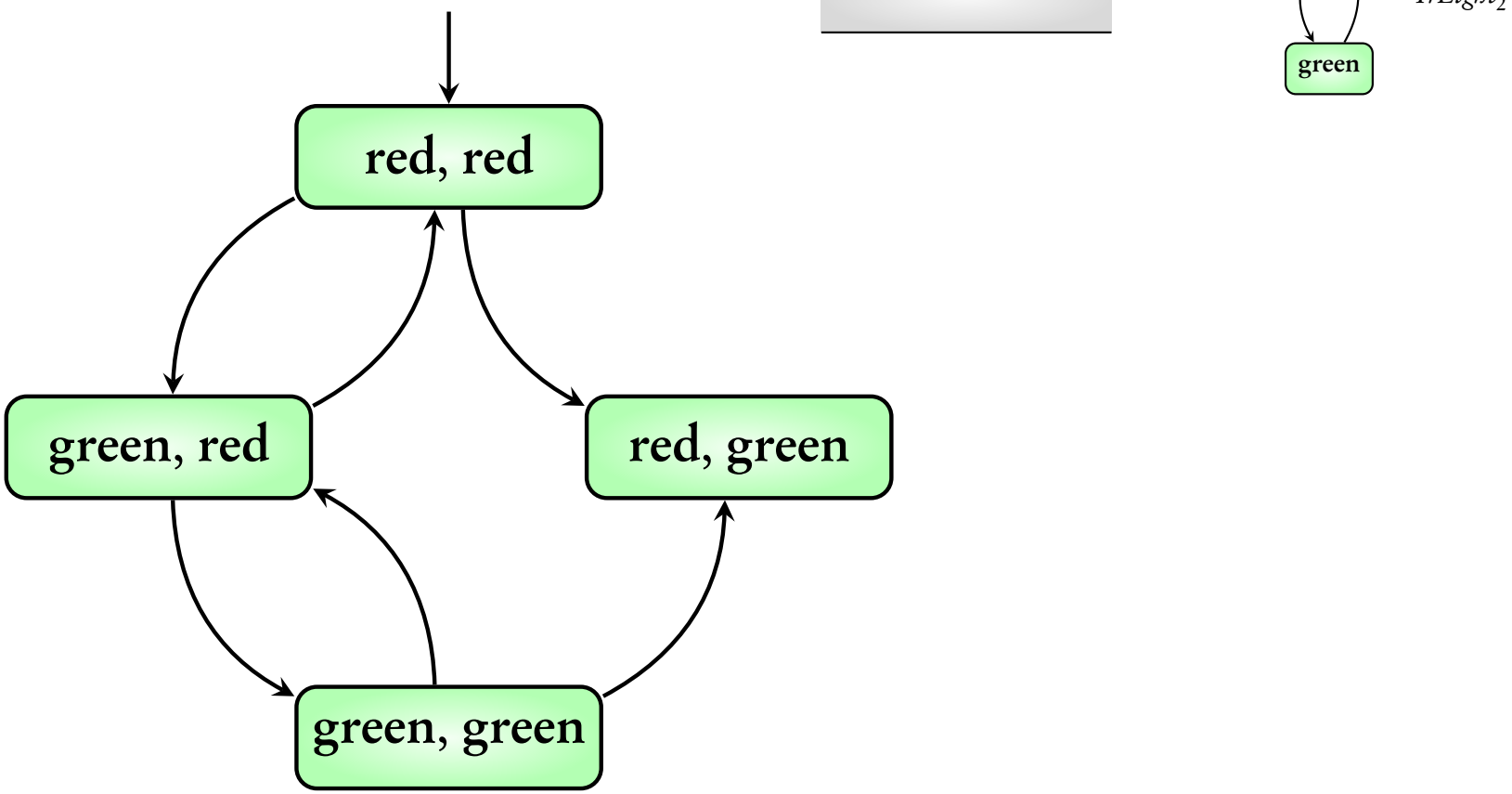


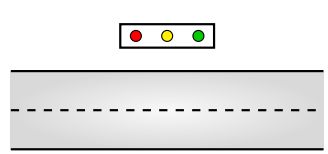


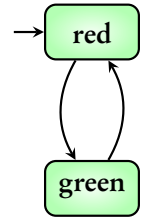
$TrLight_2$



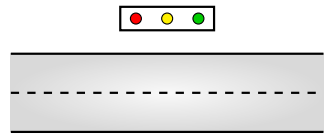

red
green
$TrLight_1$
red
green
$TrLight_2$
red, red
green, red
red, green
green, green

