Verification of Concurrent Programs under Release Acquire

S. Krishna

IIT Bombay

IARCS Verification Seminar Series
Dec 06, 2022

Concurrency

Concurrency

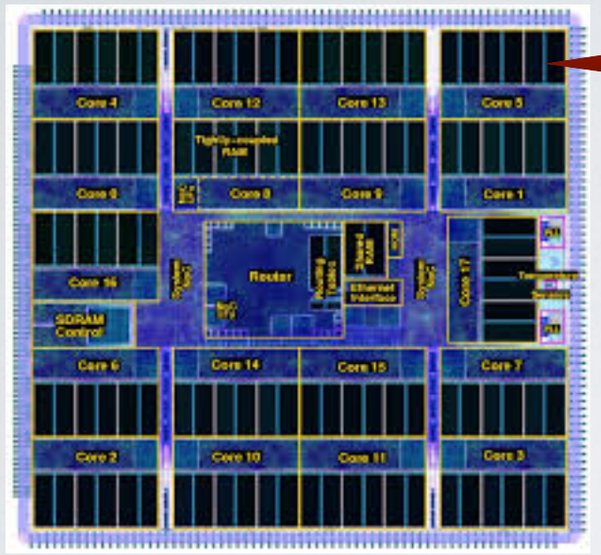
motivation

Concurrency

motivation

Concurrent systems are everywhere

Concurrency



Multicore architectures

motivation

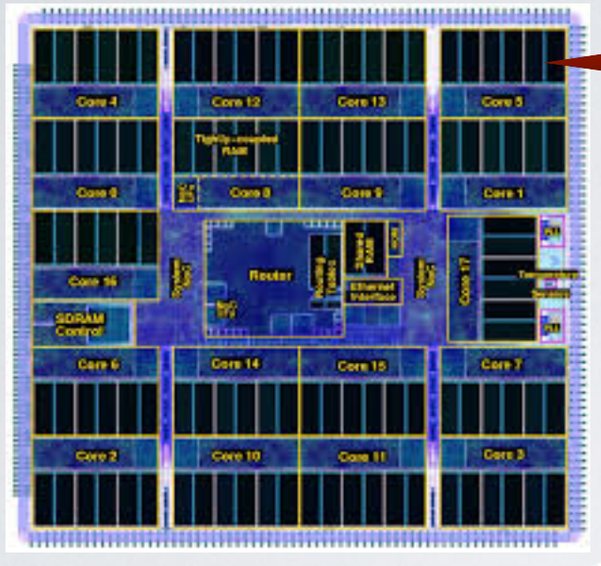
Concurrent systems are everywhere

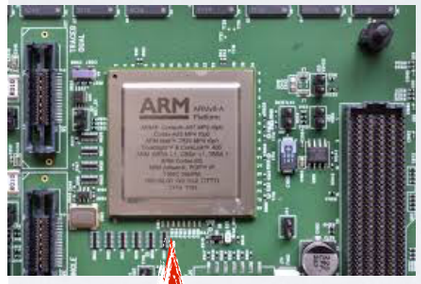
Concurrency

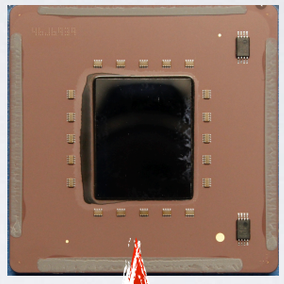
Multicore architectures

motivation

Concurrent systems are everywhere







intel

ARM

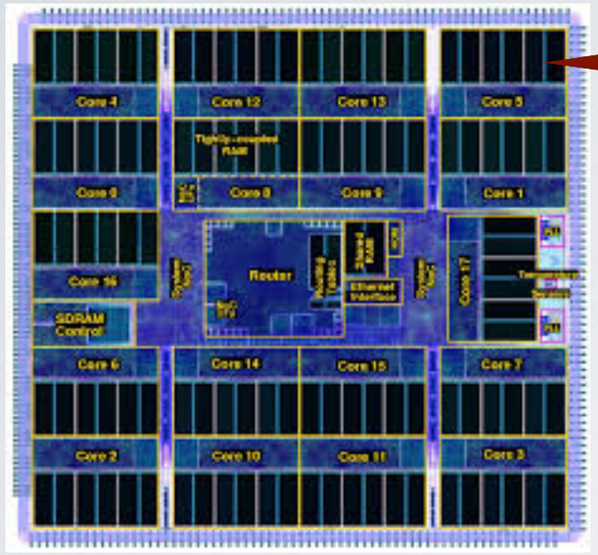
IBM Power

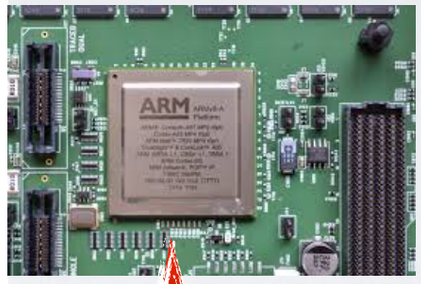
Concurrency

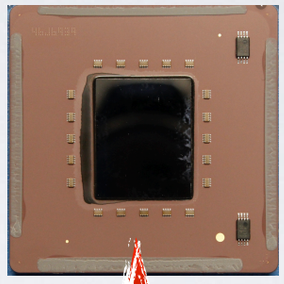
Concurrent systems are everywhere

Multicore architectures

motivation







intel

ARM

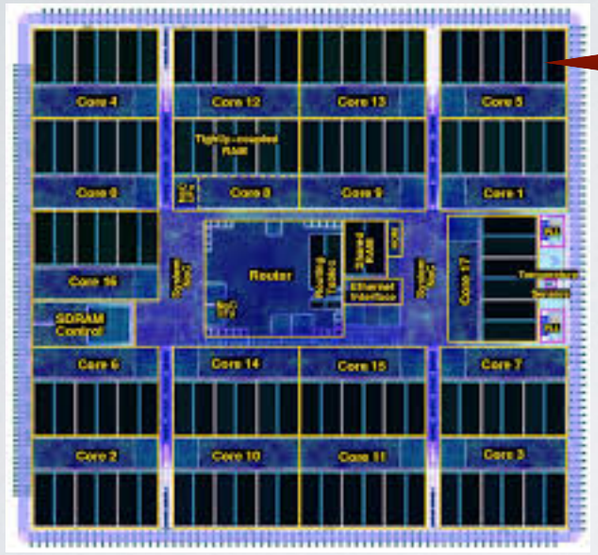
IBM Power

Distributed databases

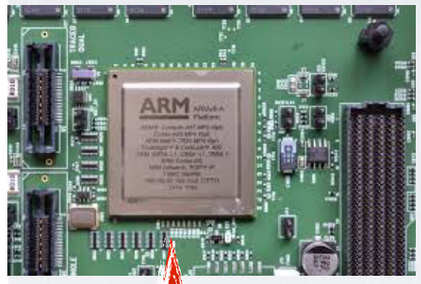
Concurrency

Concurrent systems are everywhere

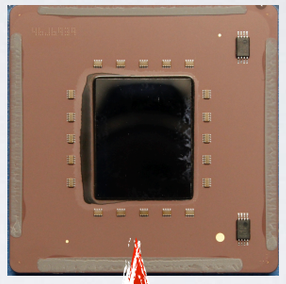
Multicore architectures — **motivation**



intel



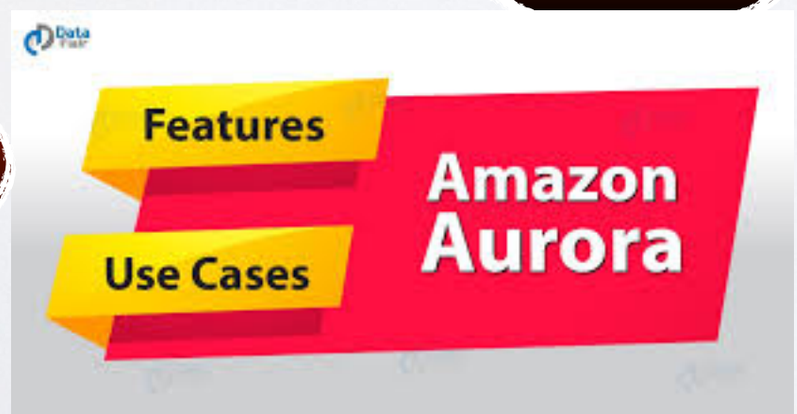
ARM



IBM Power

Distributed databases

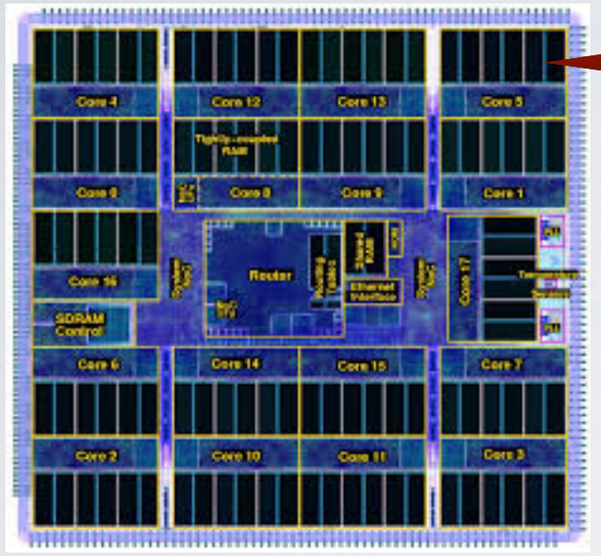
Facebook



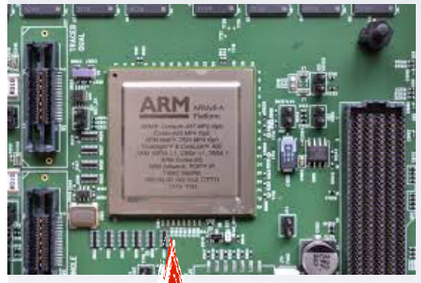
Concurrency

Concurrent systems are everywhere

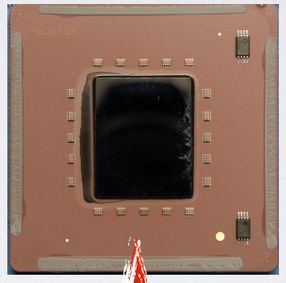
Multicore architectures — **motivation**