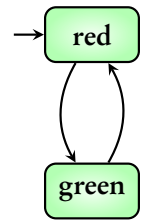




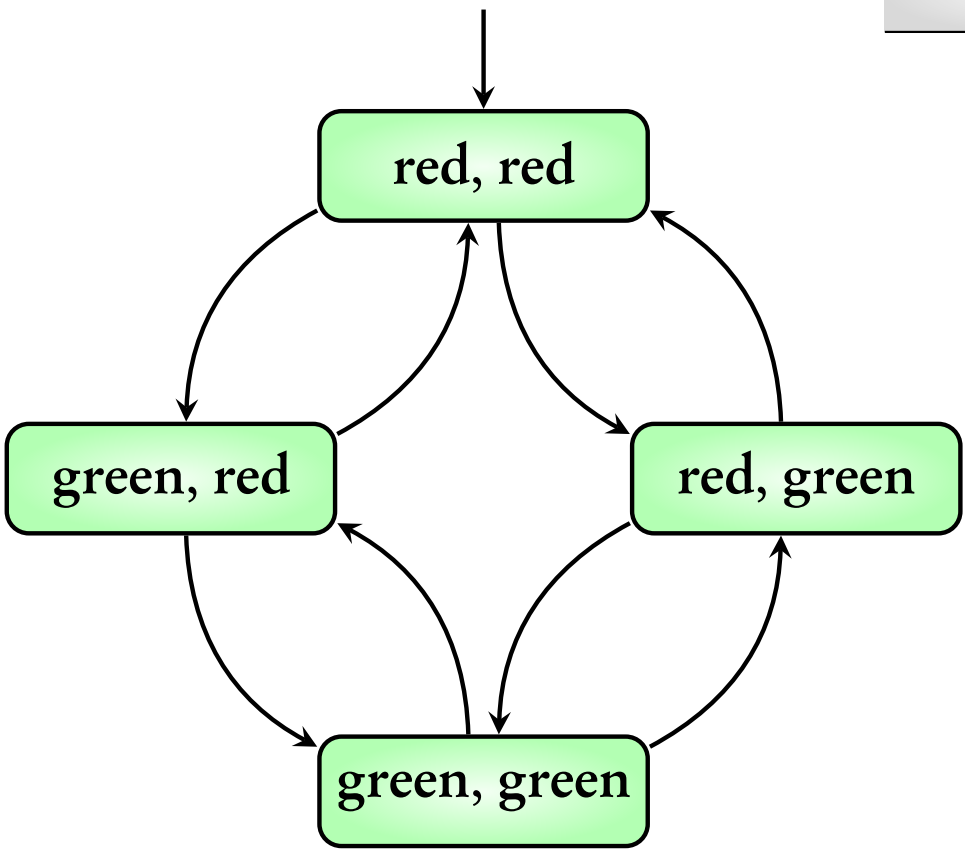


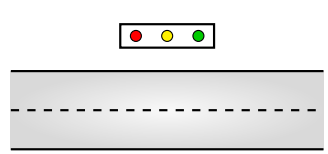
$TrLight_1$

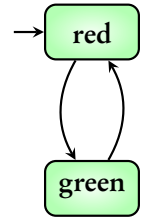




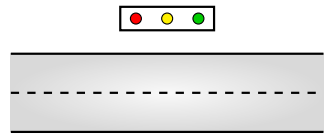
$TrLight_2$

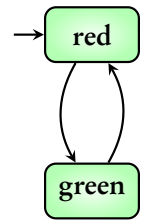




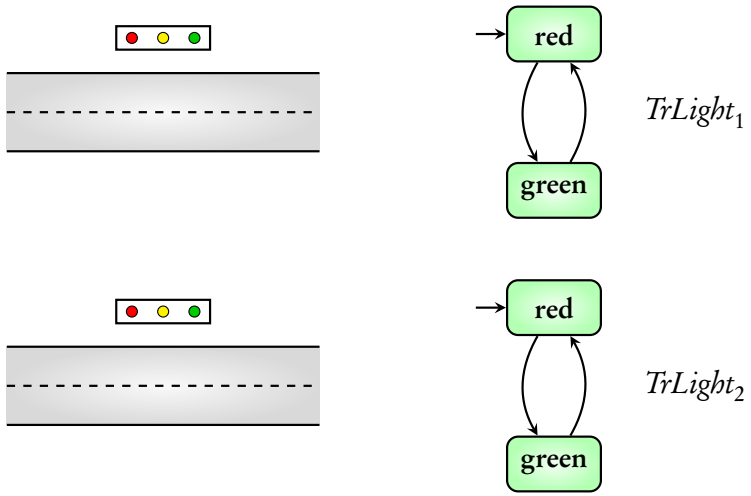


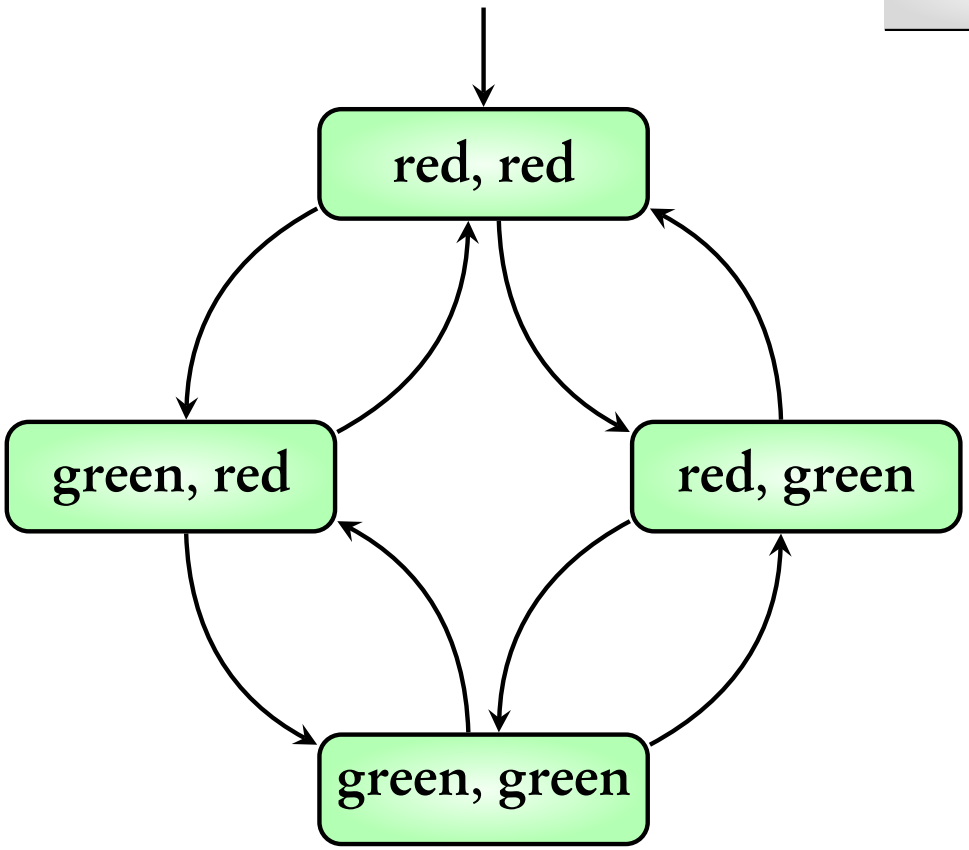
$TrLight_1$



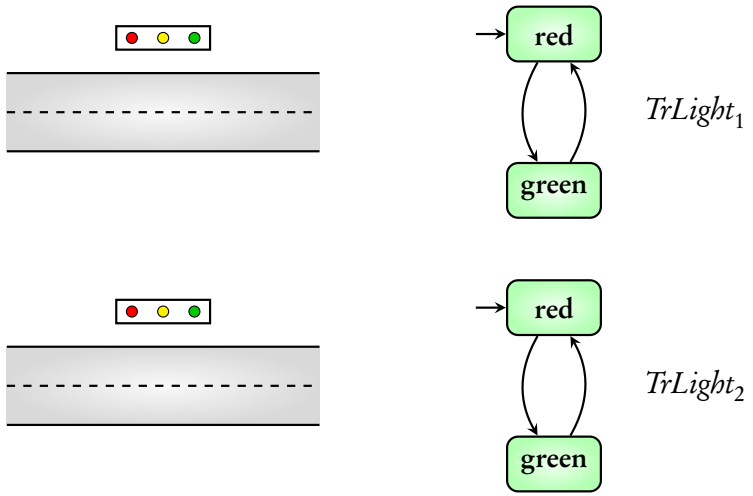


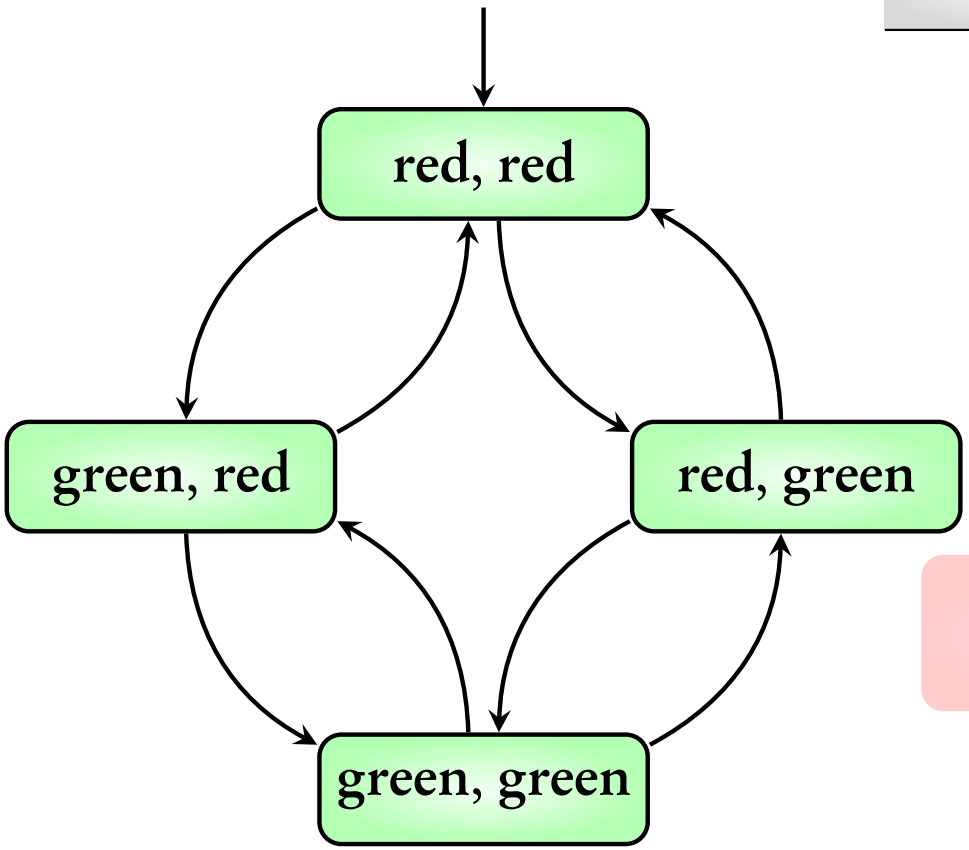
$TrLight_2$





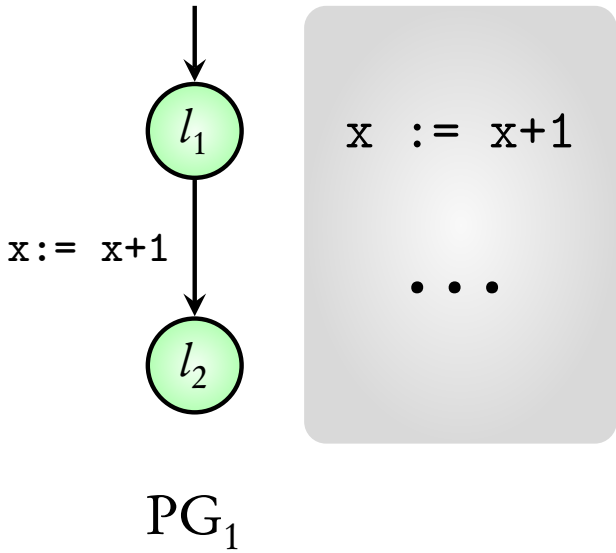
$$TrLight_1 \;|||\; TrLight_2$$



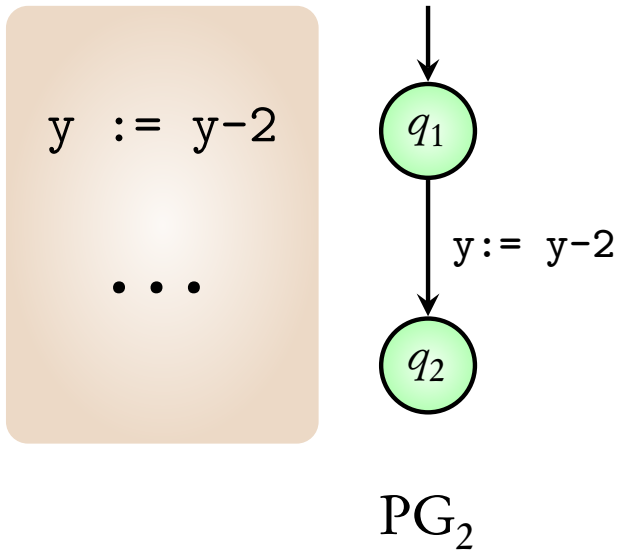


$$TrLight_1 \;|||\; TrLight_2$$

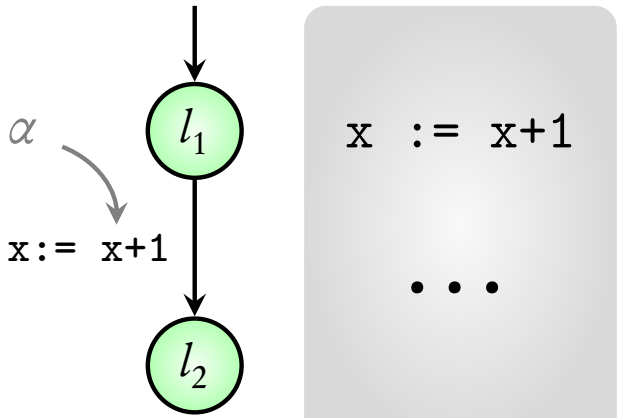
||| : Interleaving operator



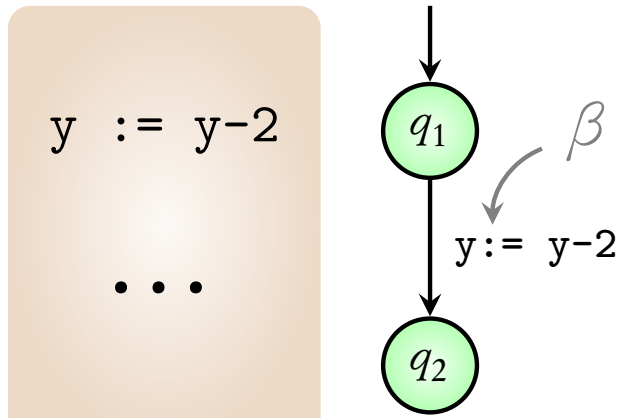
PG_1



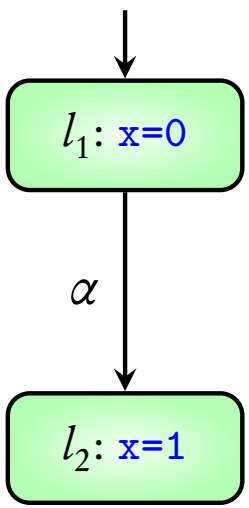
PG_2



PG_1

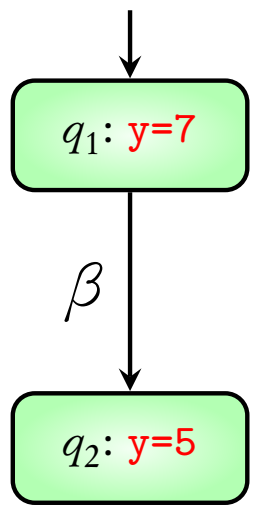


PG_2



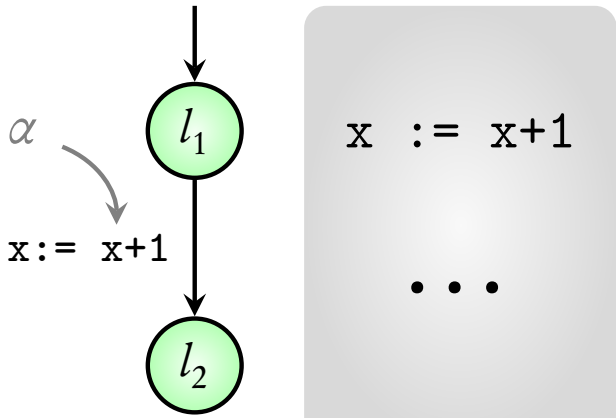
TS_1

(*initially* x=0)

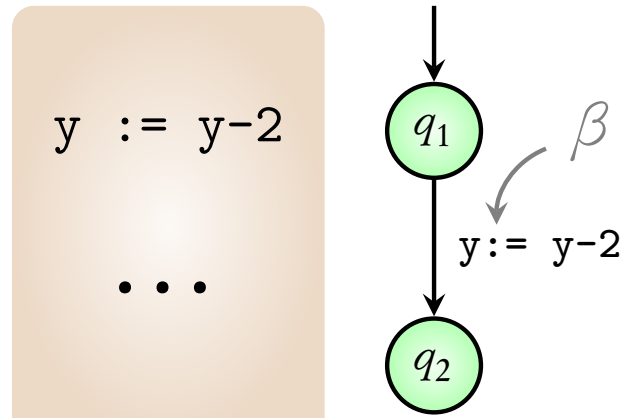


TS_2

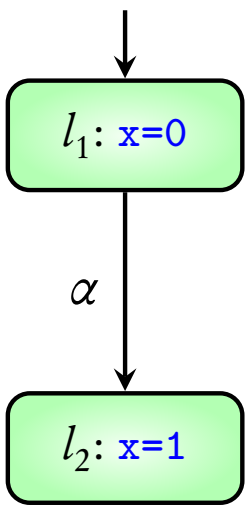
(*initially* y=7)



PG_1

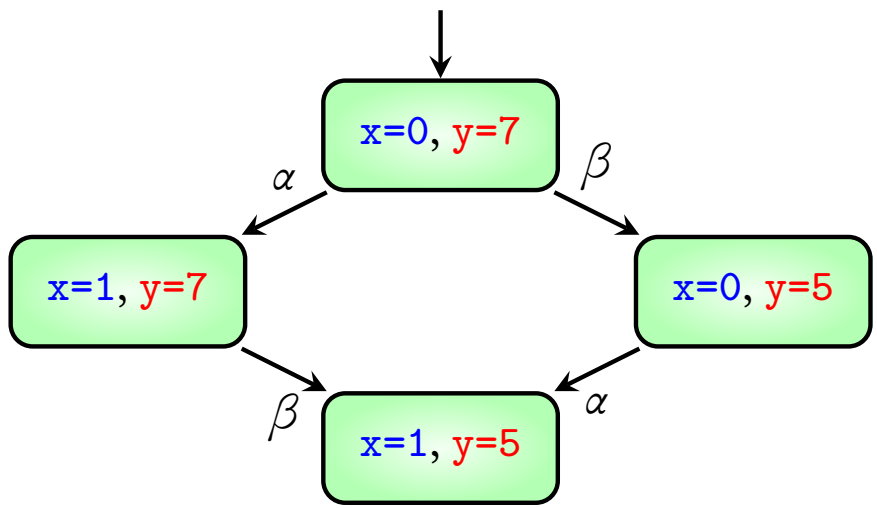


PG_2

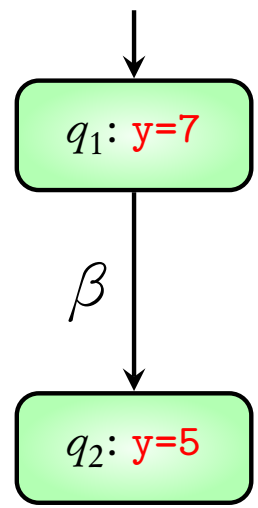


TS_1

(*initially* x=0)