intel



ARM

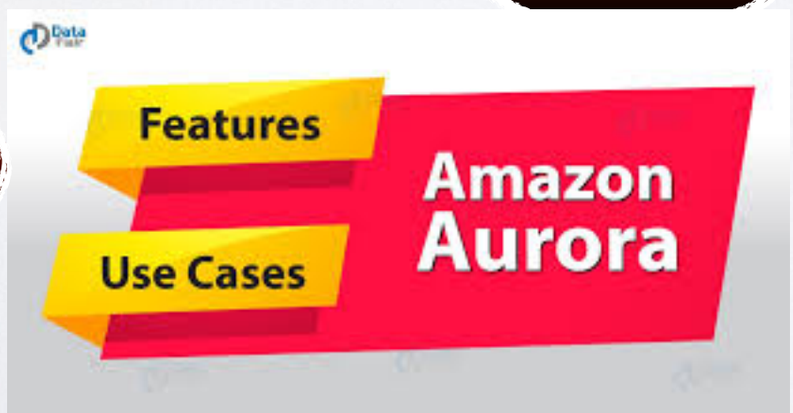


IBM Power

Distributed databases

Facebook

Programming languages



Weak memory models provide sound semantics for high performance concurrency

Weak memory models provide sound semantics for high performance concurrency

Weak memory models provide sound semantics for high performance concurrency

Interleaving semantics

Different threads observe memory events in different orders

Sequential Consistency

Release Acquire (RA)

Relaxed models

Easy Programmability

High Performance

Fragment of C11

The C11 Memory Model

The C11 Memory Model

Non-atomic Data accesses

The C11 Memory Model

Non-atomic Data accesses

Atomic Data accesses

Performance

Synchronization

The C11 Memory Model

Non-atomic Data accesses

Atomic Data accesses

Relaxed

Release-Acquire

Sequential Consistency

Performance

Synchronization

The C11 Memory Model

Non-atomic Data accesses

Atomic Data accesses

Relaxed

Release-Acquire

Sequential Consistency

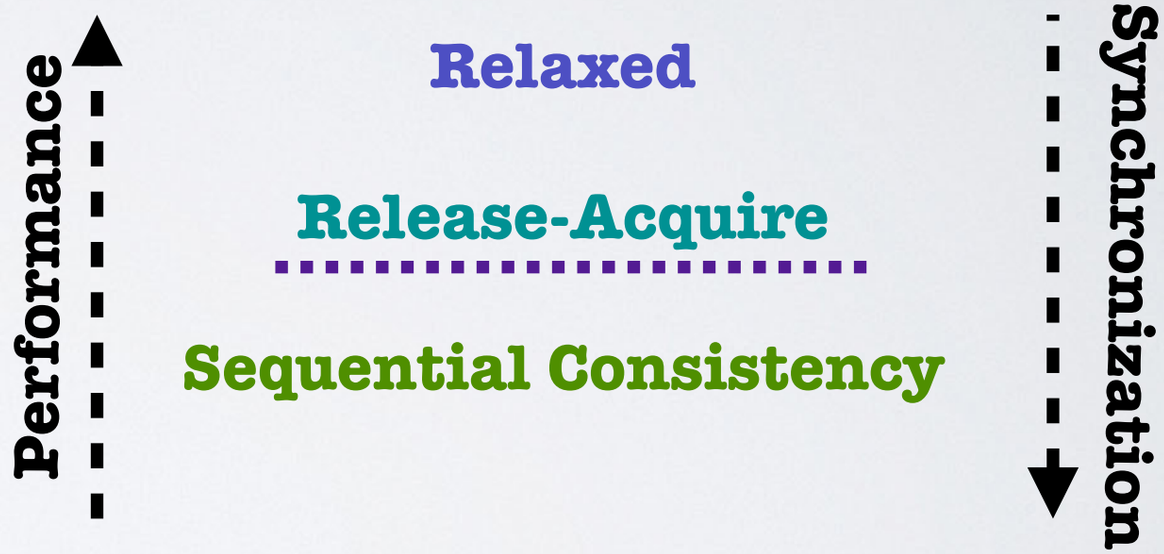
Performance

Synchronization

The C11 Memory Model

Non-atomic Data accesses

Atomic Data accesses



Release-Acquire : all reads are **acquire**, all writes are **release**, updates (atomic read writes) are **acquire/release**

Store buffer

Init: x=y=0

```
1. x=1;
2. ry=y;
```

Process 1

```
1. y=1;
2. rx=x;
```

Process 2

RA execution

x=0 y=0

Shared memory

Specification S: not (rx=0 && ry=0)

Store buffer

Init: x=y=0

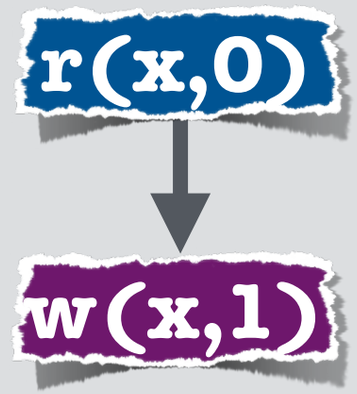
```
1. x=1;
2. ry=y;
```

Process 1

```
1. rx=x;
2. y=1;
```

Process 2

RA execution



x=0 y=0

Shared memory

Specification S: not (rx=0 && ry=0)

Store buffer

Init: x=y=0

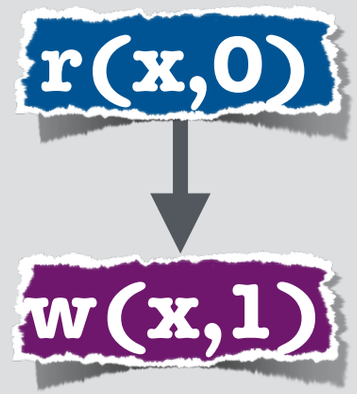
```
1. x=1;
2. ry=y;
```

Process 1

```
1.rx=x;
2. y=1;
```

Process 2

RA execution



x=1 y=0

Shared memory

Specification S: not (rx=0 && ry=0)