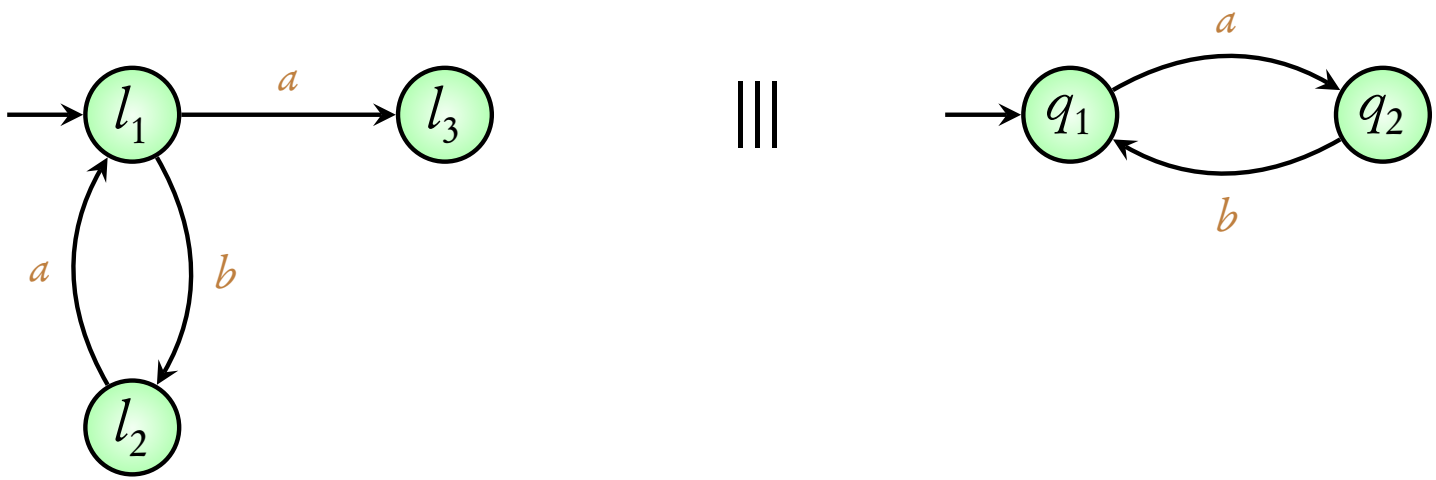


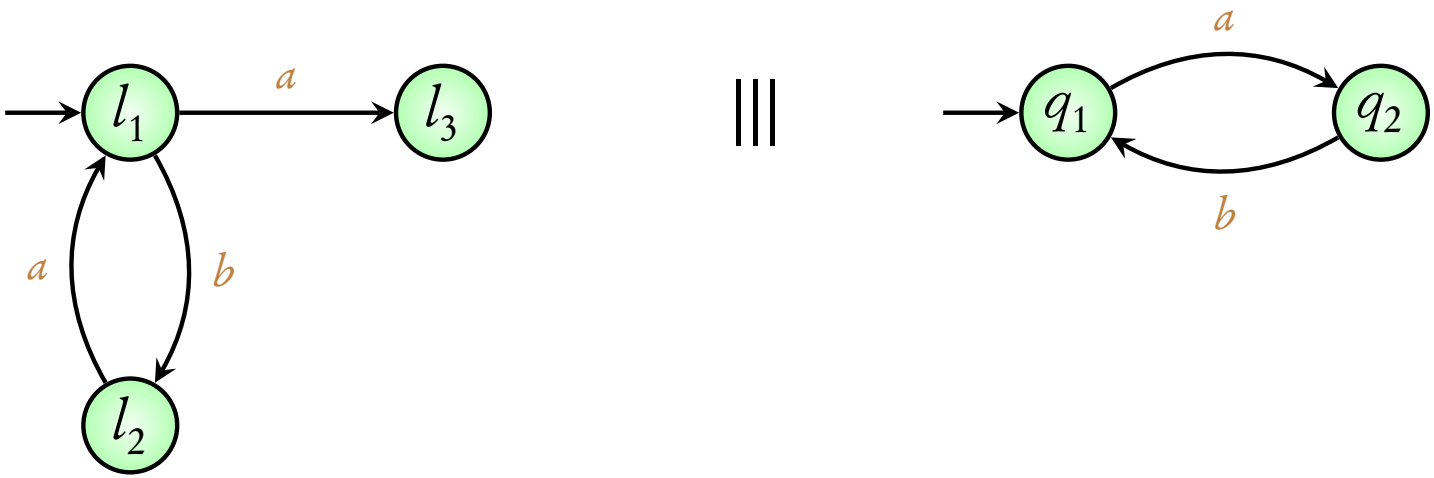
$TS_1 \;|||\; TS_2$

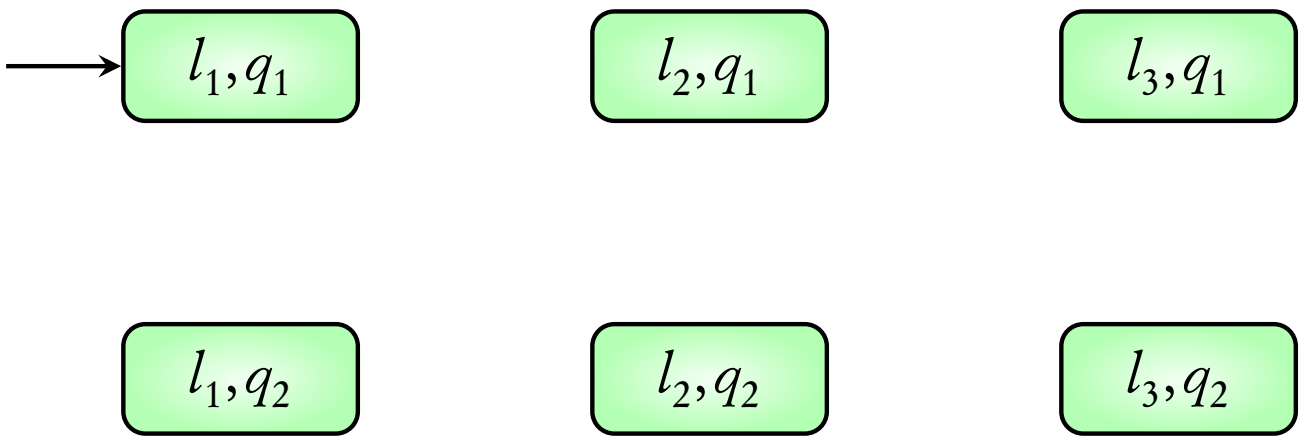


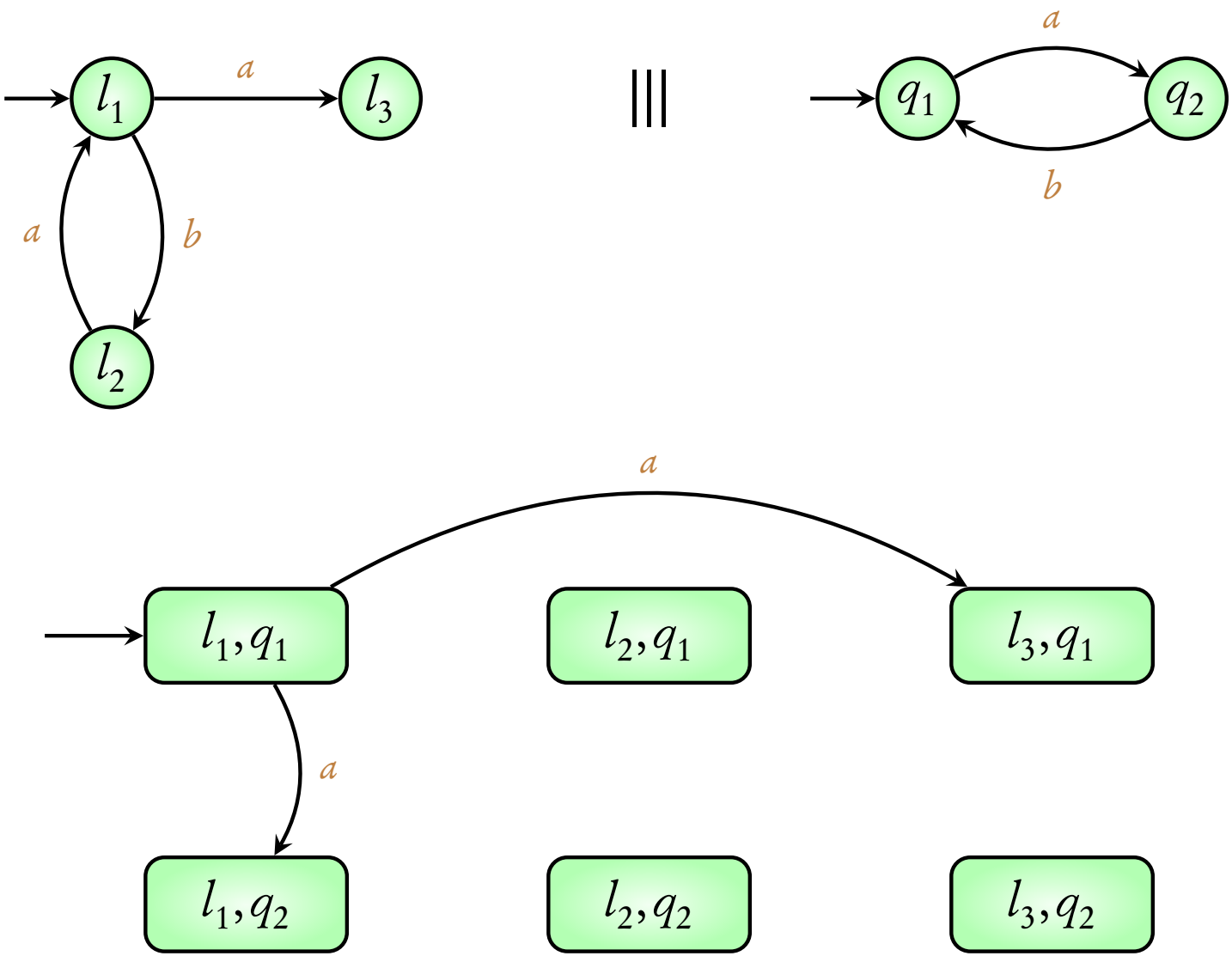
TS_2

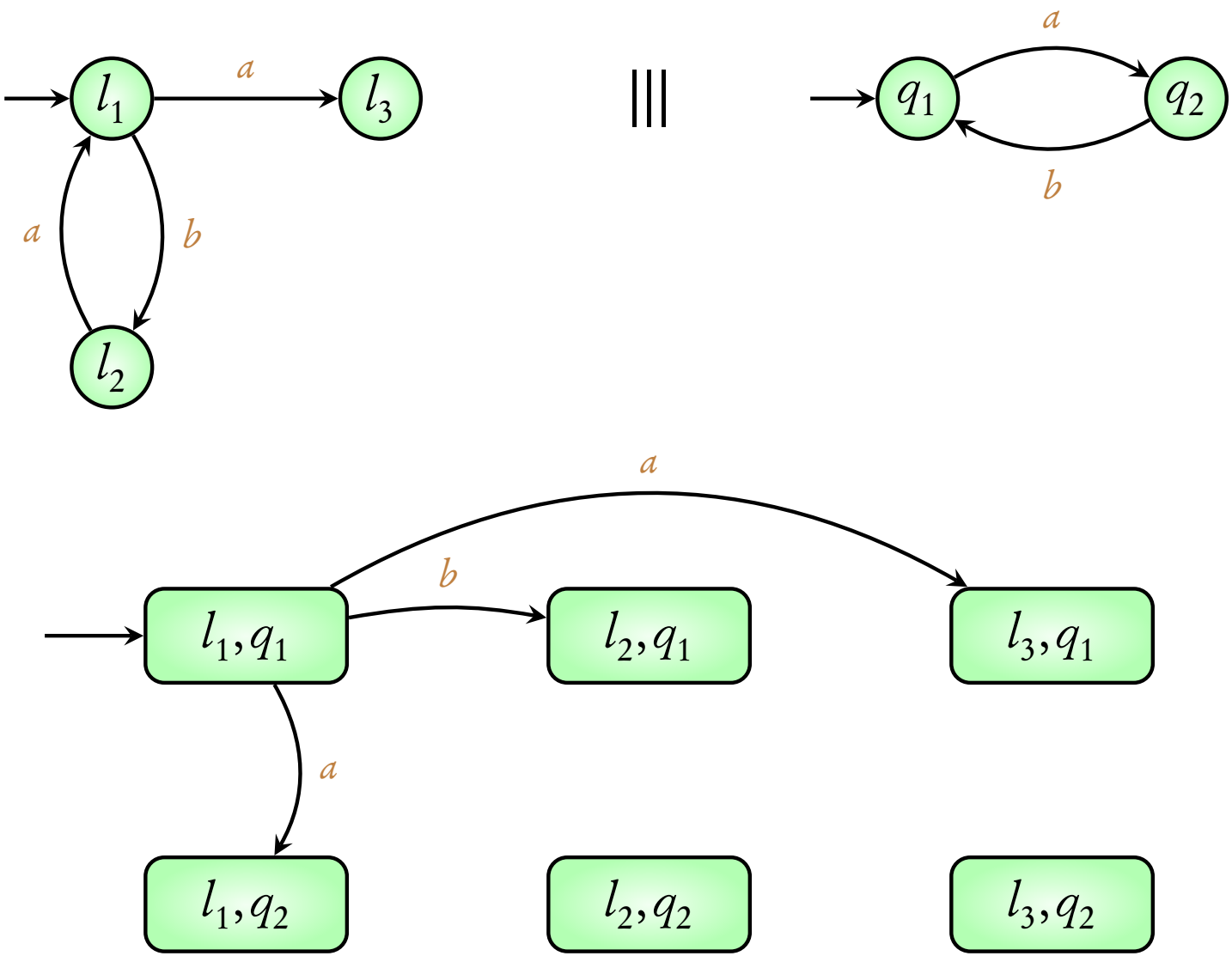
(*initially* y=7)

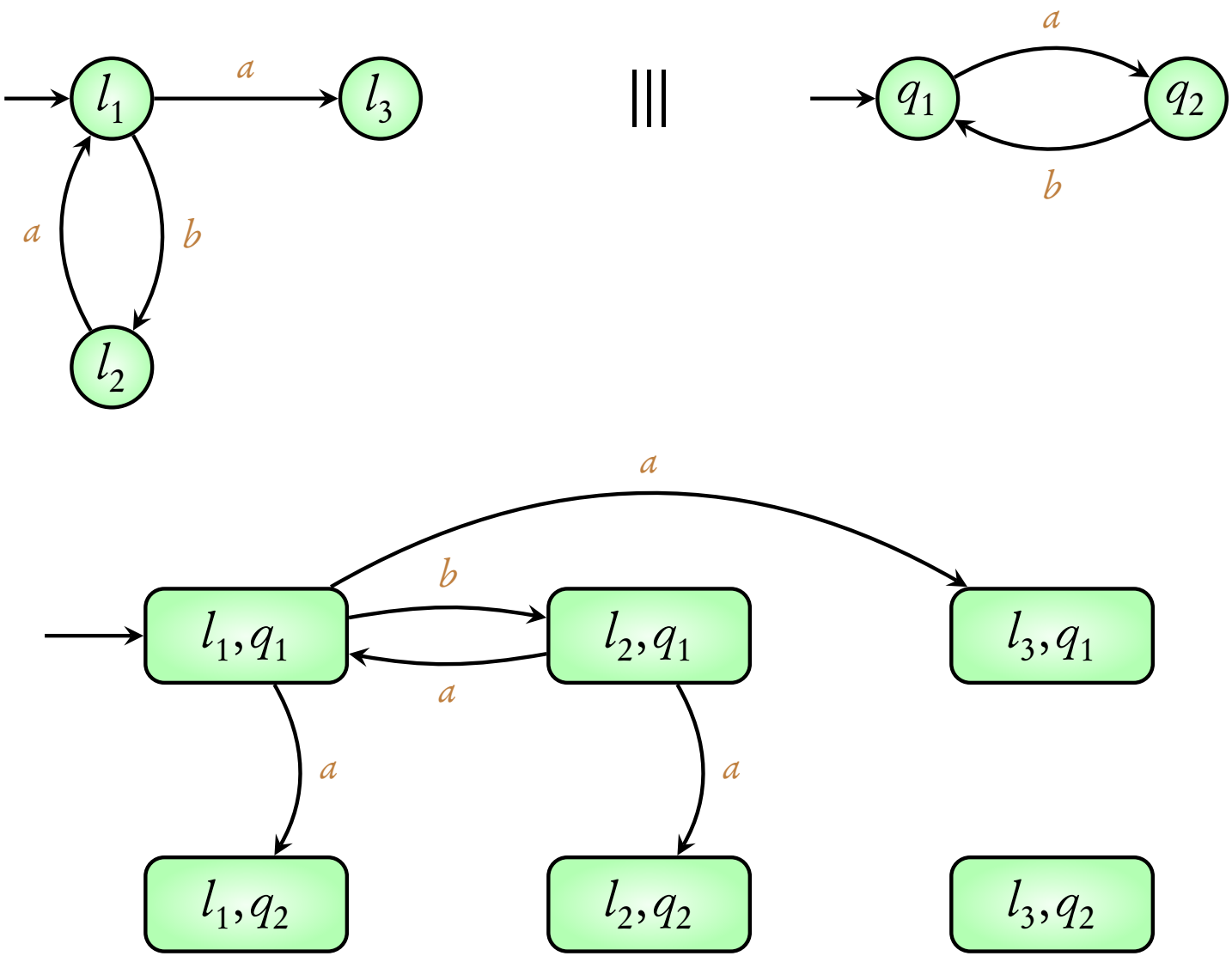

l_1
l_3
l_2
a
a
b
$\|\|$
q_1
q_2
a
b

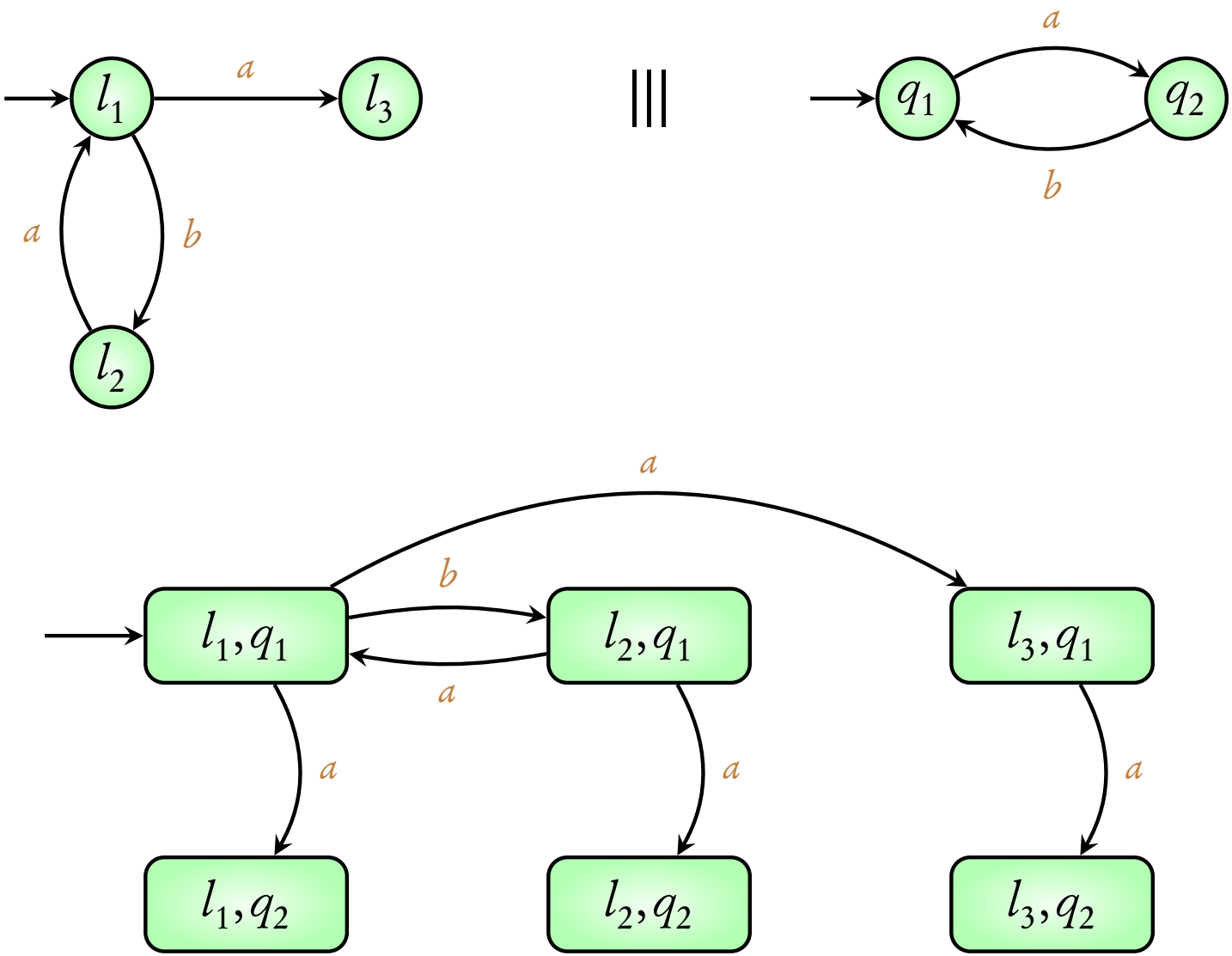

l_1
l_3
l_2
a
a
b
|||
q_1
q_2
a
b

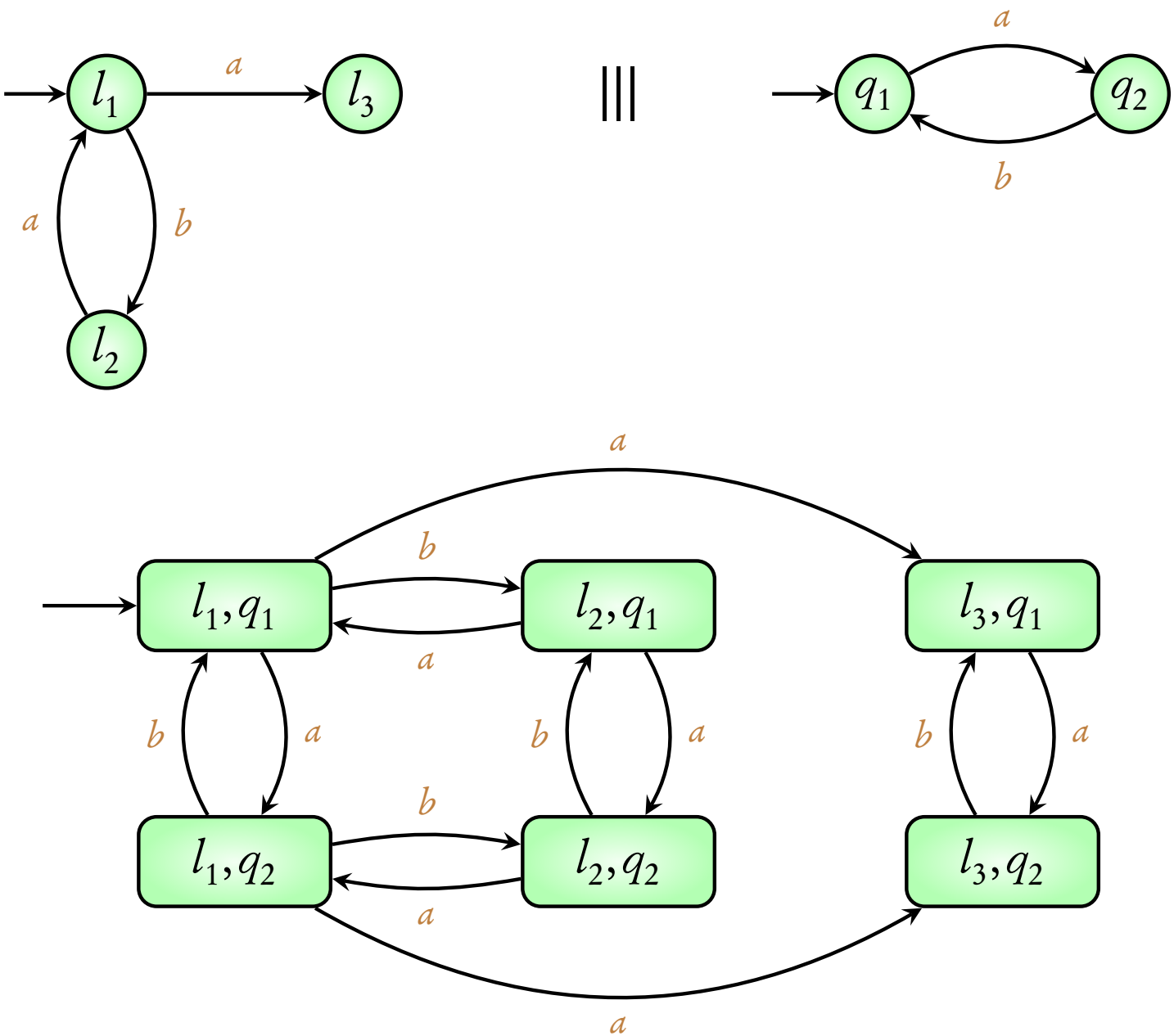

l_1, q_1
l_2, q_1
l_3, q_1
l_1, q_2
l_2, q_2
l_3, q_2


l_1
l_3
l_2
a
a
b
|||
q_1
q_2
a
b
a
l_1, q_1
l_2, q_1
l_3, q_1
a
l_1, q_2
l_2, q_2
l_3, q_2


l_1
l_3
l_2
a
a
b
|||
q_1
q_2
a
b
a
b
l_1, q_1
l_2, q_1
l_3, q_1
a
l_1, q_2
l_2, q_2
l_3, q_2


l_1
l_3
l_2
a
a
b
|||
q_1
q_2
a
b
a
b
a
a
a
l_1, q_1
l_2, q_1
l_3, q_1
l_1, q_2
l_2, q_2
l_3, q_2


l_1
l_3
l_2
a
a
b
|||
q_1
q_2
a
b
a
b
a
l_1, q_1
l_2, q_1
l_3, q_1
l_1, q_2
l_2, q_2
l_3, q_2
a
a
a


l_1
l_3
l_2
a
a
b
|||
q_1
q_2
a
b
l_1, q_1
l_2, q_1
l_3, q_1
l_1, q_2
l_2, q_2
l_3, q_2
a
b
a
b
a
b
a
b
a
b
a
a

Multiple systems

$$TS_1 \;|||\; TS_2 \;|||\; \ldots \;|||\; TS_n$$

Multiple systems

$$TS_1 \;|||\; TS_2 \;|||\; \ldots \;|||\; TS_n$$

Exercise: Try out an example of interleaving **three** systems

Concurrent systems

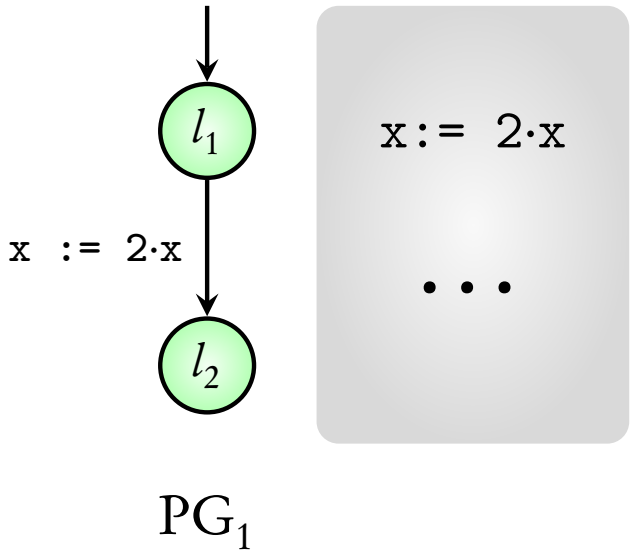
Independent

Interleaving

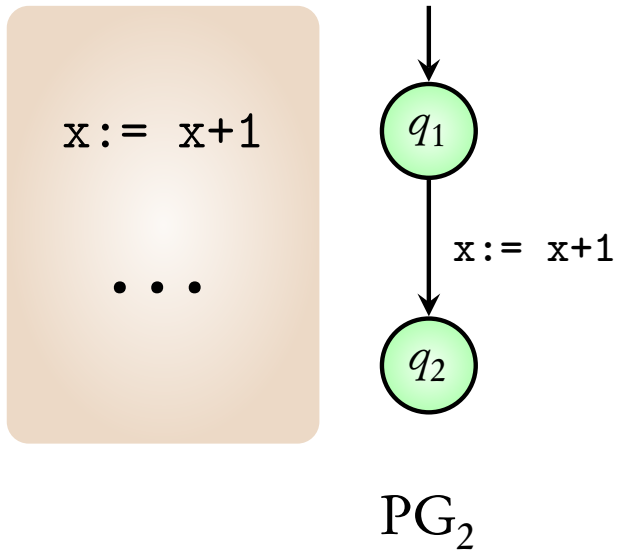
$TS_1 \;|||\; TS_2 \;|||\; \ldots \;|||\; TS_n$

Shared variables

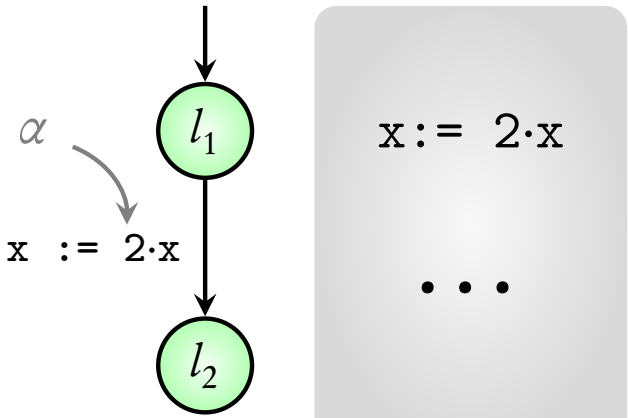
Shared actions



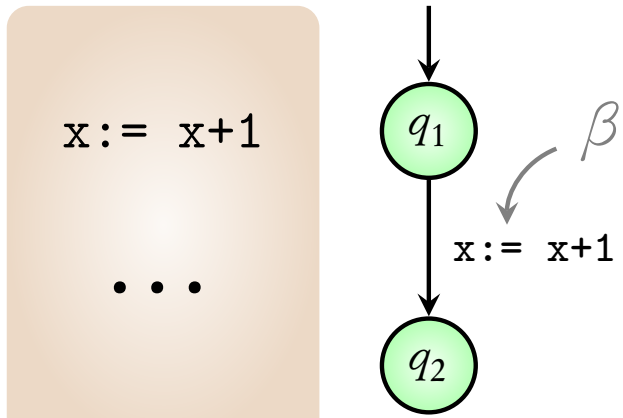
PG_1



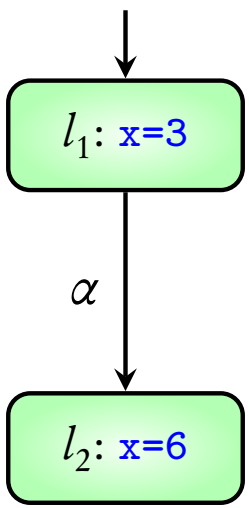
PG_2



PG_1

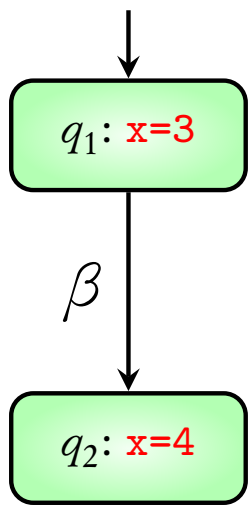


PG_2



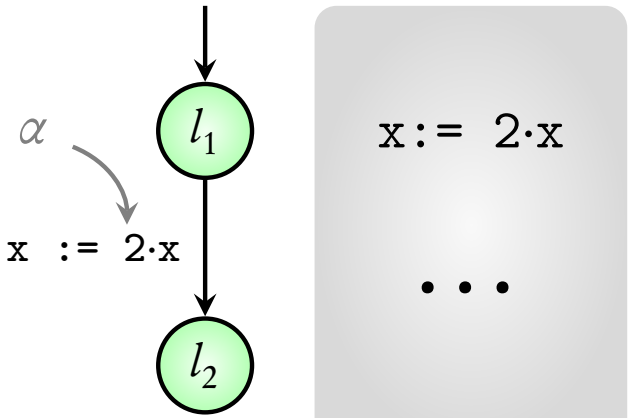
TS_1

(*initially* x=3)

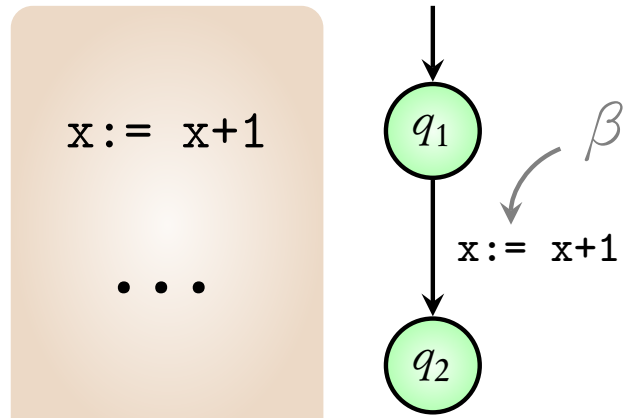


TS_2

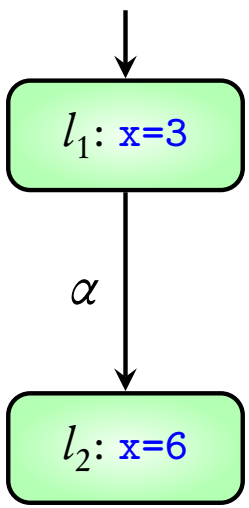
(*initially* x=3)



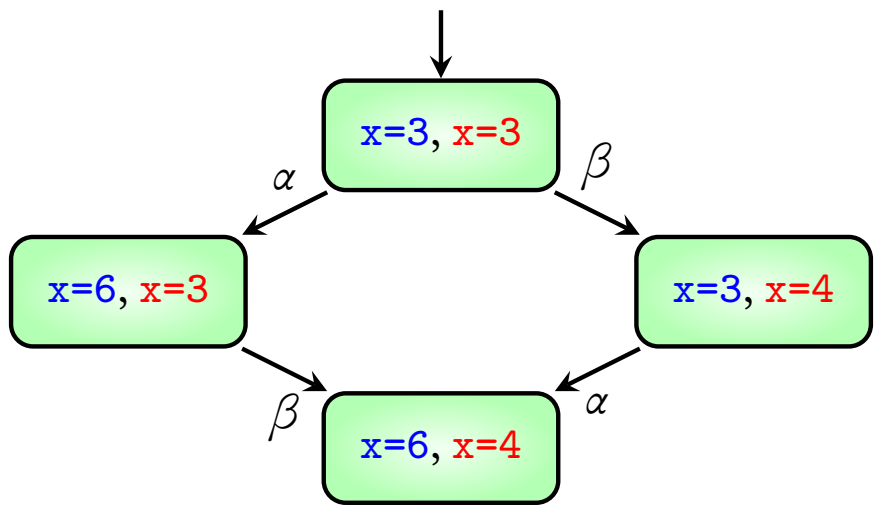
PG_1



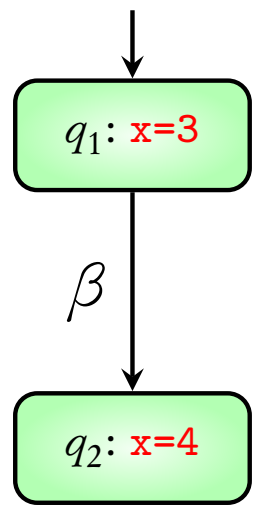
PG_2



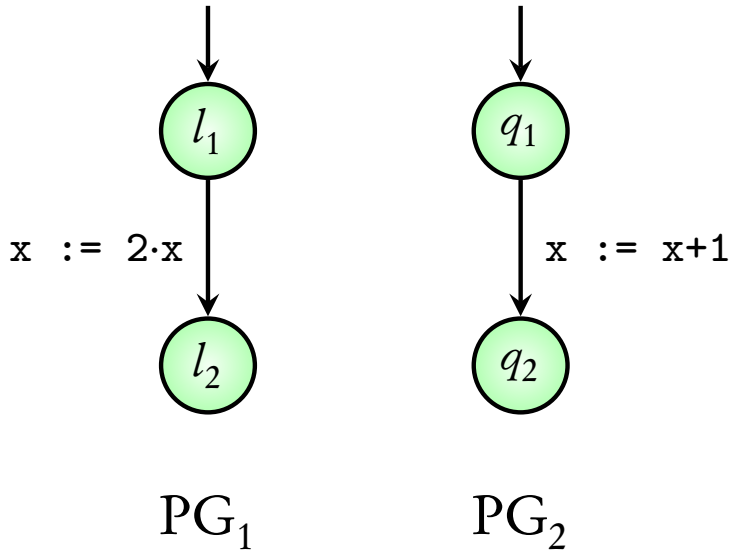
TS_1
(*initially* x=3)

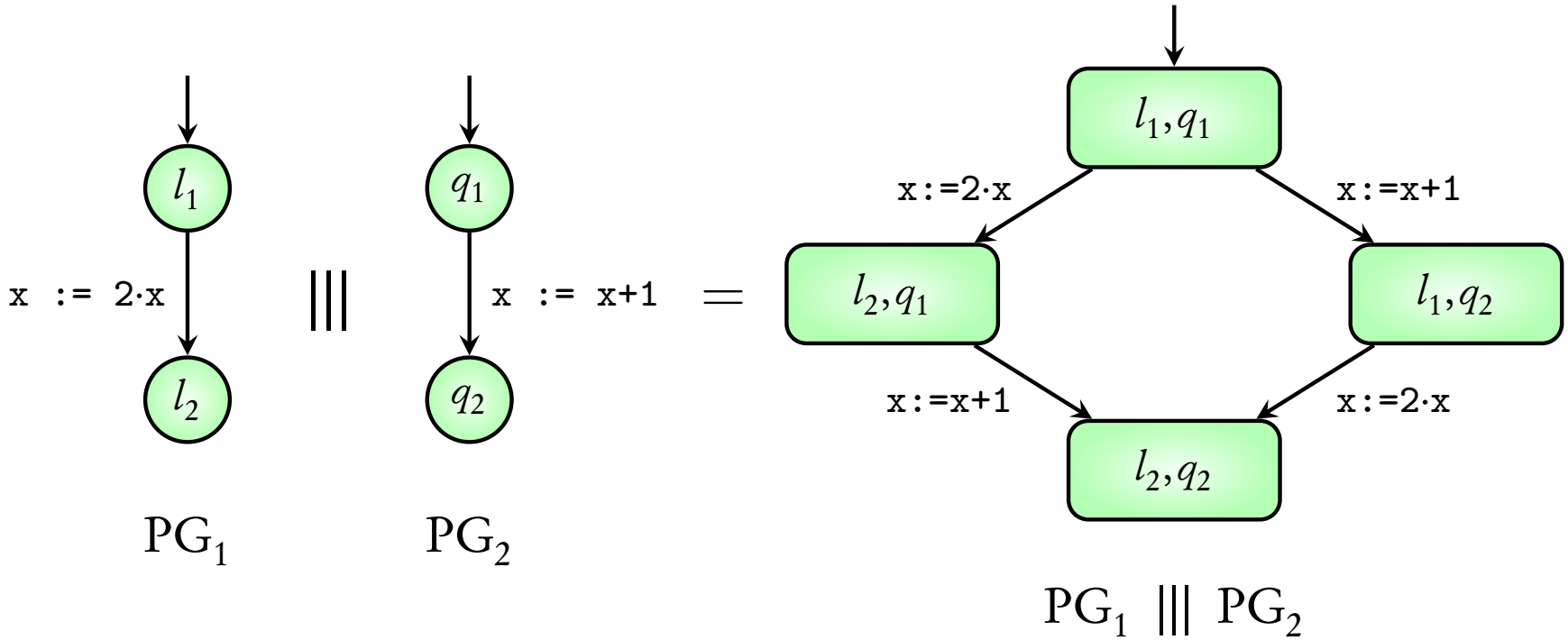


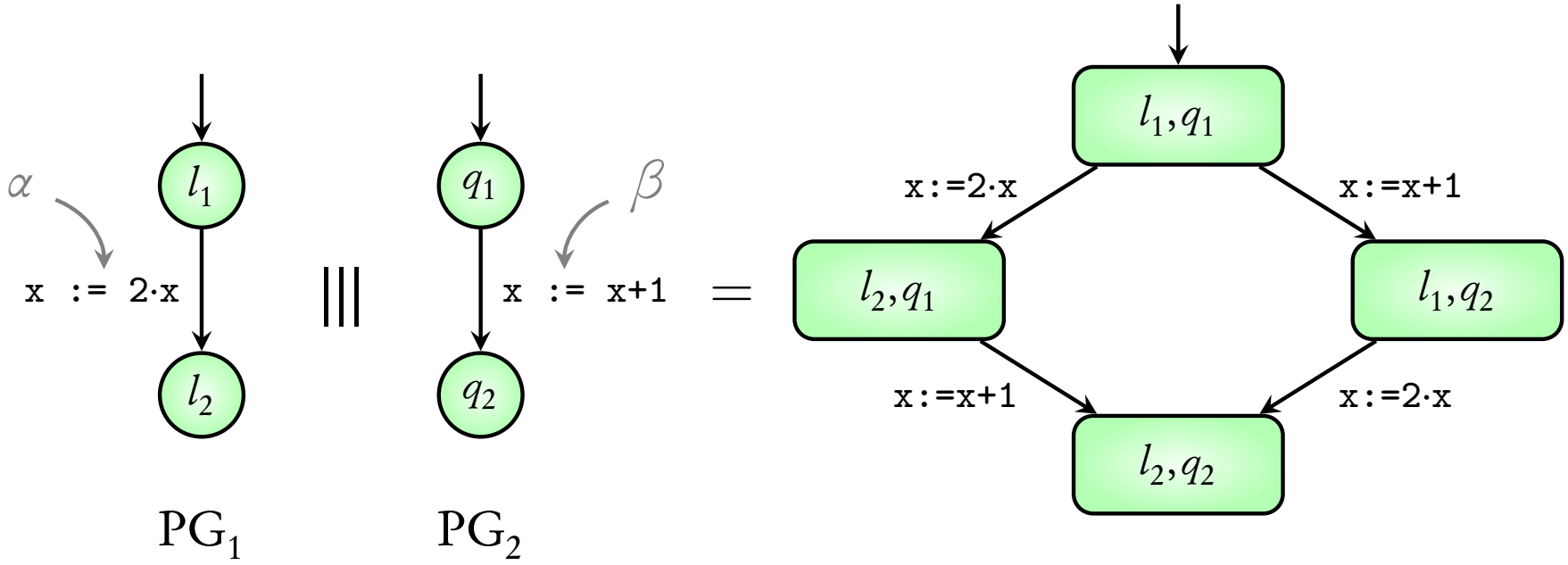
$TS_1 \;|||\; TS_2$



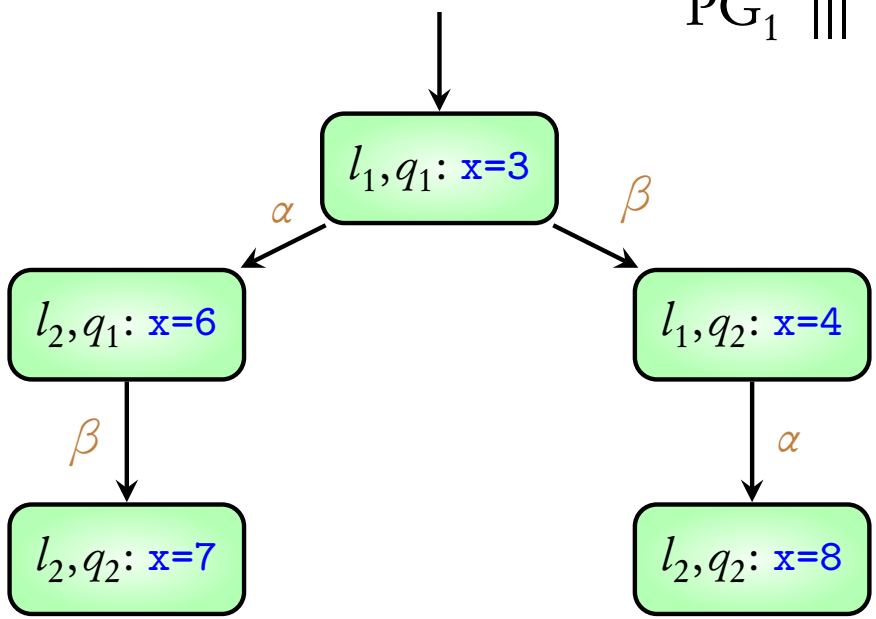
TS_2
(*initially* x=3)


l_1
x := 2·x
l_2
q_1
x := x+1
q_2
PG_1
PG_2


l_1
x := 2·x
l_2
q_1
x := x+1
q_2
=
l_1, q_1
x:=2·x
x:=x+1
l_2, q_1
l_1, q_2
x:=x+1
x:=2·x
l_2, q_2
PG_1
PG_2
$PG_1 \;|||\; PG_2$


α
x := 2·x
l_1
l_2
|||
q_1
q_2
β
x := x+1
=
l_1, q_1
x:=2·x
x:=x+1
l_2, q_1
l_1, q_2
x:=x+1
x:=2·x
l_2, q_2
PG1
PG2
PG1 ||| PG2

TS(PG$_1$ ||| PG$_2$):


l_1, q_1: x=3
α
β
l_2, q_1: x=6
l_1, q_2: x=4
β
α
l_2, q_2: x=7
l_2, q_2: x=8

Concurrent systems

Independent

Interleaving

$TS_1 \;|||\; TS_2 \;|||\; \ldots \;|||\; TS_n$

Shared variables

$TS(\,PG_1 \;|||\; PG_2 \;|||\; \ldots \;|||\; PG_n\,)$

Shared actions

Coming next: Another example

```
while x < 200
    x := x+1
```

```
while x>0
    x := x-1
```

```
while x=200
    x := 0
```

```
while x < 200
    x := x+1
```

```
while x>0
    x := x-1
```

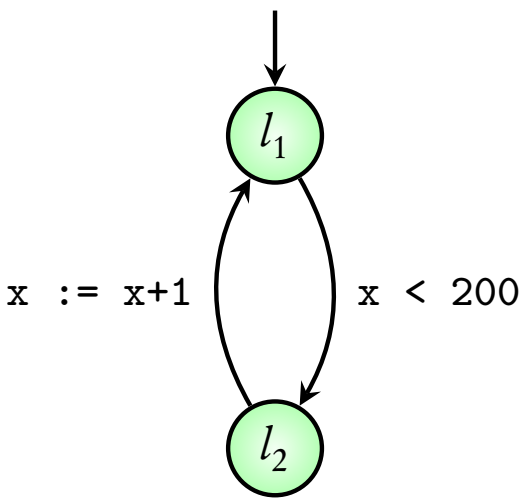
```
while x=200
    x := 0
```

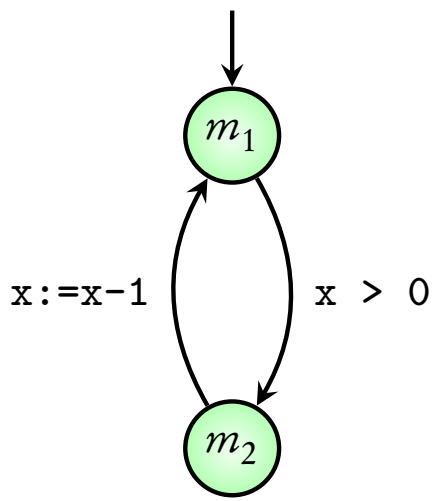
Is the value of x **always between** 0 and 200?