Store buffer

Init: x=y=0

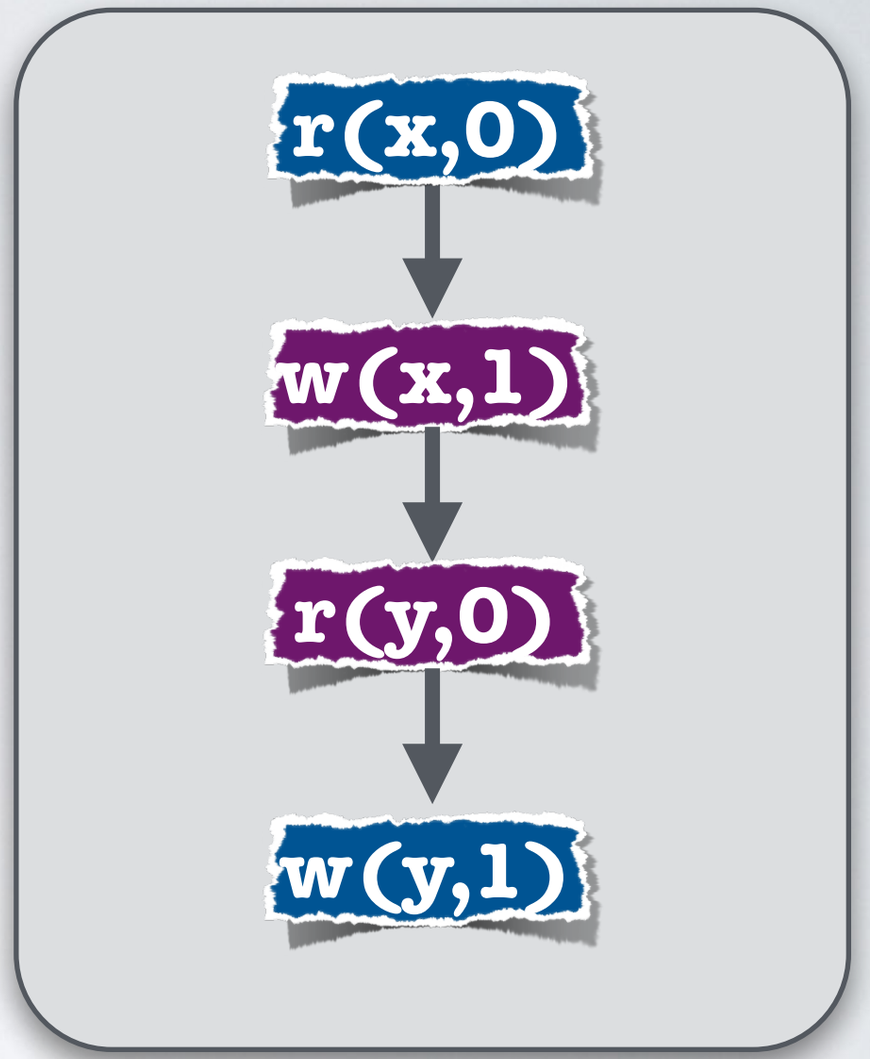
```
1. x=1;
2. ry=y;
```

Process 1

```
1.rx=x;
2. y=1;
```

Process 2

RA execution



x=1 y=0

Shared memory

Specification S: not (rx=0 && ry=0)

Store buffer

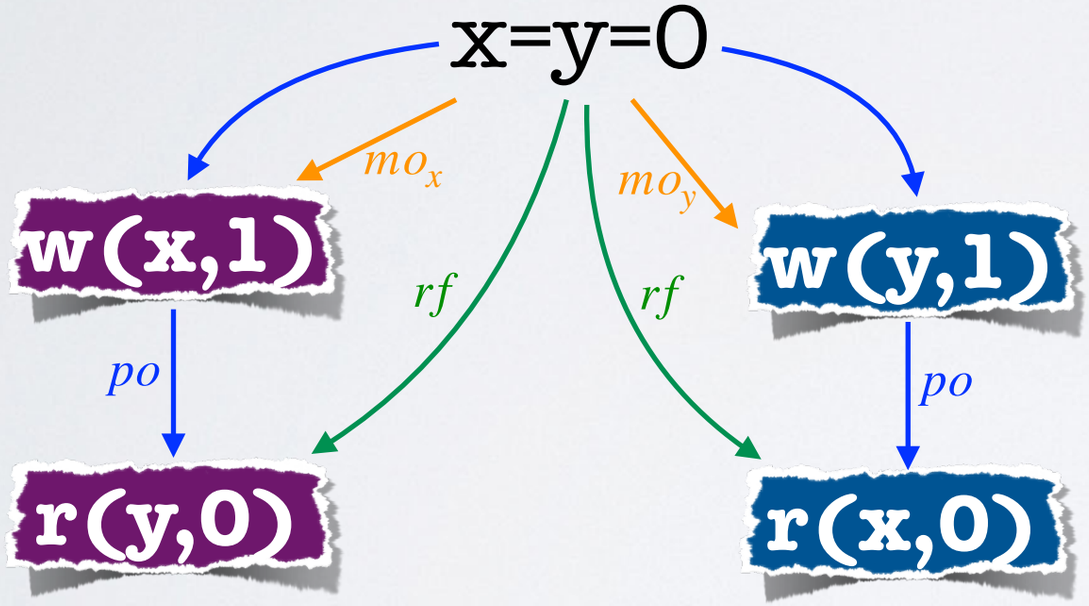
Init: x=y=0

```
1. x=1;
2. ry=y;
```

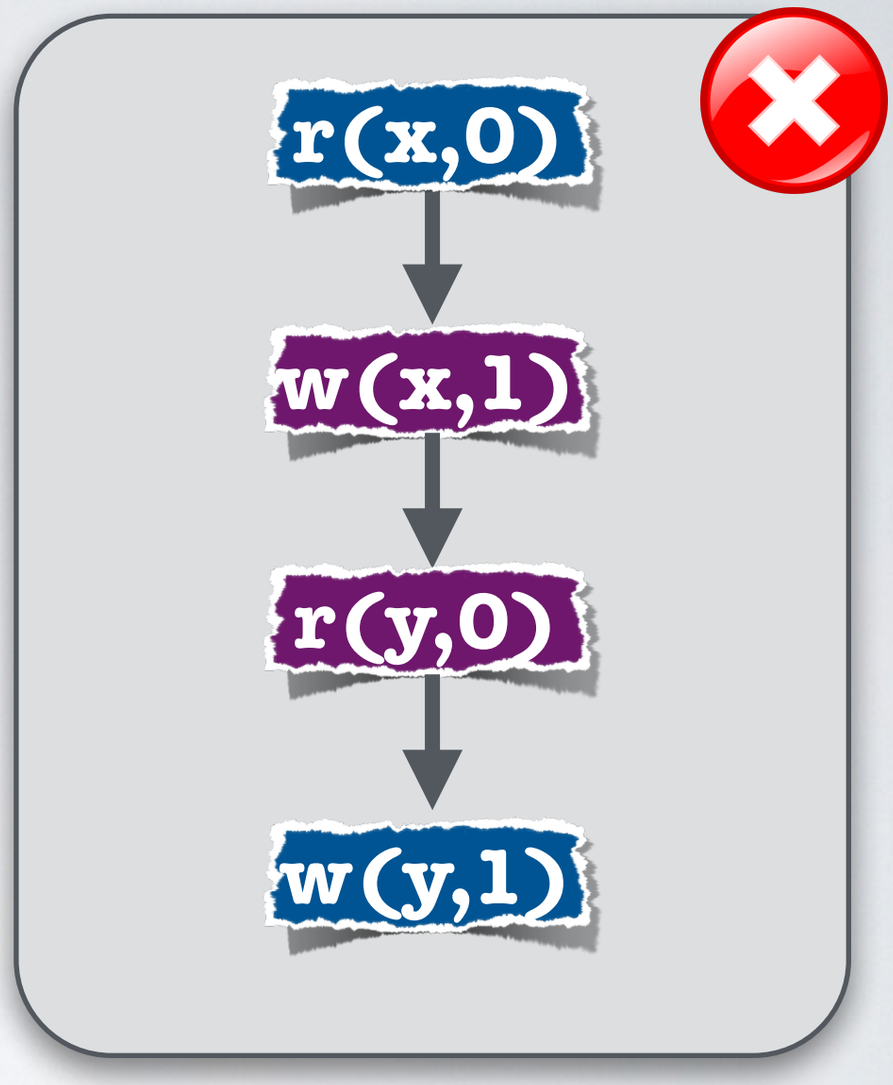
Process 1

```
1.rx=x;
2. y=1;
```

Process 2



RA execution

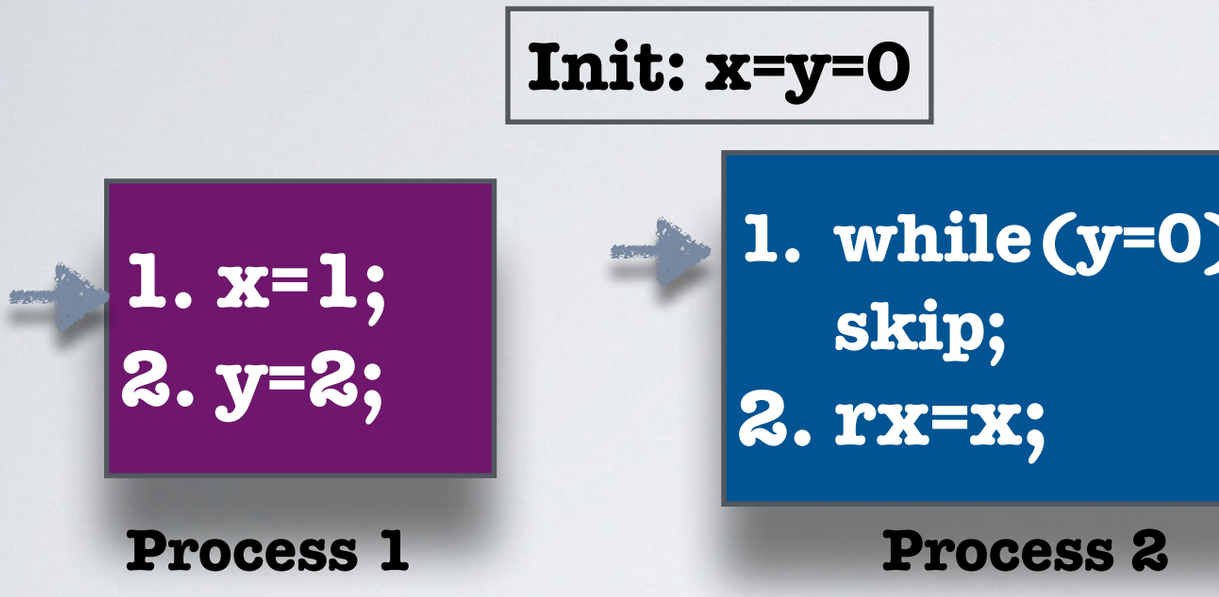


x=1 y=1

Shared memory

Specification S: not (rx=0 && ry=0)

Message passing



Specification S: not (rx=0)

No weak behaviours

Message passing

Init: x=y=0

```
1. x=1;
2. y=2;
```

Process 1

```
1. while(y=0)
     skip;
2. rx=x;
```

Process 2

Specification S: not (rx=0)

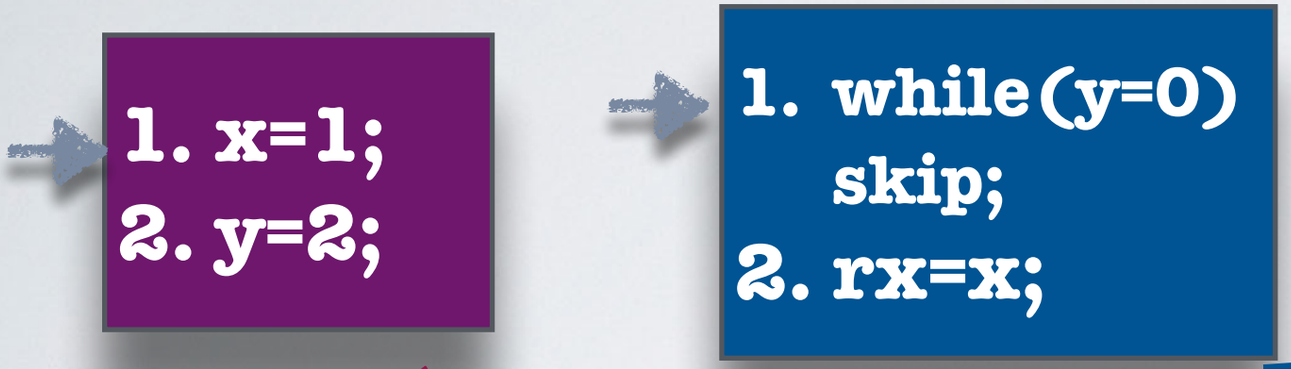
Reordering writes not allowed in RA

No weak behaviours

Message passing

Init: x=y=0

Specification S: not (rx=0)



No weak behaviours

Message passing

Init: x=y=0

```
1. x=1;
2. y=2;
```

Process 1

```
1. while(y=0)
     skip;
2. rx=x;
```

Process 2

Specification S: not (rx=0)

Reordering writes not allowed in RA

Reordering of reads not allowed in RA

Process 2 is aware of the write of 1 to x when it reads 2 from y

No weak behaviours

Message passing

Init: x=y=0

```
1. x=1;
2. y=2;
```

Process 1

```
1. while(y=0)
     skip;
2. rx=x;
```

Process 2

Specification S: not (rx=0)

Reordering writes not allowed in RA

Reordering of reads not allowed in RA

Process 2 is aware of the write of 1 to x when it reads 2 from y

No weak behaviours

Message passing

Init: x=y=0

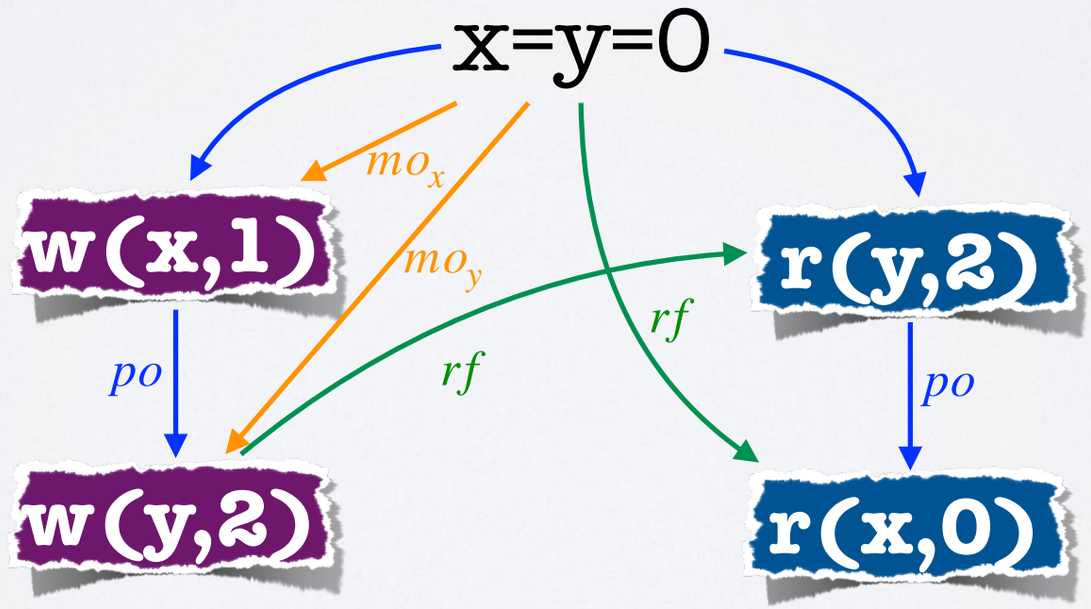
```
1. x=1;
2. y=2;
```

Process 1

```
1. while(y=0)
     skip;
2. rx=x;
```

Process 2

Specification S: not (rx=0)



Message passing

Init: x=y=0

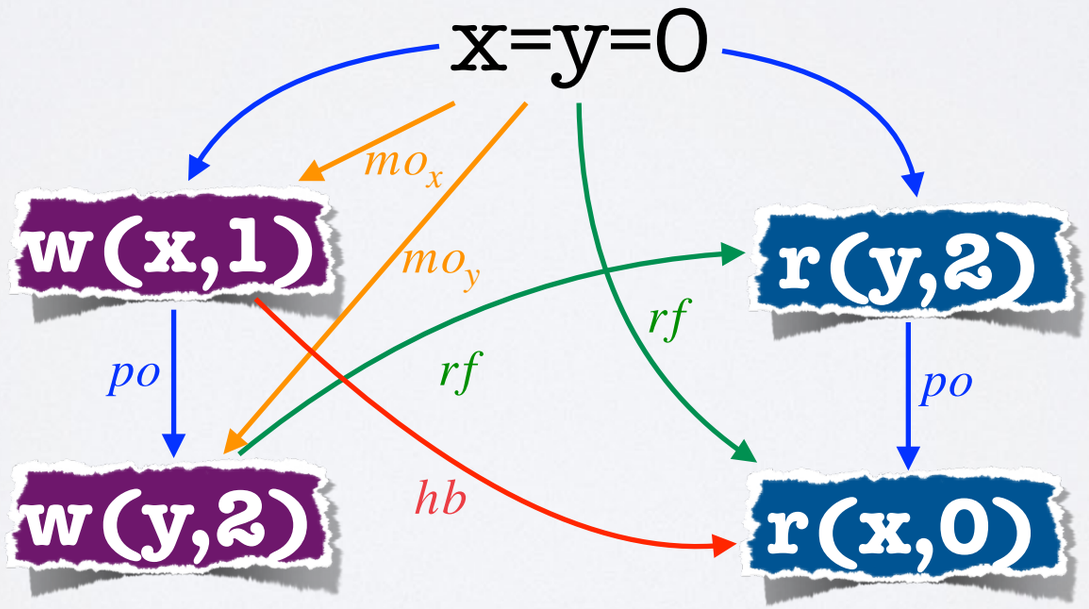
```
1. x=1;
2. y=2;
```

Process 1

```
1. while(y=0)
     skip;
2. rx=x;
```

Process 2

Specification S: not (rx=0)



Declarative Semantics for RA

reads-from (rf) : maps reads to corresponding writes s.t.

happens-before $(hb) = (\ po \cup rf\ )^{+}$ is irreflexive

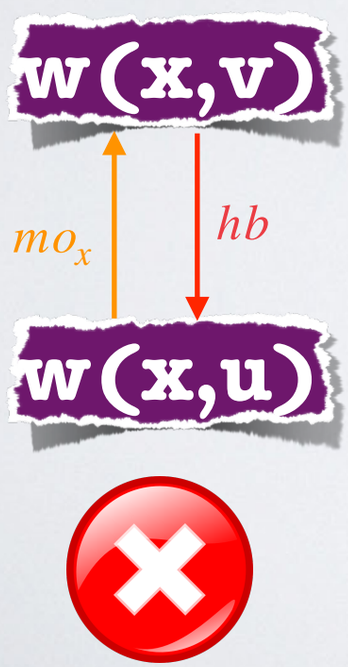
modification-order (mo) : total order on same-location writes s.t.

Declarative Semantics for RA

reads-from (rf) : maps reads to corresponding writes s.t.

happens-before $(hb) = (\ po \cup rf\ )^{+}$ is irreflexive

modification-order (mo) : total order on same-location writes s.t.