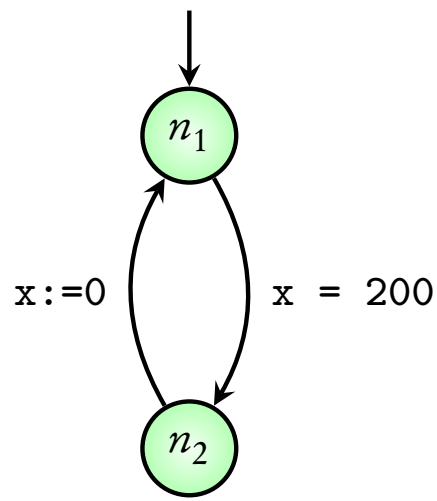
```
while x < 200
    x := x+1
```

```
while x>0
    x := x-1
```

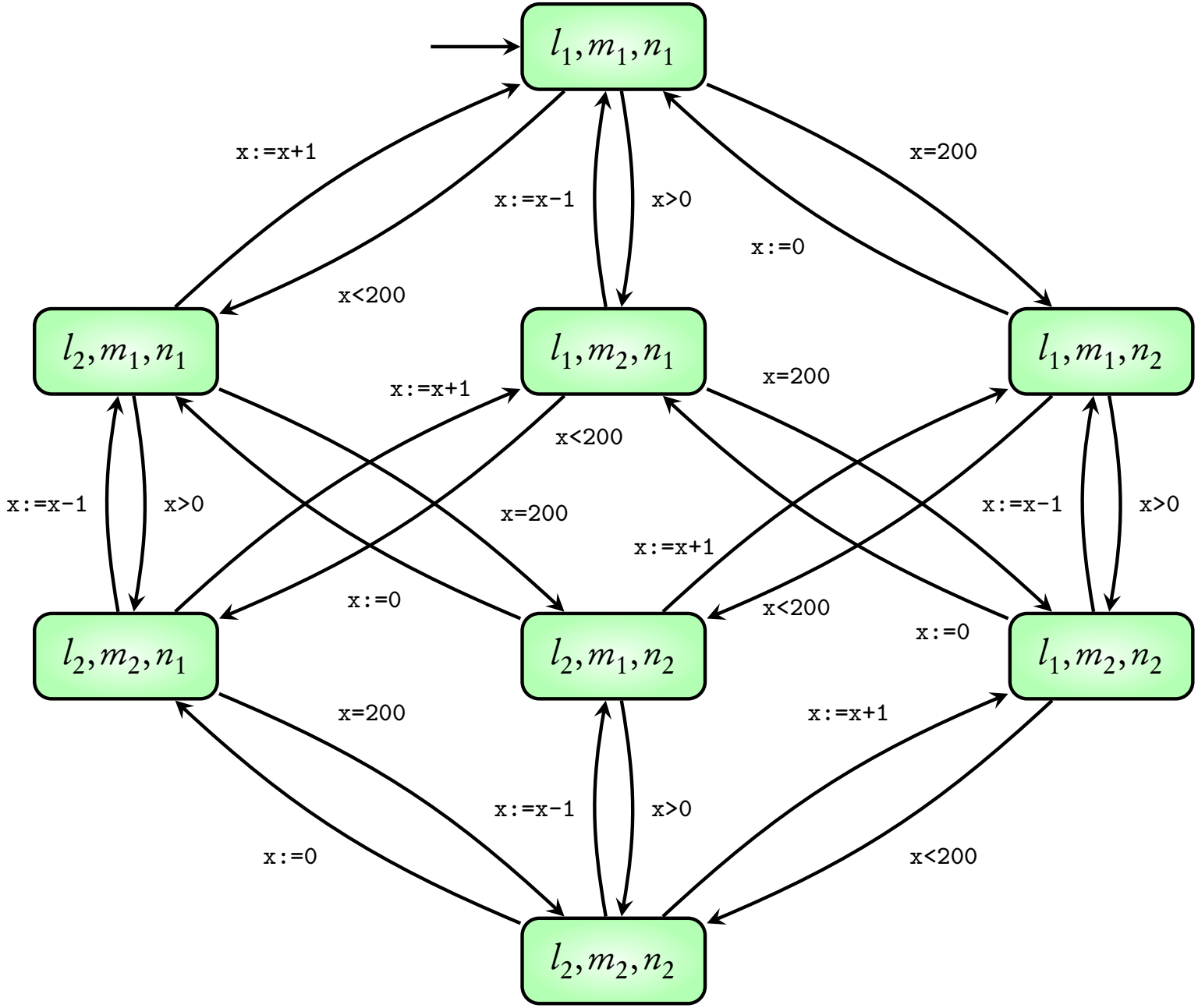
```
while x=200
    x := 0
```

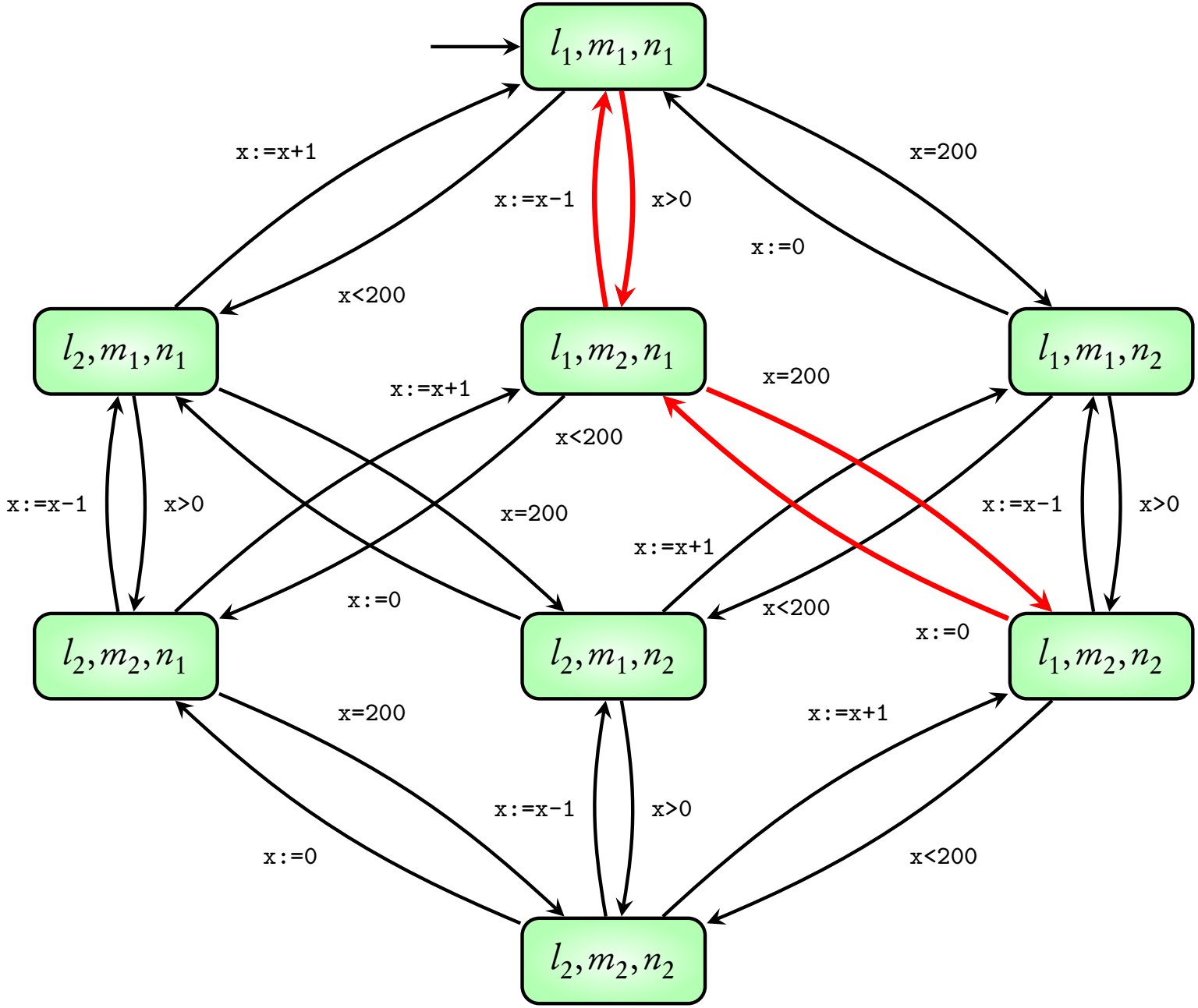


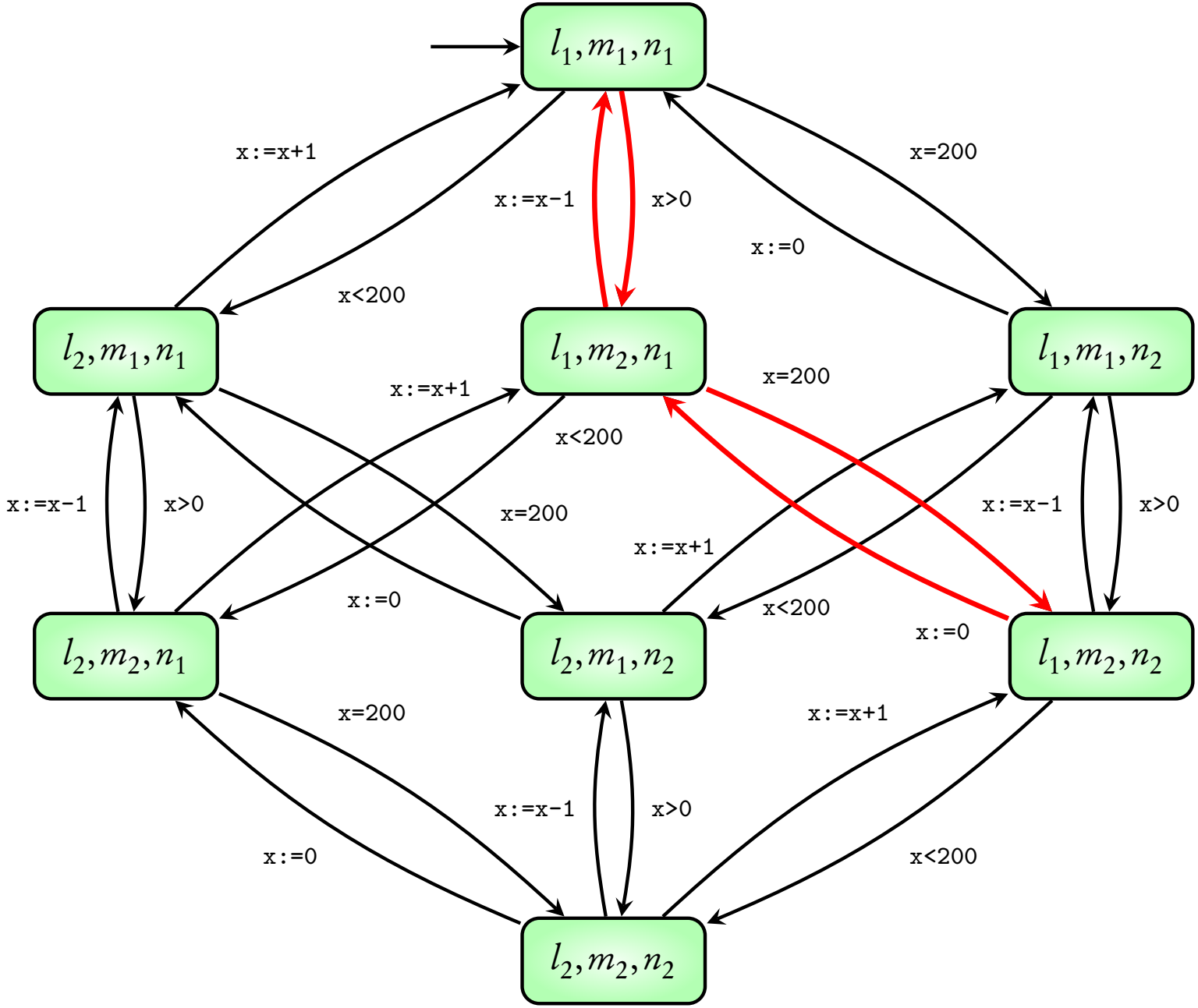




Is the value of x **always between** 0 and 200?

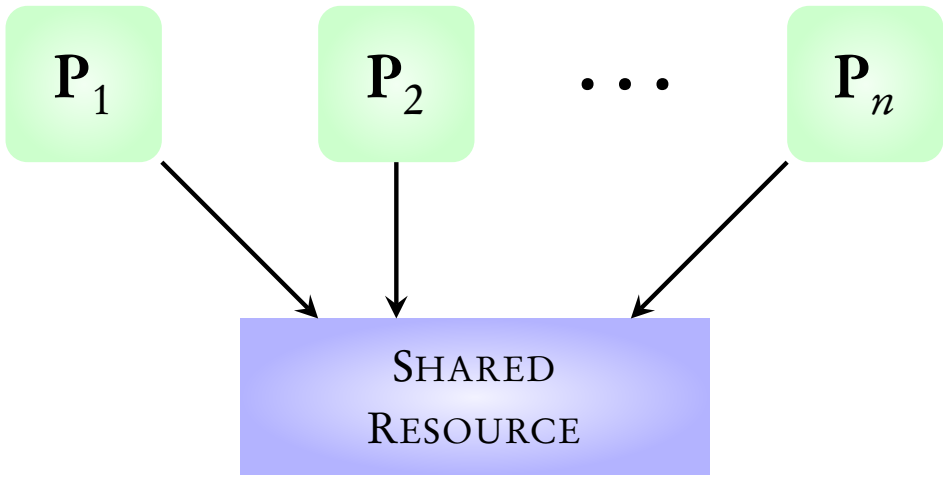

l_1, m_1, n_1
l_2, m_1, n_1
l_1, m_2, n_1
l_1, m_1, n_2
l_2, m_2, n_1
l_2, m_1, n_2
l_1, m_2, n_2
l_2, m_2, n_2
x:=x+1
x:=x-1
x>0
x=200
x:=0
x<200
x:=x+1
x=200
x<200
x:=x-1
x>0
x=200
x:=x+1
x:=0
x<200
x:=0
x=200
x:=x-1
x>0
x:=x+1
x:=0
x<200


l_1, m_1, n_1
l_2, m_1, n_1
l_1, m_2, n_1
l_1, m_1, n_2
l_2, m_2, n_1
l_2, m_1, n_2
l_1, m_2, n_2
l_2, m_2, n_2
x:=x+1
x:=x-1
x>0
x=200
x:=0
x<200
x:=x+1
x=200
x<200
x:=x-1
x>0
x=200
x:=x+1
x:=0
x<200
x:=0
x=200
x:=x+1
x:=x-1
x>0
x:=0
x<200



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

Coming next: Mutual exclusion



(variable, printer, ...)

Mutual Exclusion: **No two processes** can access the resource **simultaneously**

Goal: Modeling the **protocols** used for mutual exclusion

P_1

loop forever

⋮ *non-critical actions*

request

critical section

release

⋮ *non-critical actions*

end loop

P_2

loop forever

⋮ *non-critical actions*

request

critical section

release

⋮ *non-critical actions*

end loop

P_1

loop forever

⋮ *non-critical actions*

request

critical section

release

⋮ *non-critical actions*

end loop

P_2

loop forever

⋮ *non-critical actions*

request

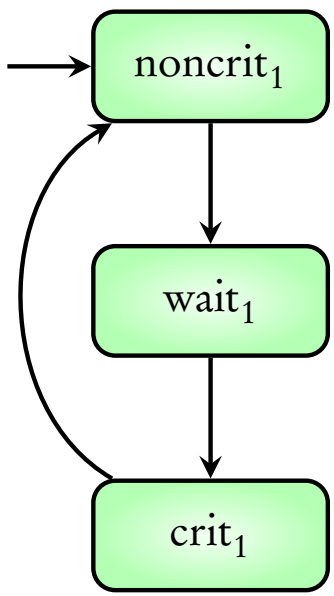
critical section

release

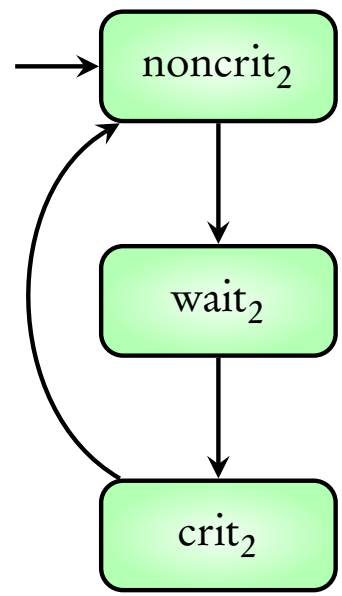
⋮ *non-critical actions*

end loop

PG_1



PG_2



P_1

loop forever

⋮ *non-critical actions*

⟨ if y>0: y:=y−1 ⟩ *request*

critical section

y:=y+1 *release*

⋮ *non-critical actions*

end loop

P_2

loop forever

⋮ *non-critical actions*

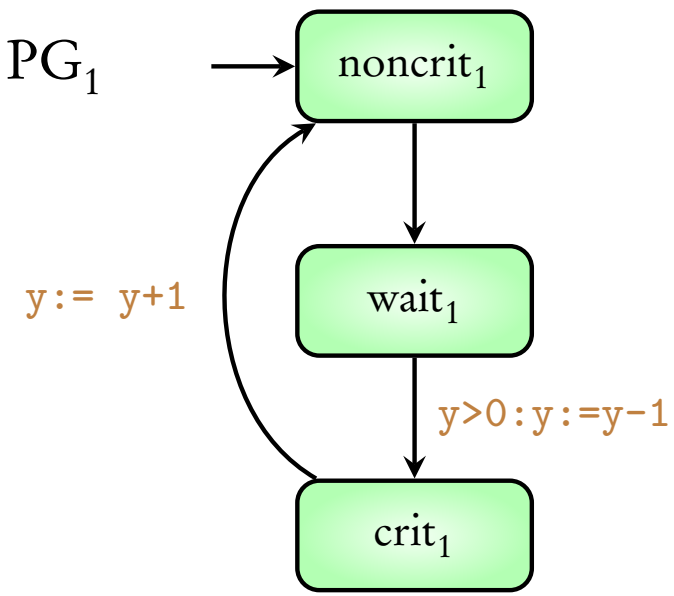
⟨ if y>0: y:=y−1 ⟩ *request*

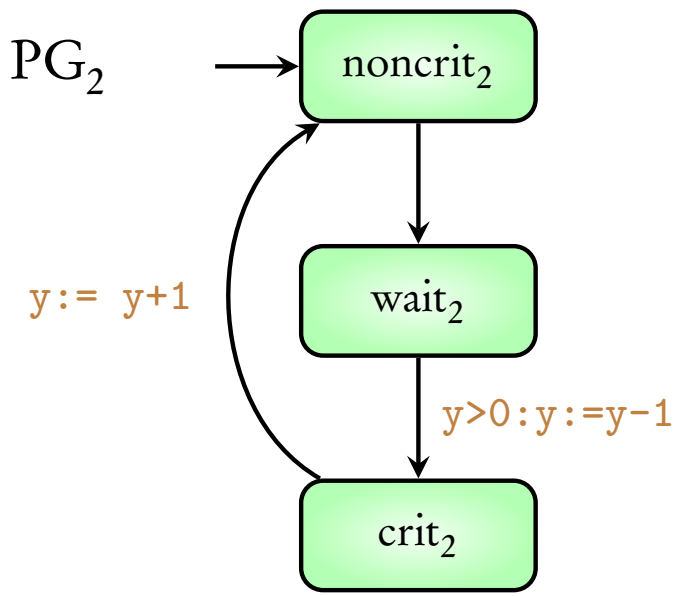
critical section

y:=y+1 *release*

⋮ *non-critical actions*

end loop





P$_1$

loop forever

⋮ *non-critical actions*

⟨ if y>0: y:=y−1 ⟩ *request*

critical section

y:=y+1 *release*

⋮ *non-critical actions*

end loop

P$_2$

loop forever

⋮ *non-critical actions*

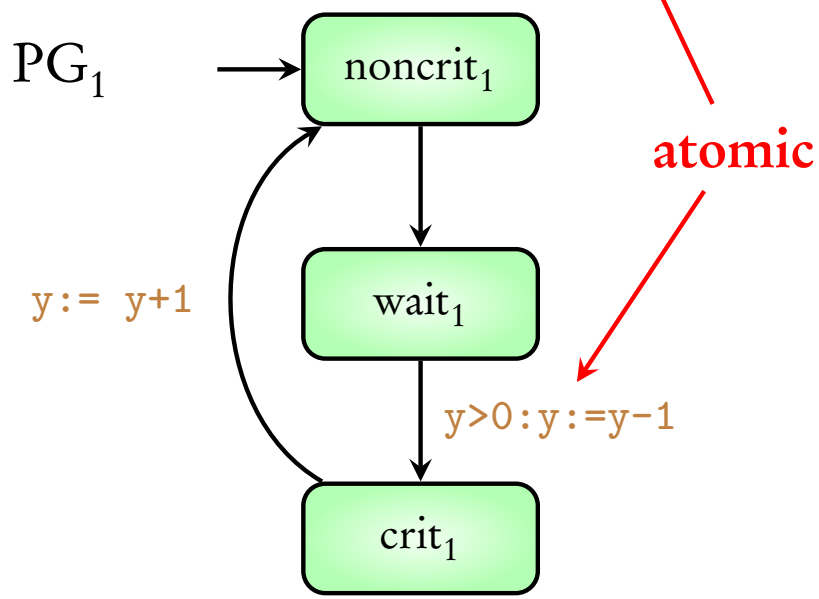
⟨ if y>0: y:=y−1 ⟩ *request*

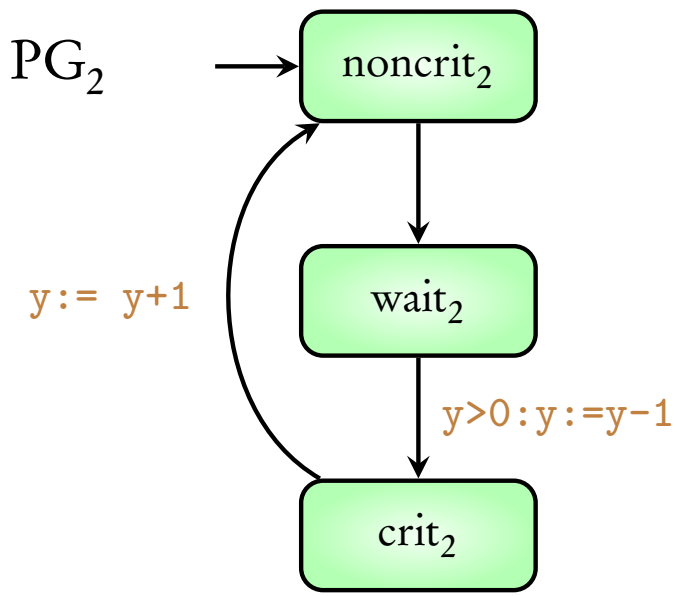
critical section

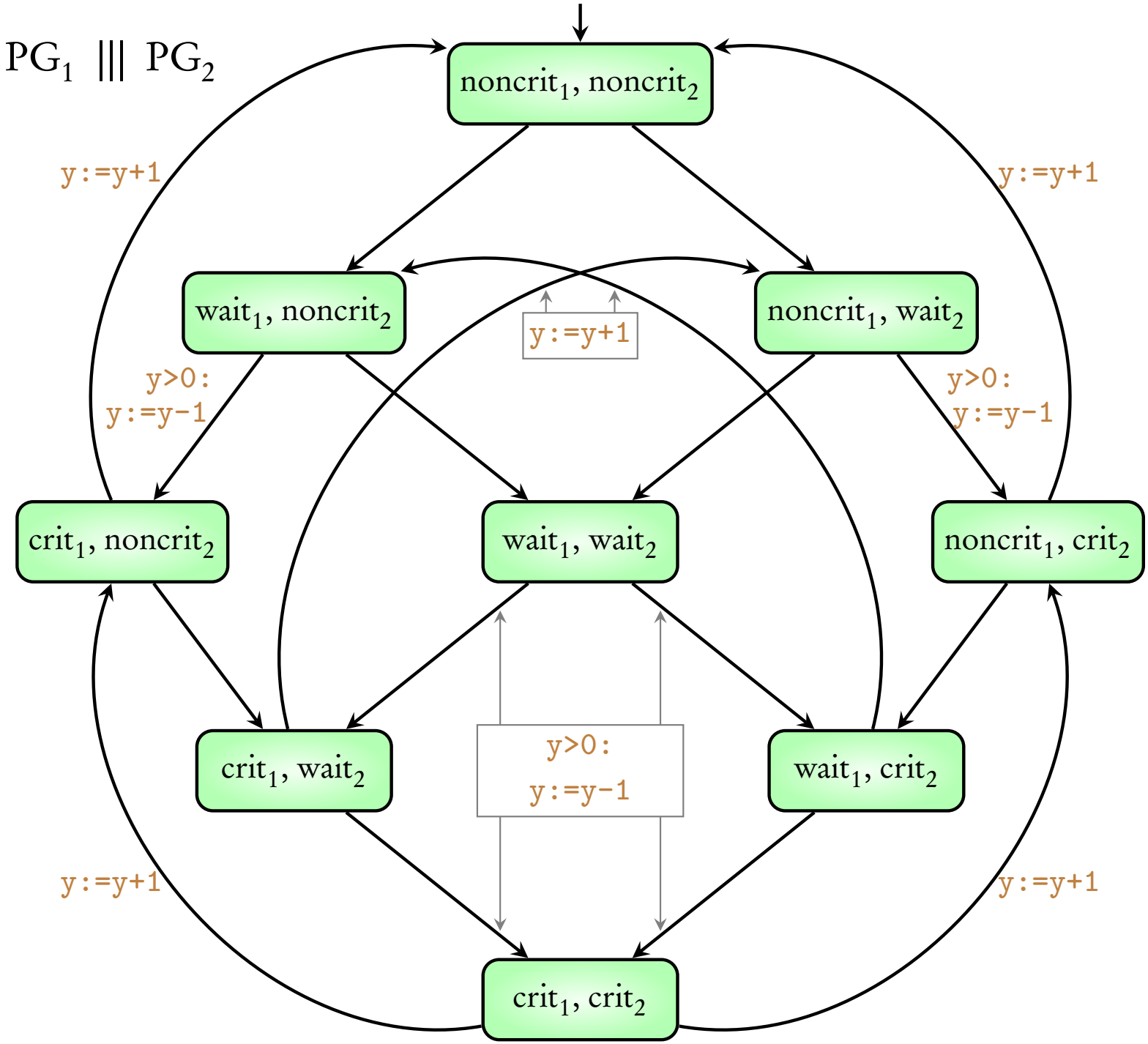
y:=y+1 *release*

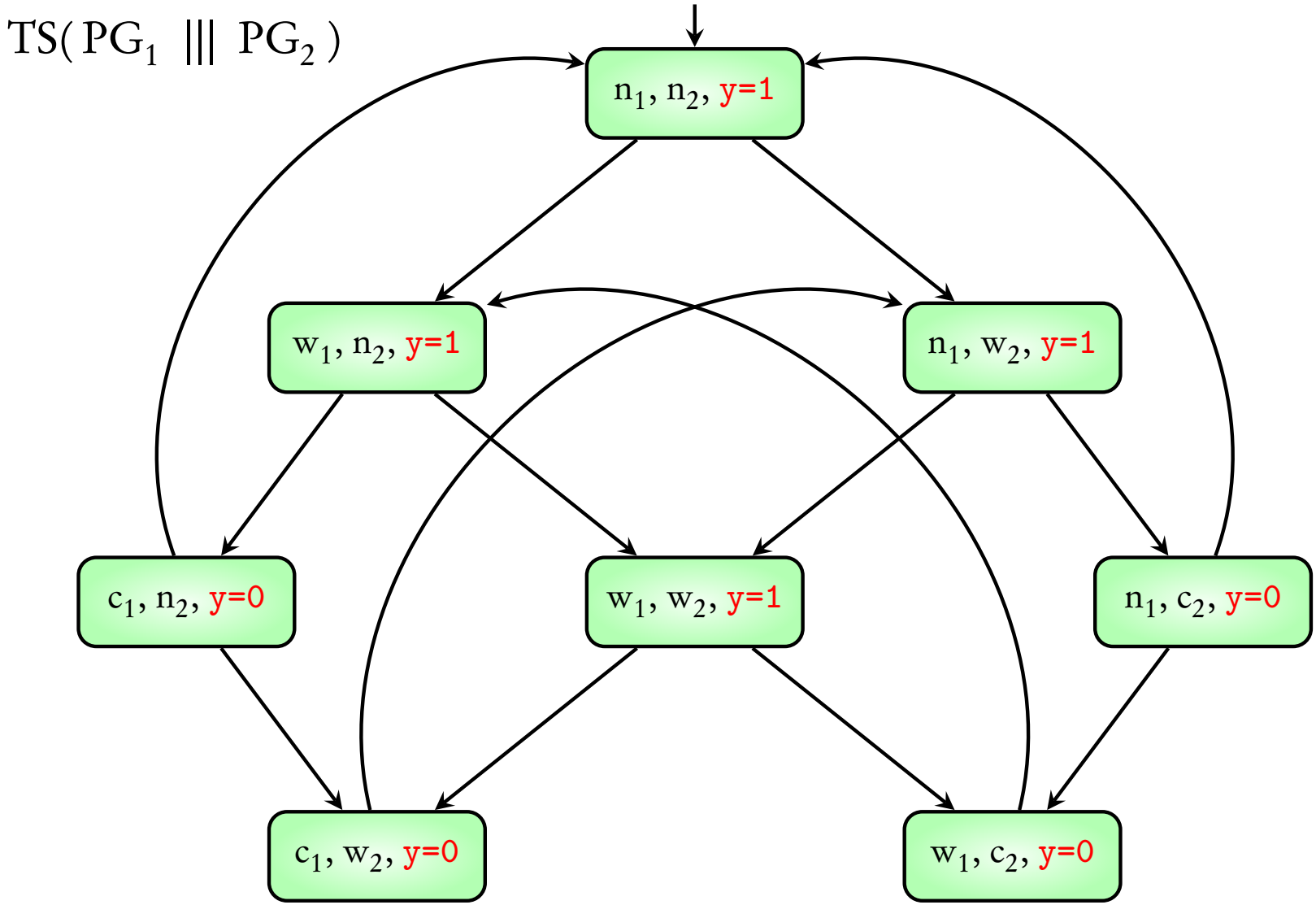
⋮ *non-critical actions*

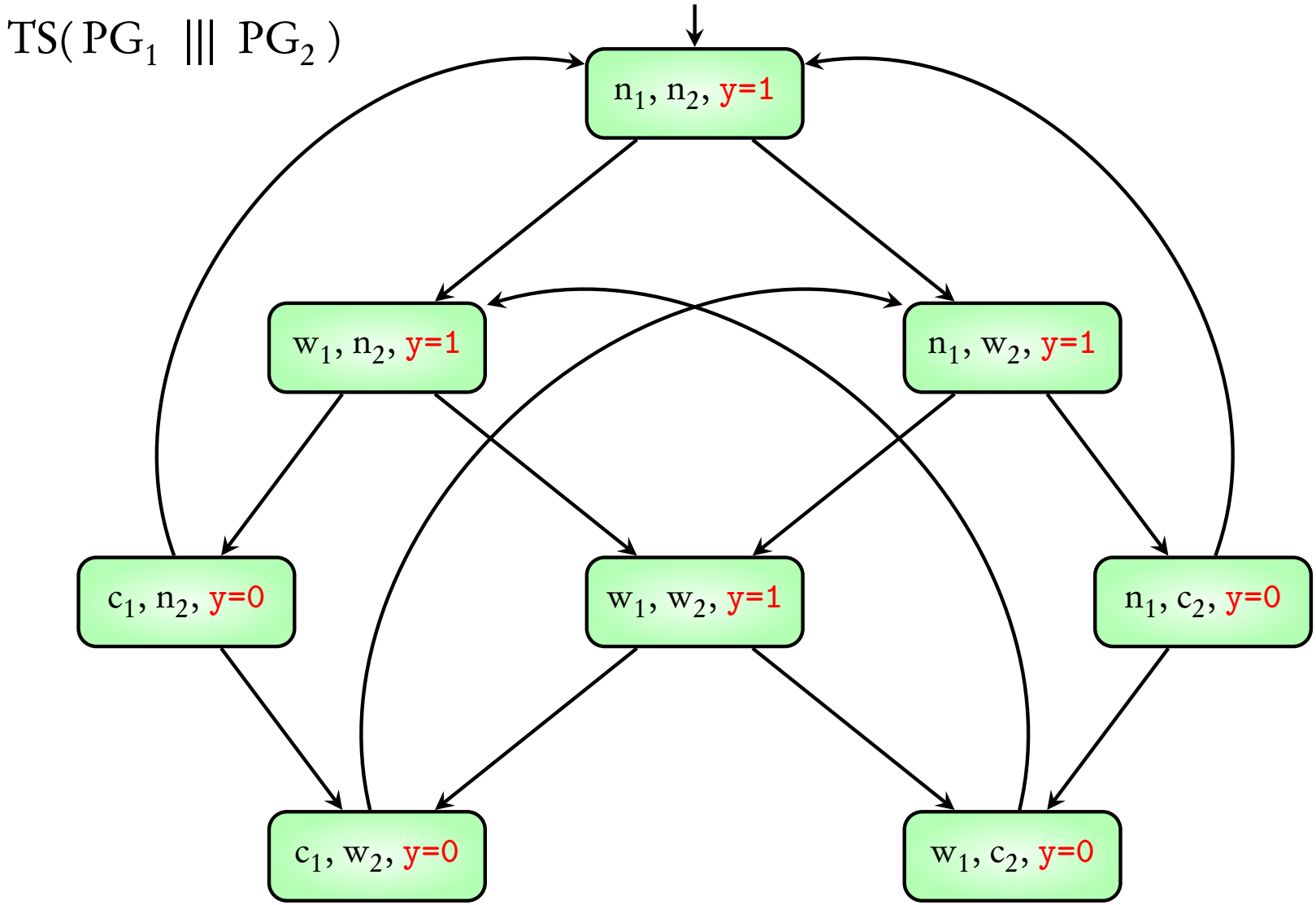
end loop






PG1 ||| PG2
noncrit1, noncrit2
y:=y+1
y:=y+1
wait1, noncrit2
noncrit1, wait2
y:=y+1
y>0:
y:=y-1
y>0:
y:=y-1
crit1, noncrit2
wait1, wait2
noncrit1, crit2
crit1, wait2
y>0:
y:=y-1
wait1, crit2
y:=y+1
y:=y+1
crit1, crit2