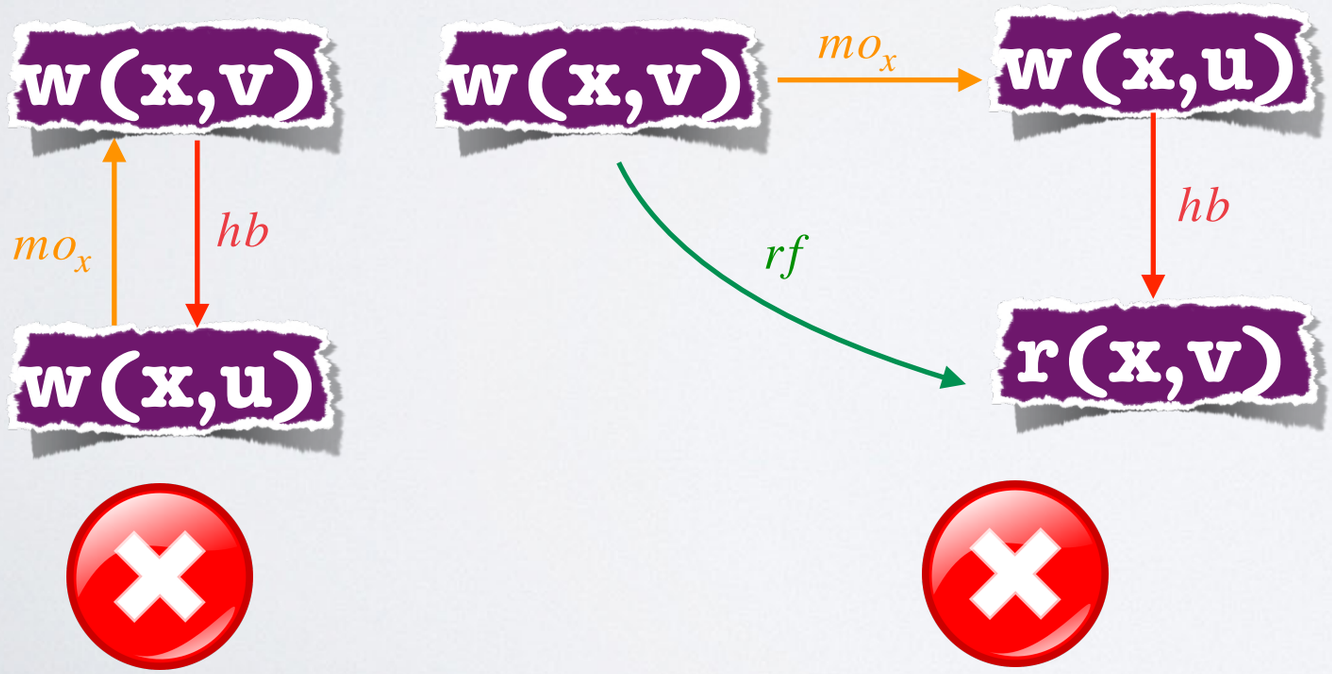


Declarative Semantics for RA

reads-from (rf) : maps reads to corresponding writes s.t.

happens-before $(hb) = (\ po \cup rf\ )^{+}$ is irreflexive

modification-order (mo) : total order on same-location writes s.t.

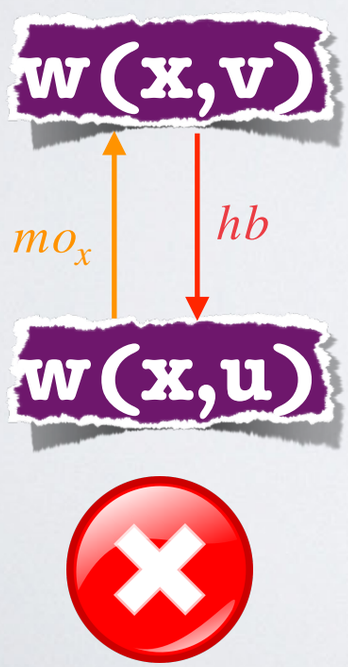


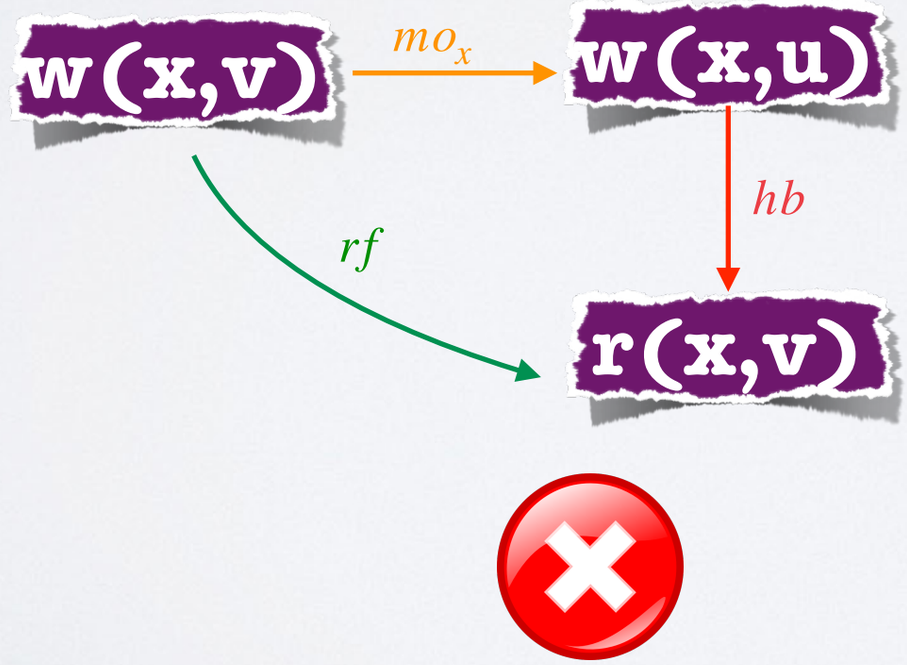
Declarative Semantics for RA

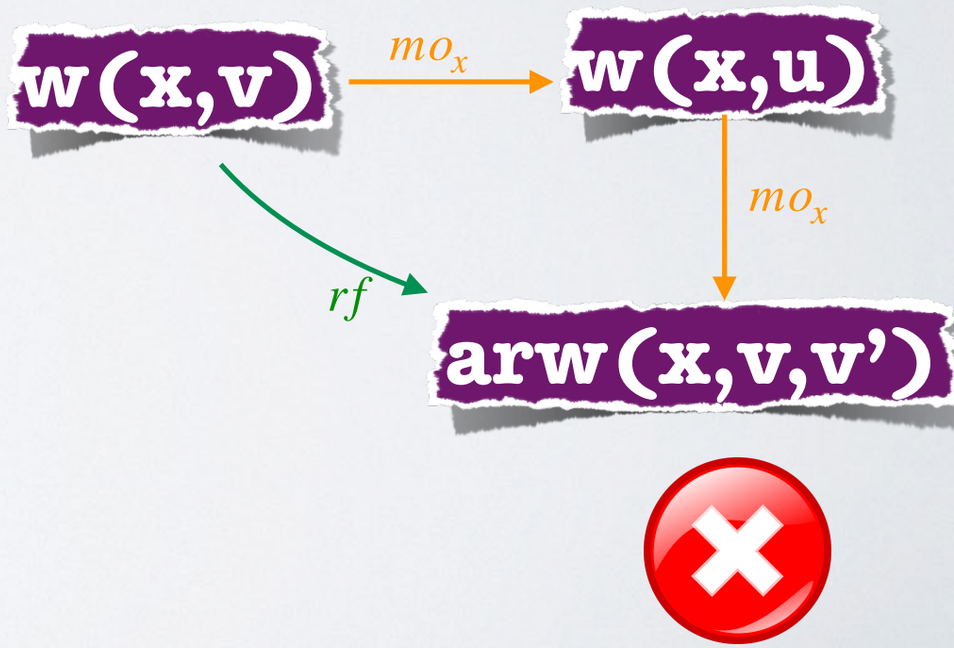
reads-from (rf) : maps reads to corresponding writes s.t.

happens-before $(hb) = (\ po \cup rf\ )^{+}$ is irreflexive

modification-order (mo) : total order on same-location writes s.t.







Release-Acquire (RA)

Verified compilation schemes to TSO and Power

Batty et al'11
Batty et al'12
Sarkar et al'12

Supports optimizations Wx –> Ry to Ry–> Wx

SC does not!

Many program logics

RSL, GPS,OGRA

Absence of data races in SC implies no weak behaviours

Not true for full C11!

Intuitive semantics

Operational Model for RA

[J. Kang et al. POPL 2017, A. Podkopaev et al. 2016, Arxiv]

Init: x=y=0

```
1. x=1;
2. ry=y;
```

Process 1

```
1. y=1;
2. rx=x;
```

Process 2

rx=0 and ry=0?

Init: x=y=0

```
1. x=1;
2. y=2;
```

Process 1

```
1. while(y=0)
     skip;
2. rx=x;
```

Process 2

rx=1?

Store buffer

Init: x=y=0

Process 1	Process 2
1. x=1;	1. y=1;
2. ry=y;	2. rx=x;

rx=0 and ry=0?