TS(PG1 ||| PG2)
n1, n2, y=1
w1, n2, y=1
n1, w2, y=1
c1, n2, y=0
w1, w2, y=1
n1, c2, y=0
c1, w2, y=0
w1, c2, y=0



Both processes **cannot be** in critical section **simultaneously**

Concurrent systems

Independent

Interleaving

$TS_1 \;|||\; TS_2 \;|||\; \ldots \;|||\; TS_n$

Shared variables

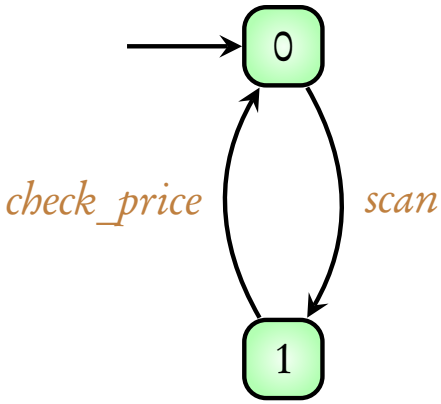
$TS(\,PG_1 \;|||\; PG_2 \;|||\; \ldots \;|||\; PG_n\,)$

Mutual Exclusion

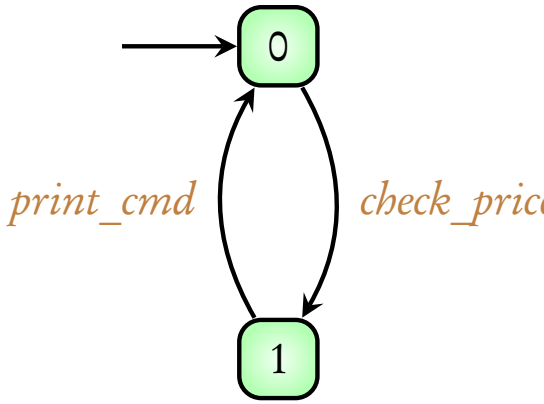
Shared actions

Coming next: **Book-keeping** system in a supermarket

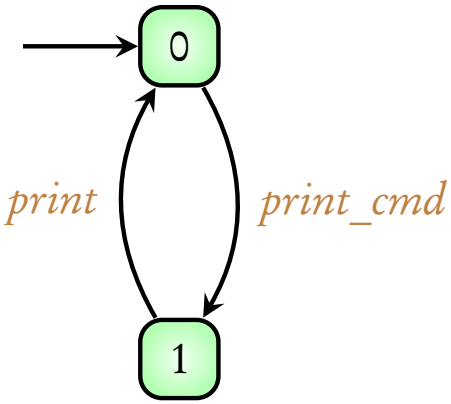
Bar-Code Reader (BCR)



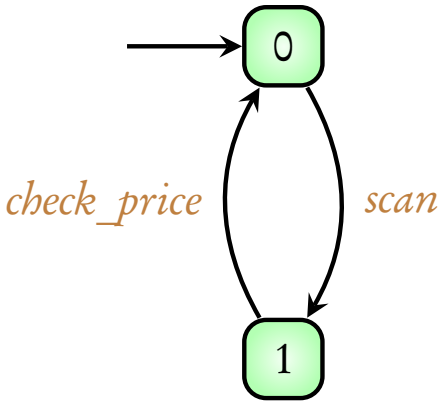
Booking Program (BP)



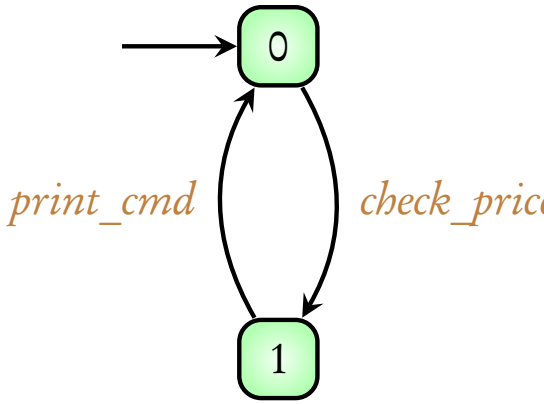
Printer (P)



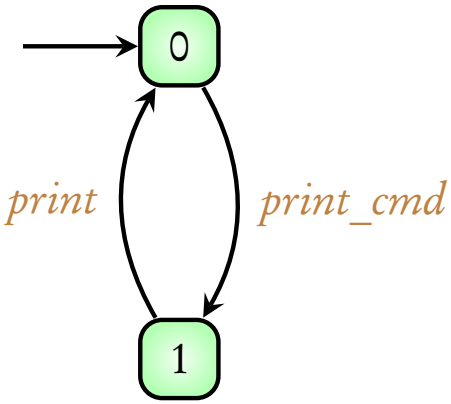
Bar-Code Reader (BCR)

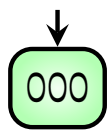


Booking Program (BP)

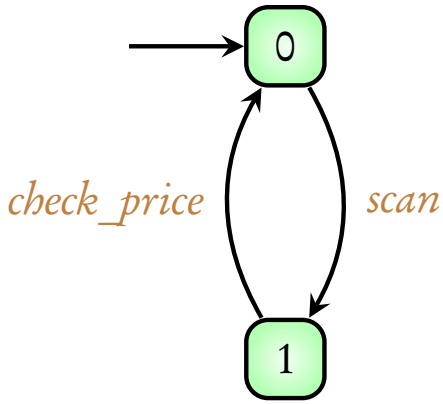


Printer (P)

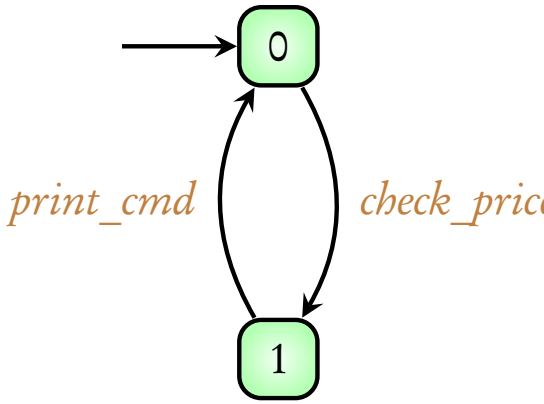




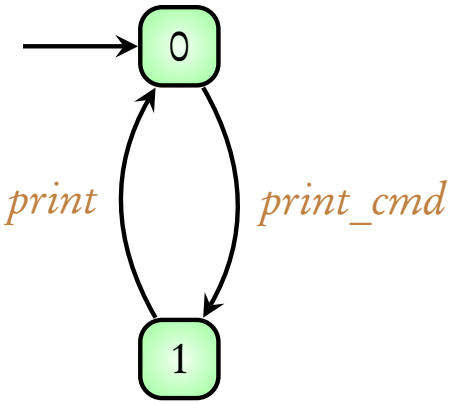
Bar-Code Reader (BCR)



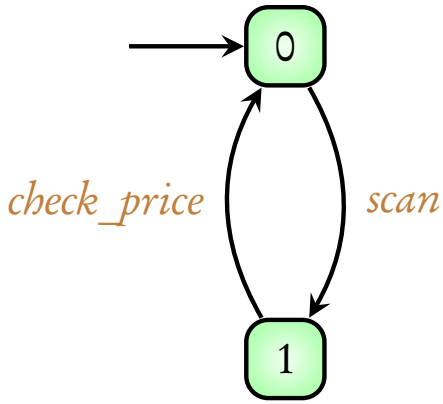
Booking Program (BP)



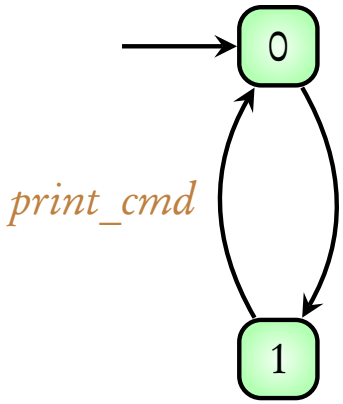
Printer (P)



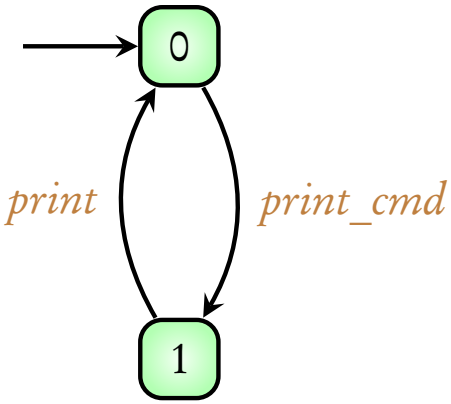
Bar-Code Reader (BCR)

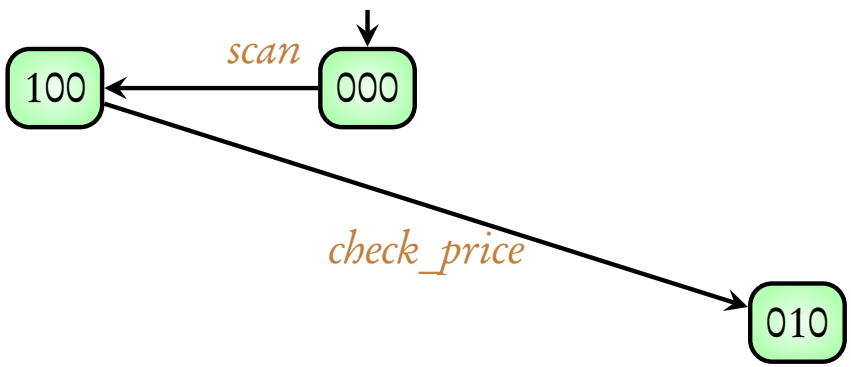


Booking Program (BP)

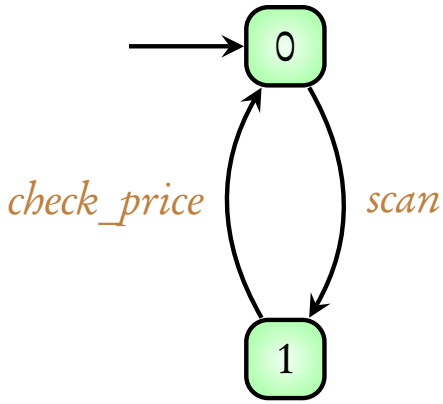


Printer (P)

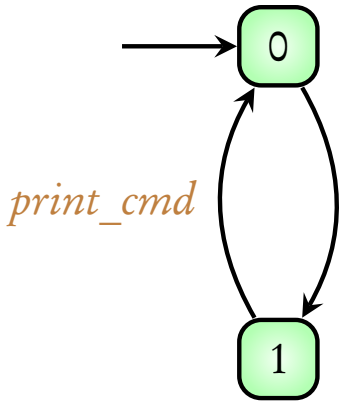




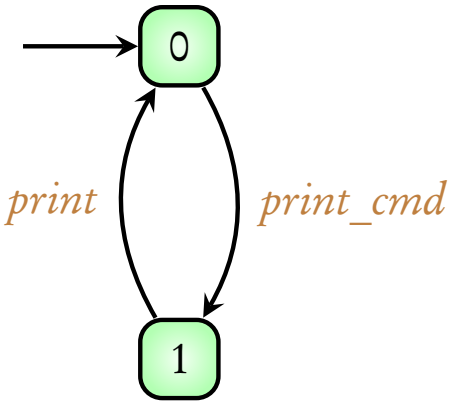
Bar-Code Reader (BCR)

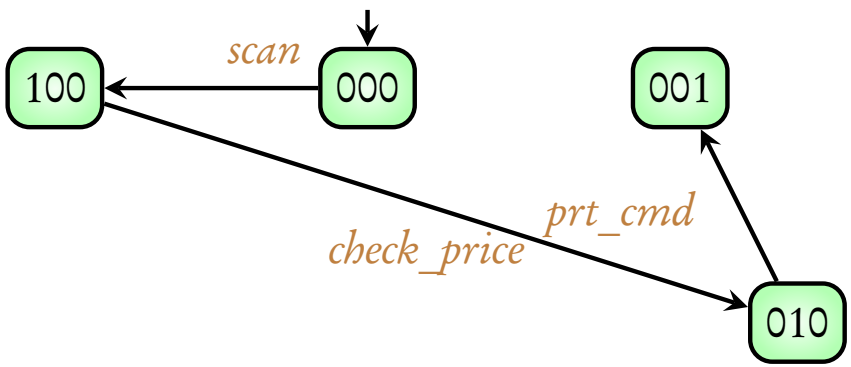


Booking Program (BP)

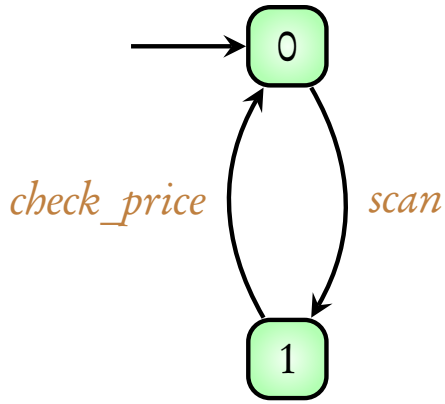


Printer (P)

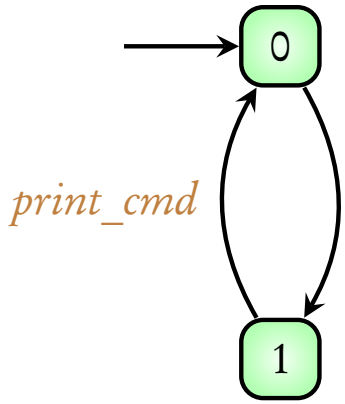




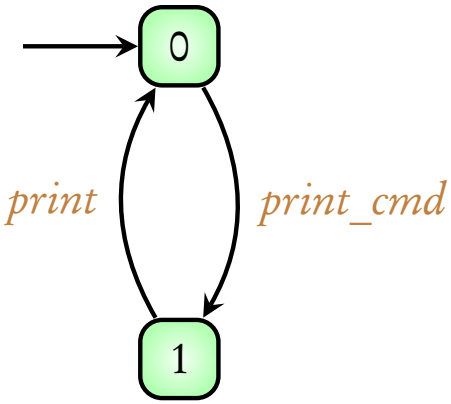
Bar-Code Reader (BCR)

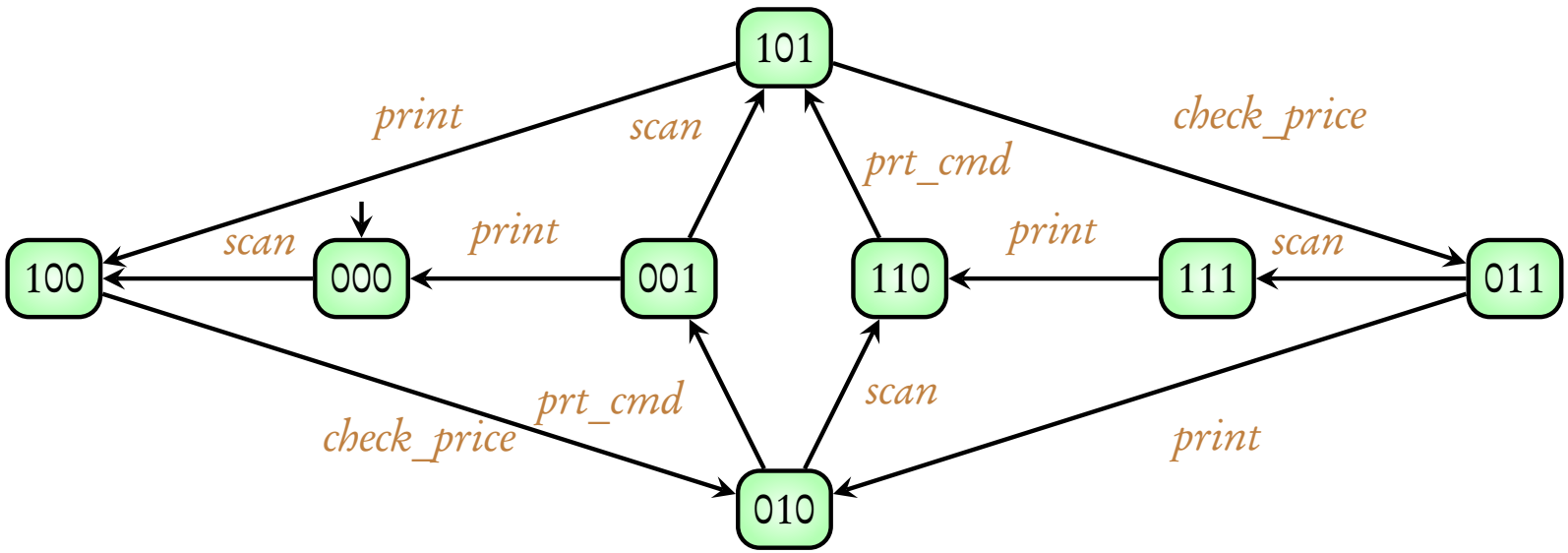


Booking Program (BP)