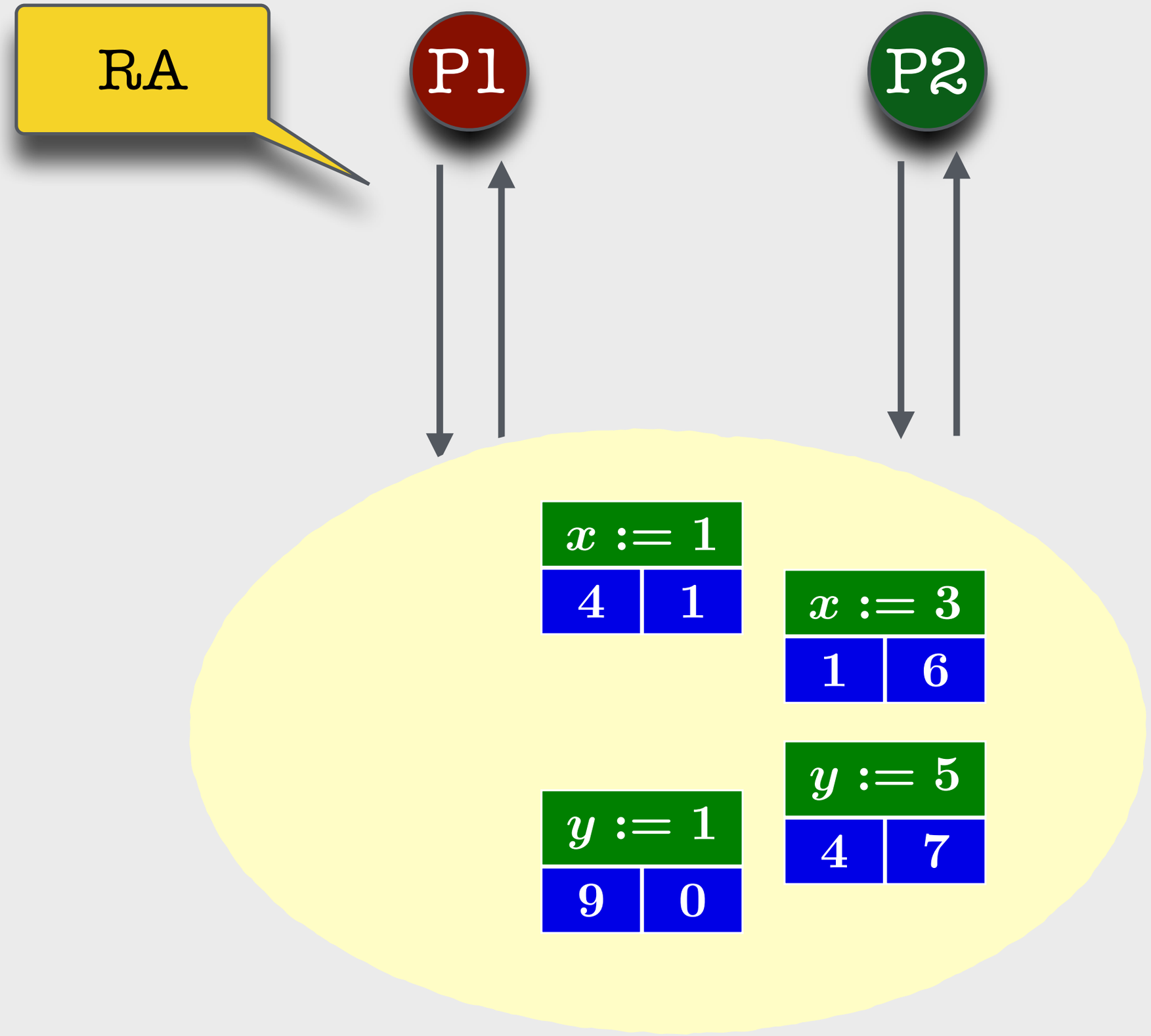
Message passing

Init: x=y=0

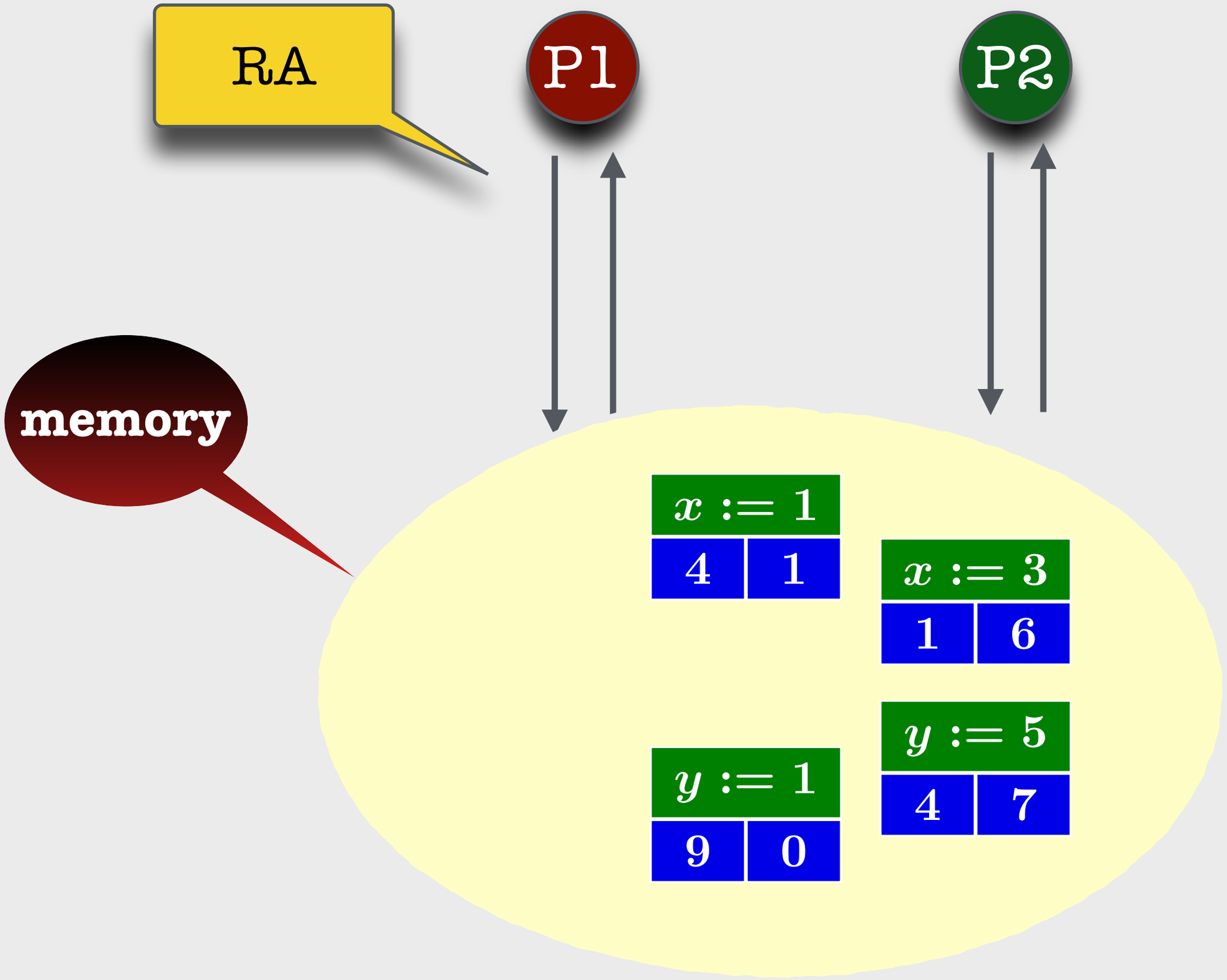
Process 1	Process 2
1. x=1;	1. while(y=0)
2. y=2;	skip;
	2. rx=x;

rx=1?

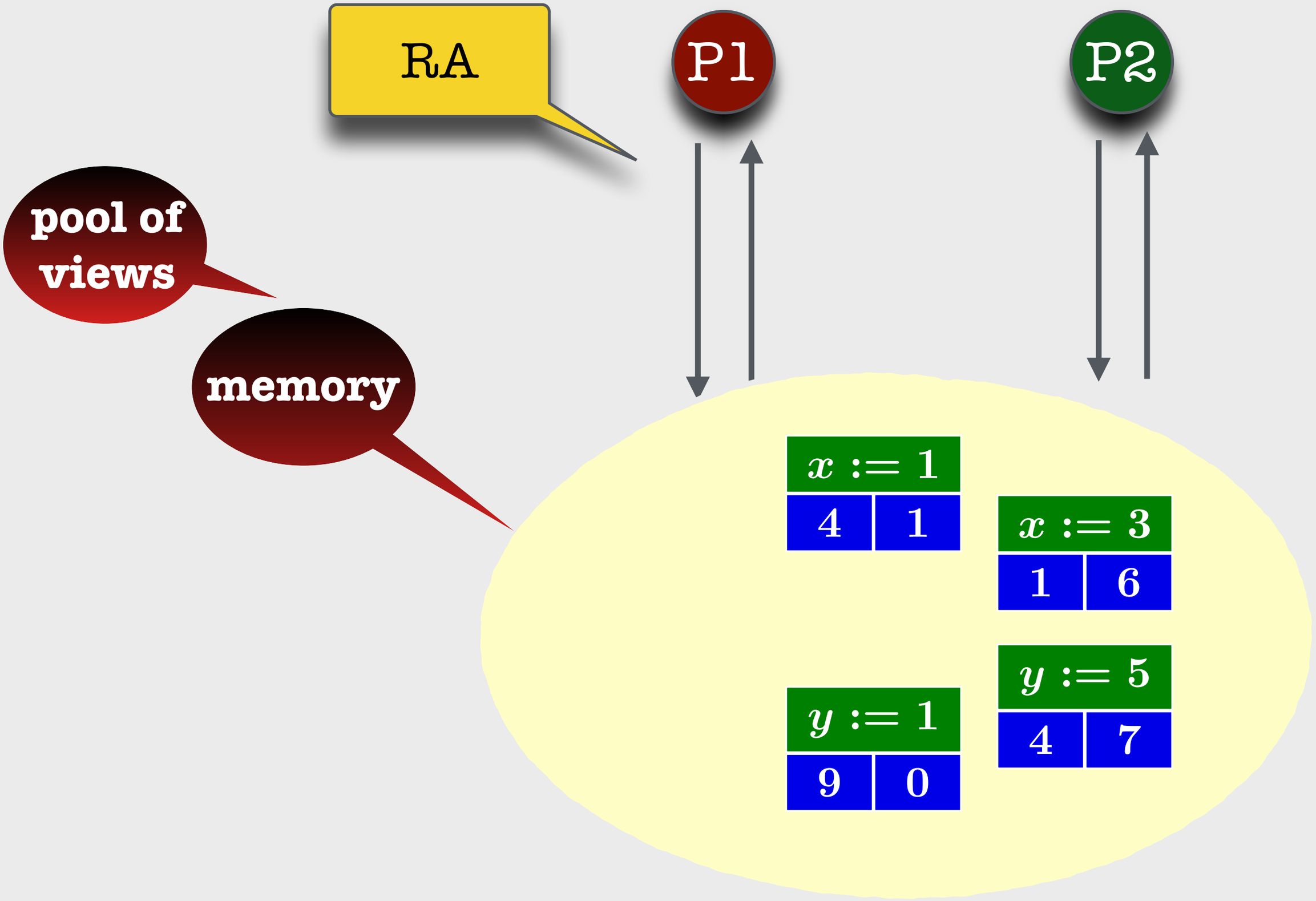
RA: High Level Description



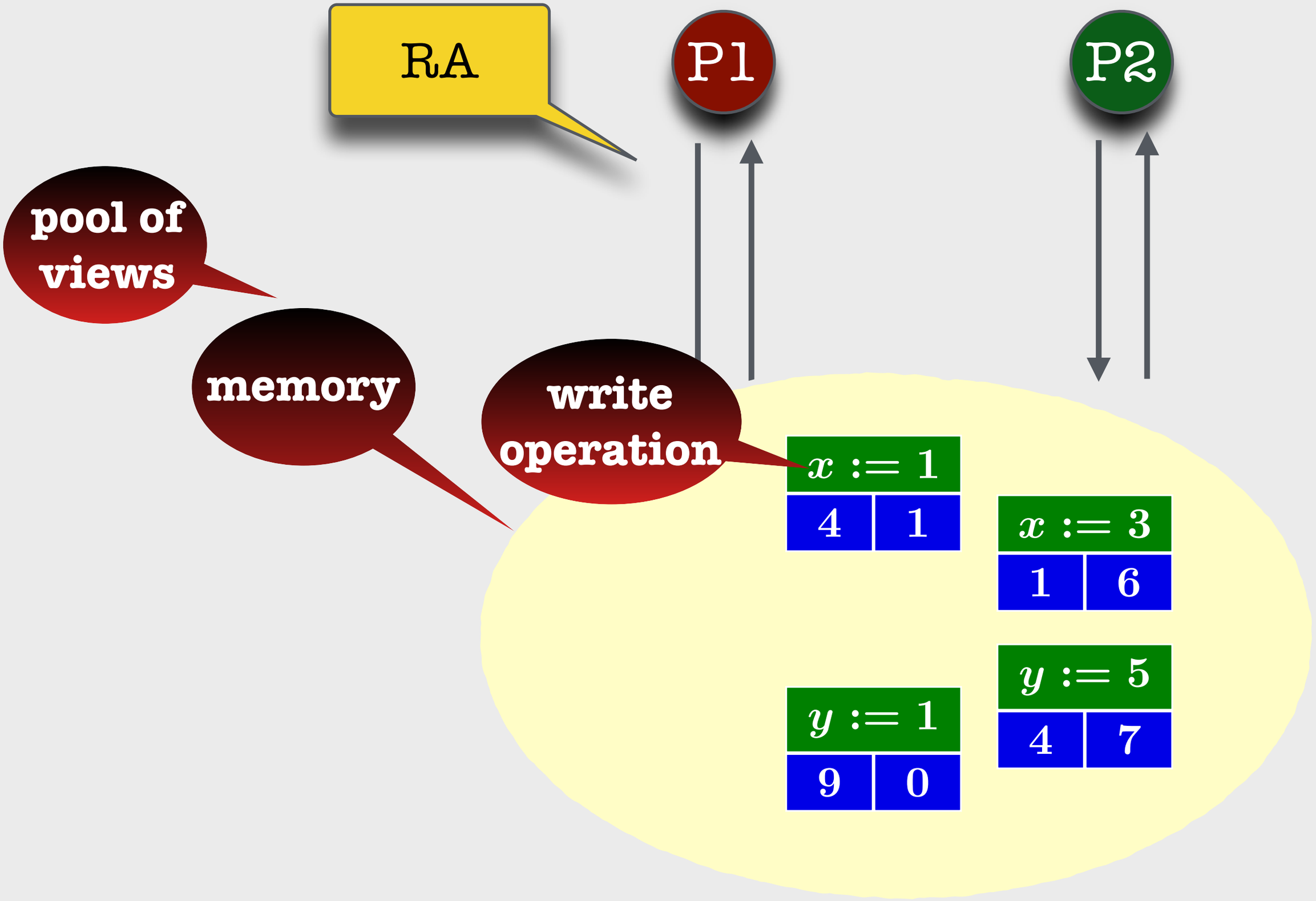
RA: High Level Description



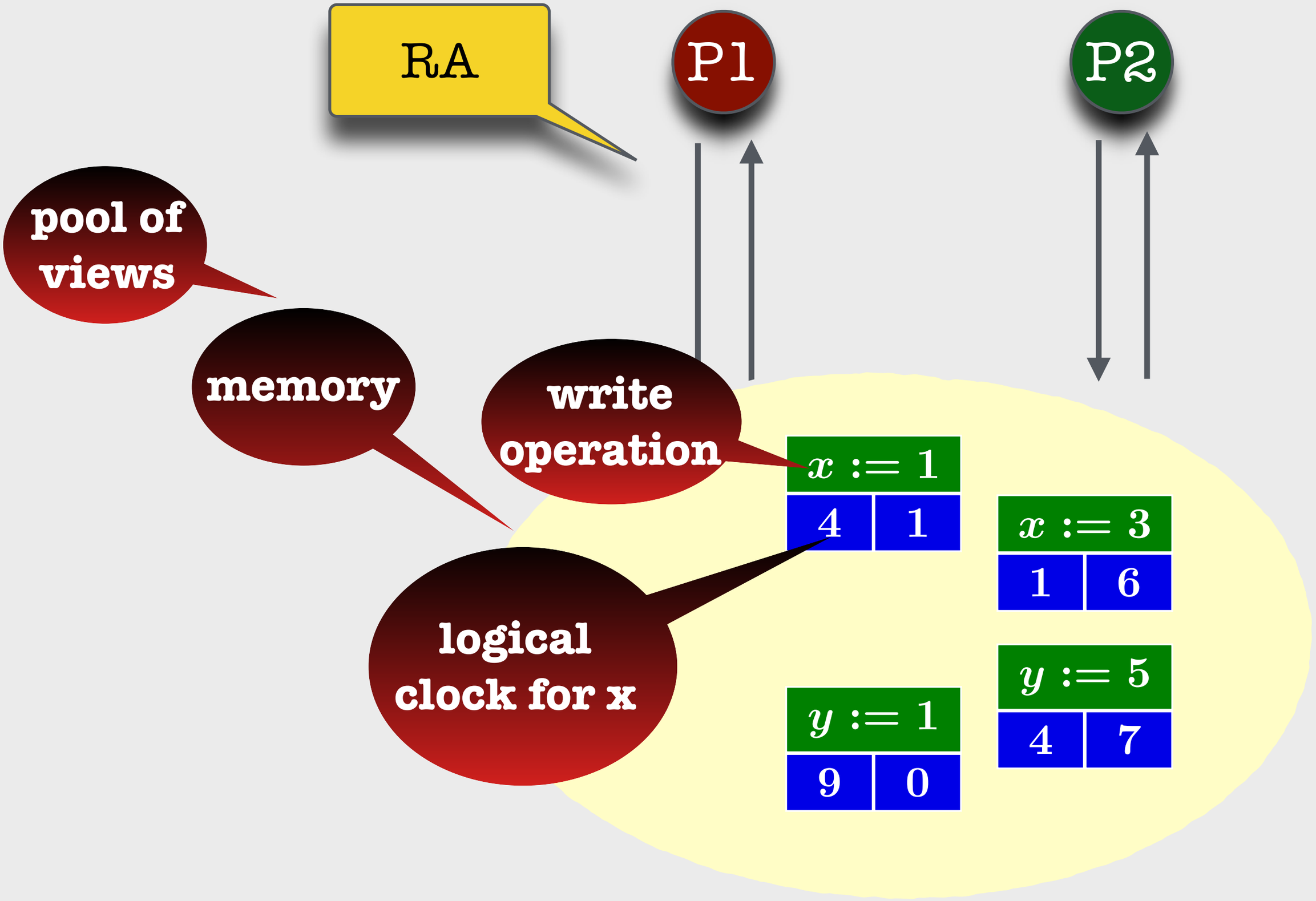
RA: High Level Description