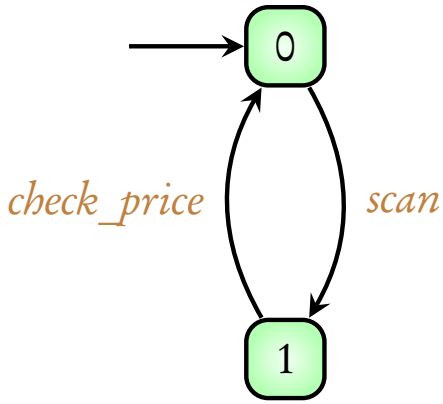


Printer (P)

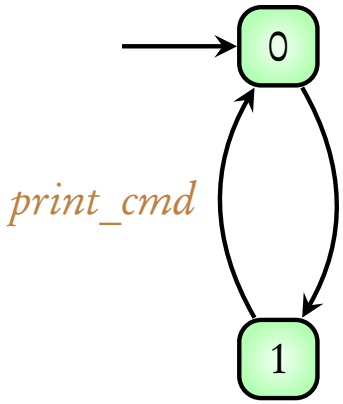




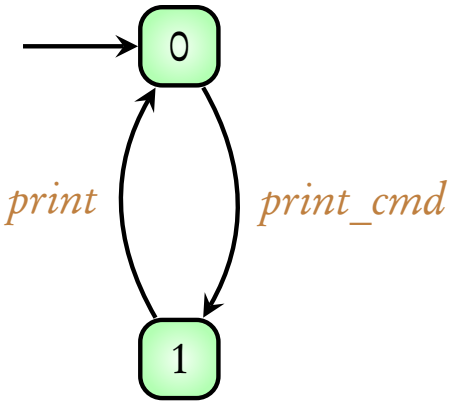
Bar-Code Reader (BCR)



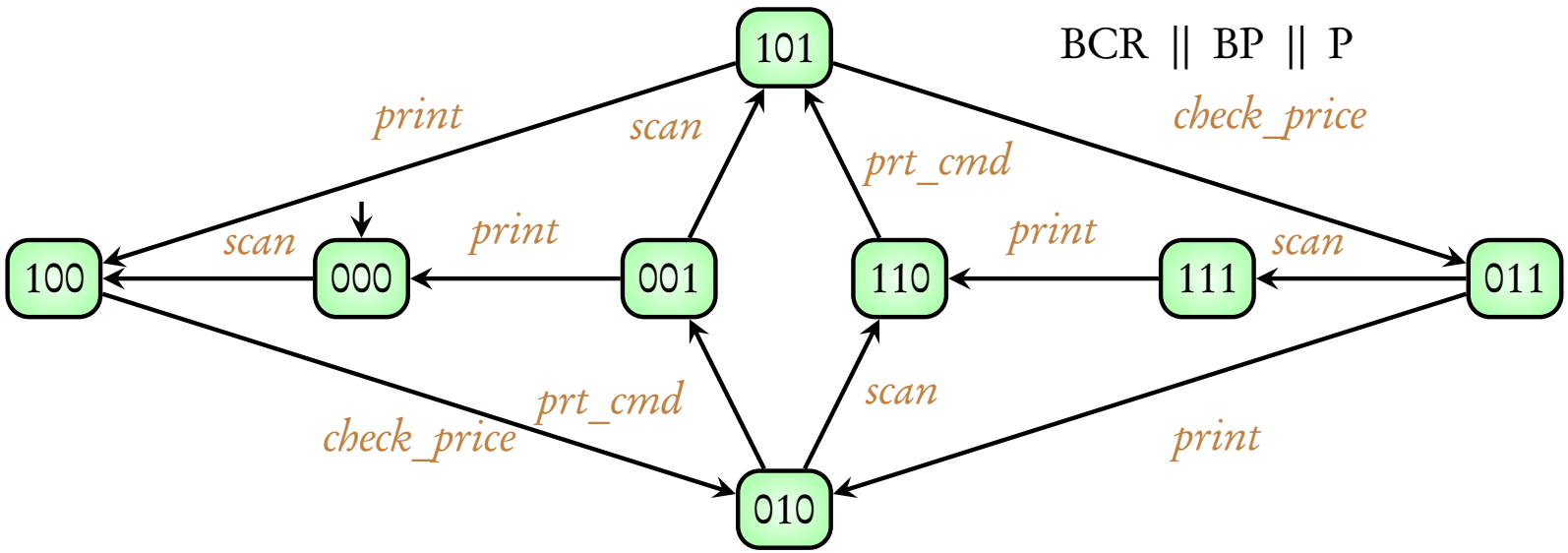
Booking Program (BP)



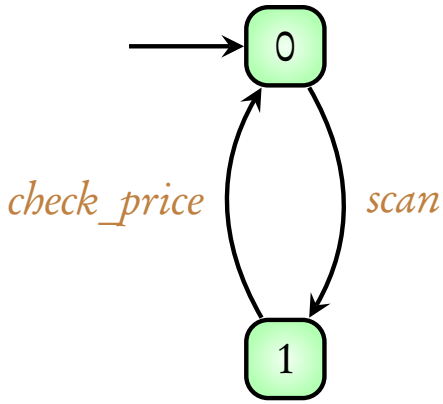
Printer (P)



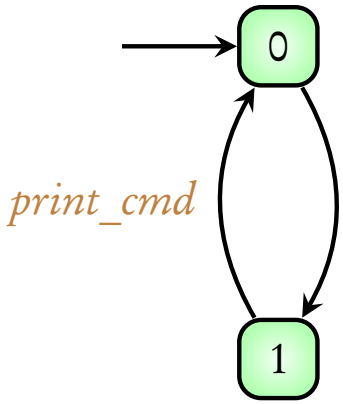
BCR || BP || P



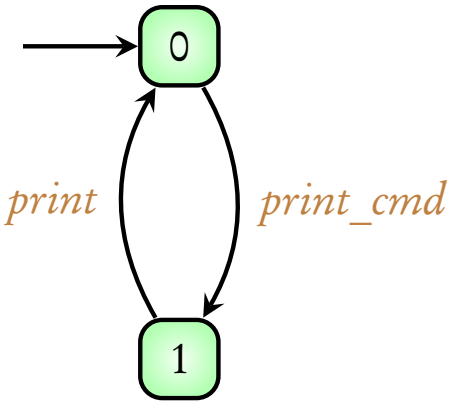
Bar-Code Reader (BCR)



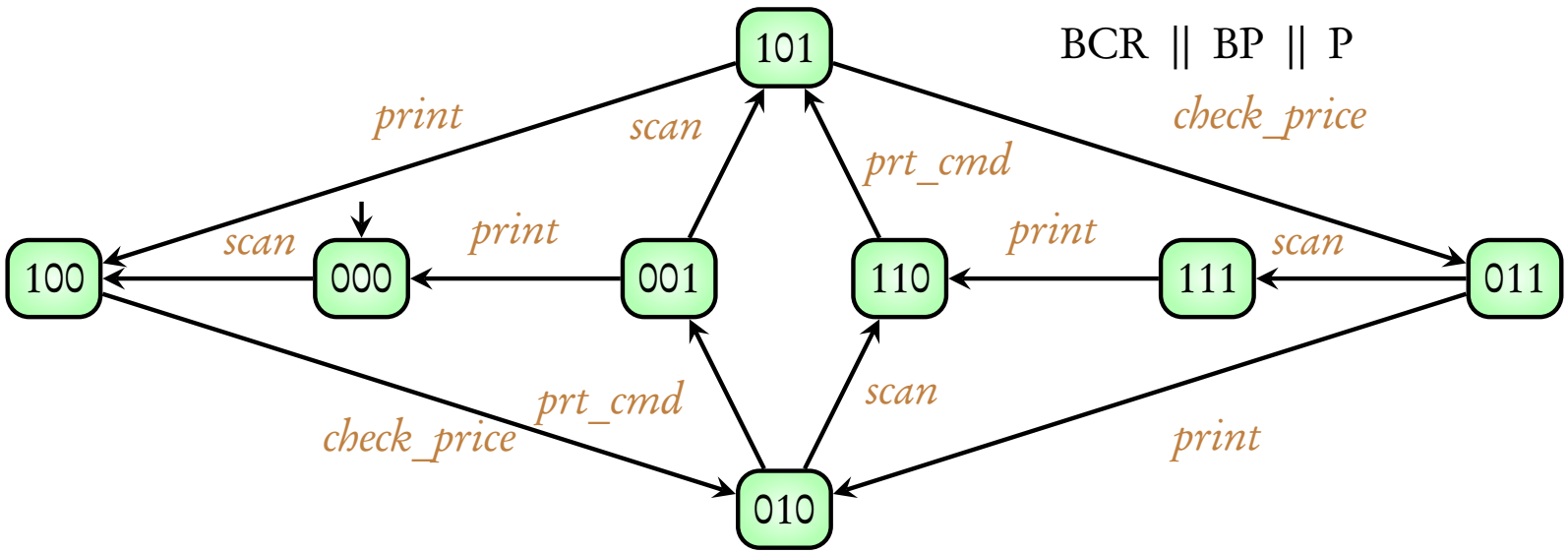
Booking Program (BP)



Printer (P)

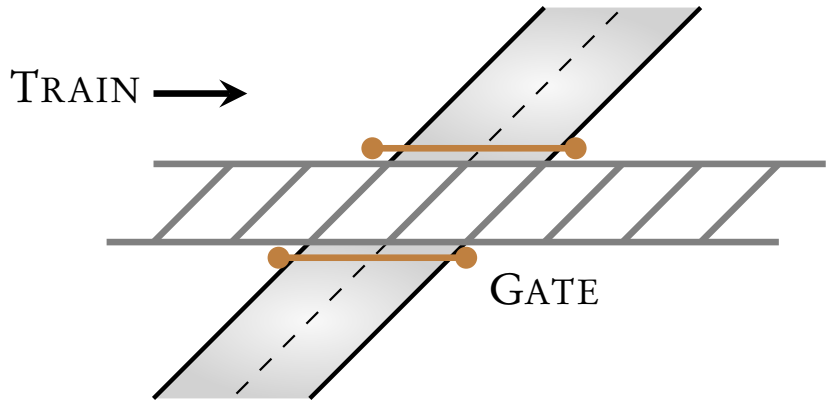


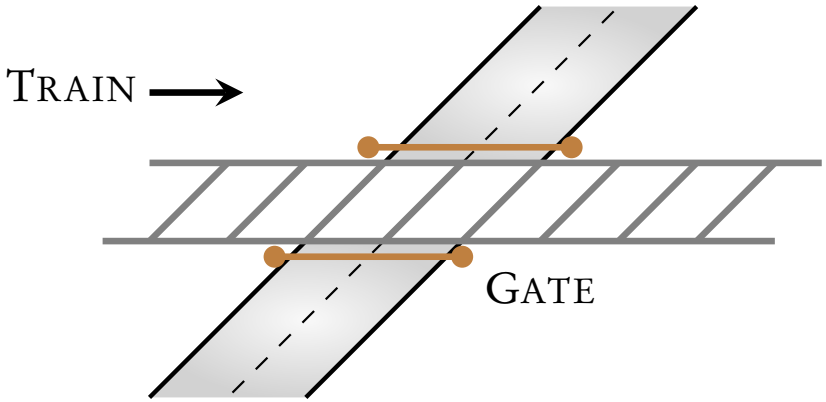
BCR || BP || P

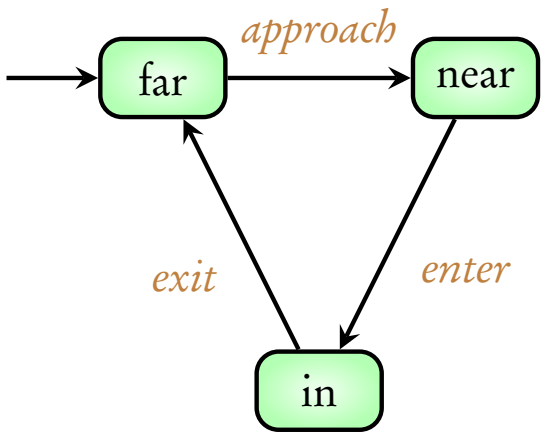


check_price, print_cmd: **Shared** actions (*also called* **handshaking** actions)

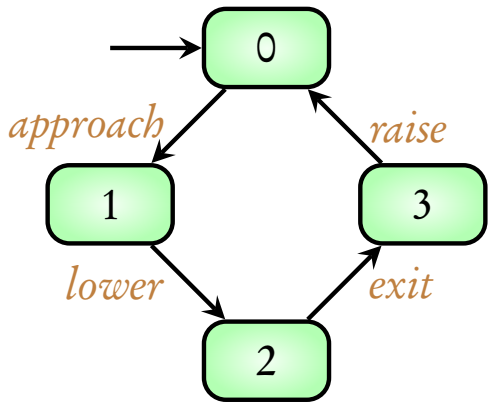
Next example: Train-Gate-Controller


Train
Gate

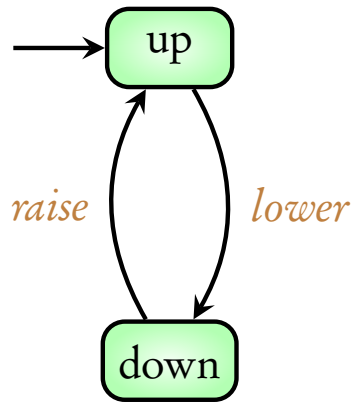




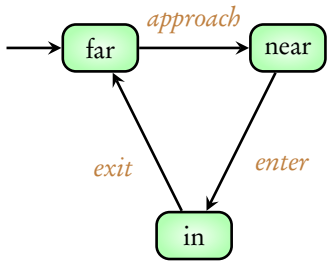
Train



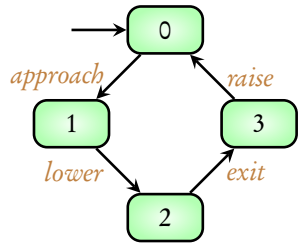
Controller



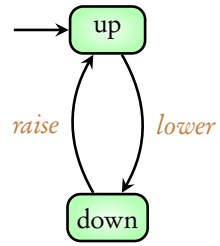
Gate



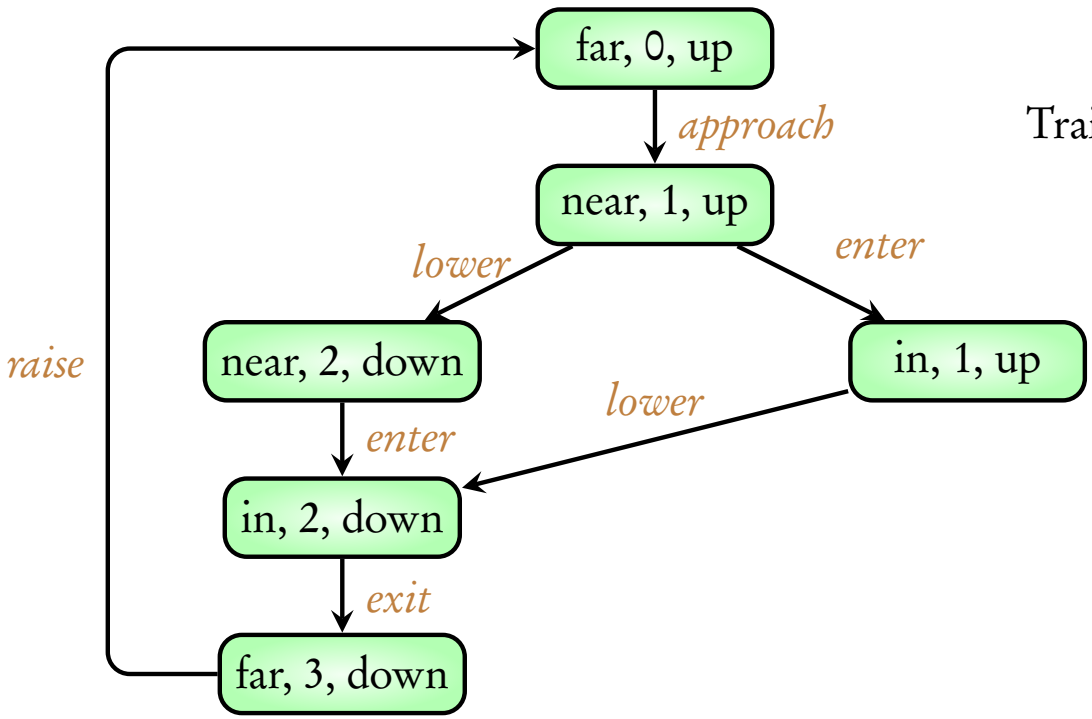
Train



Controller



Gate



Train || Controller || Gate

|| : **Handshake** operator

Independent

Interleaving

$TS_1 \;|||\; TS_2 \;|||\; \ldots \;|||\; TS_n$

Shared variables

$TS(\,PG_1 \;|||\; PG_2 \;|||\; \ldots \;|||\; PG_n\,)$

Mutual Exclusion

Shared actions

$TS_1 \;||\; TS_2$

Reference: Principles of Model Checking, *Baier and Katoen*, MIT Press (2008)
Pages 35 - 53