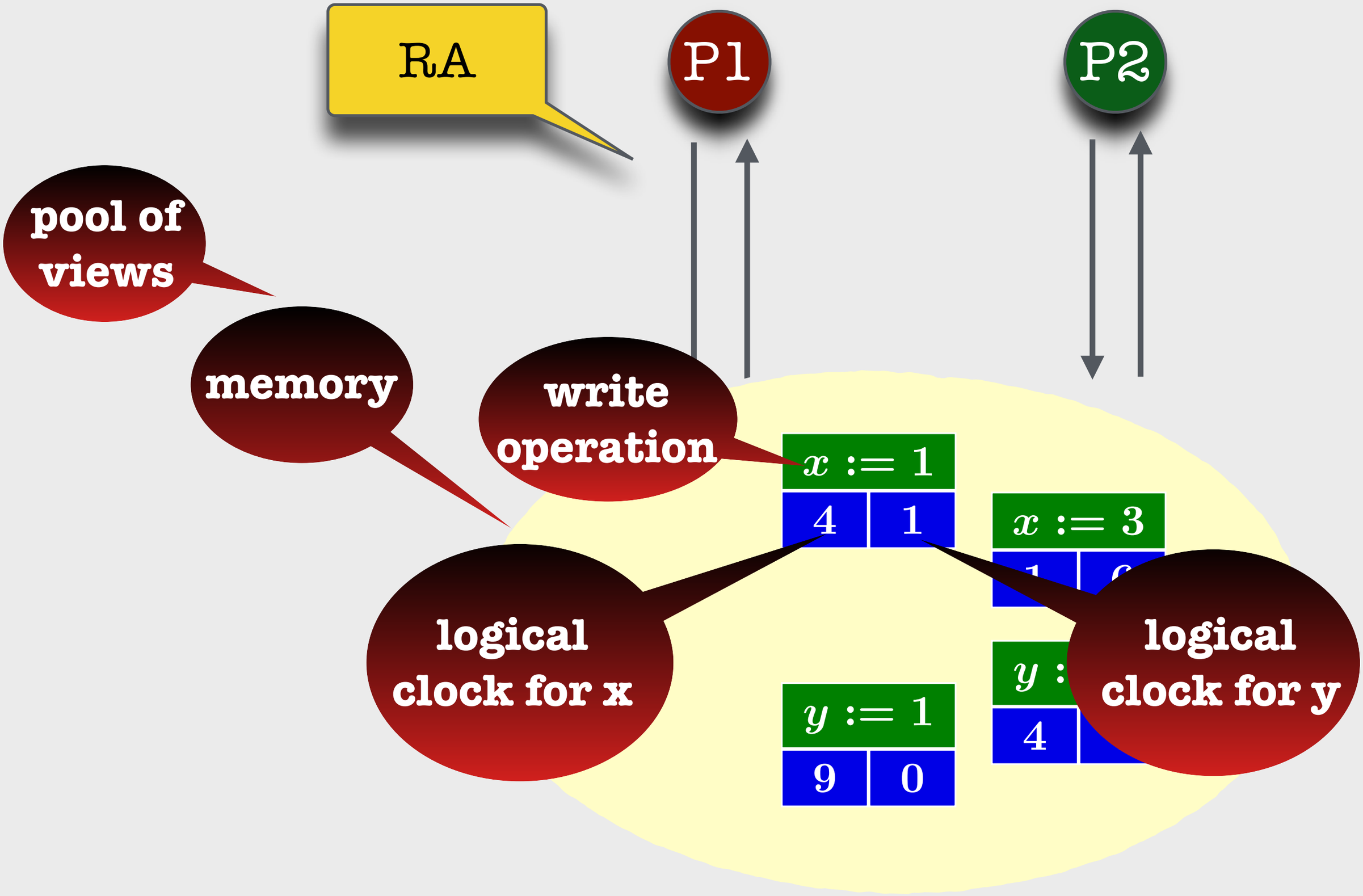
RA: High Level Description



RA: High Level Description



RA: High Level Description

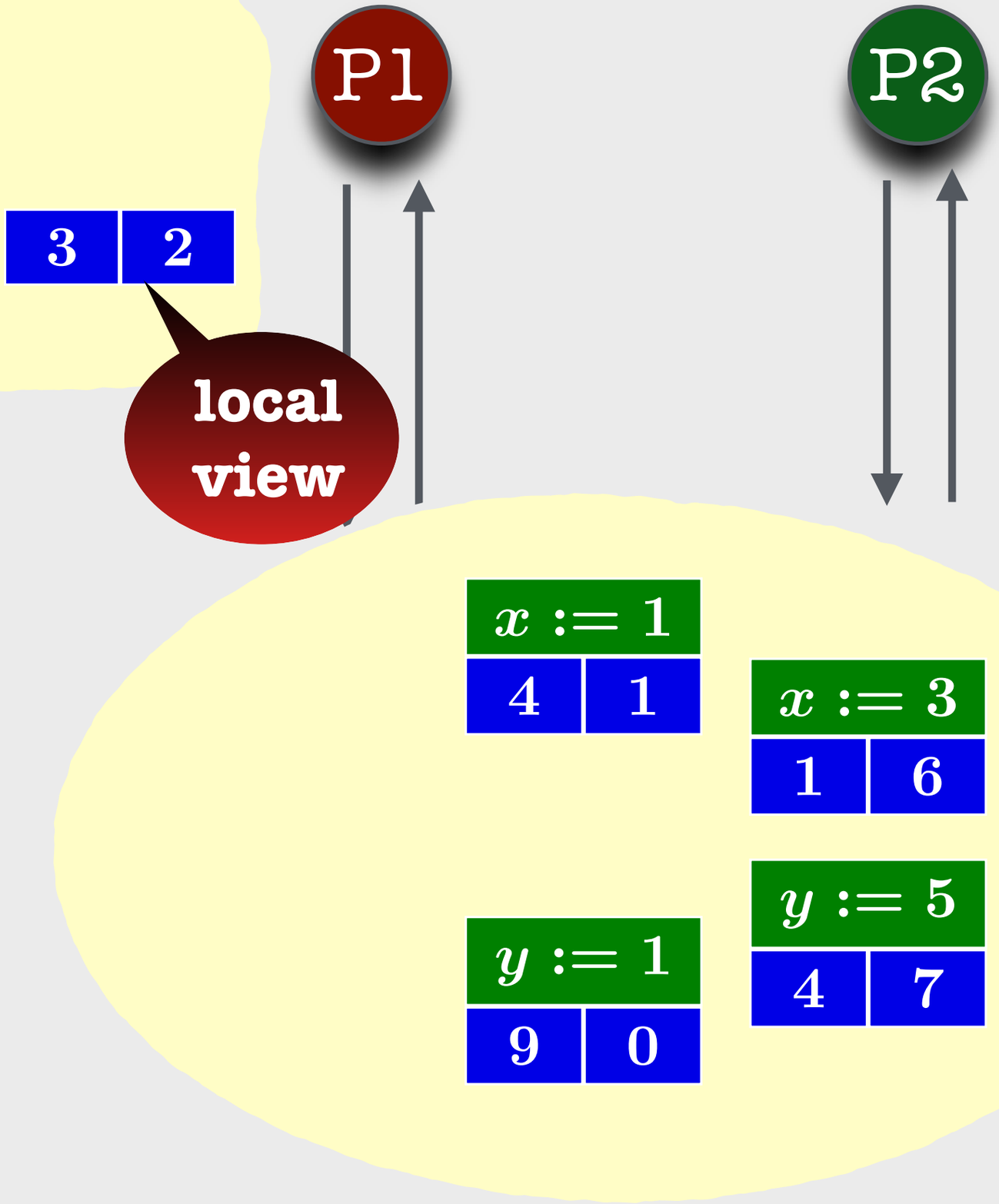


RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$

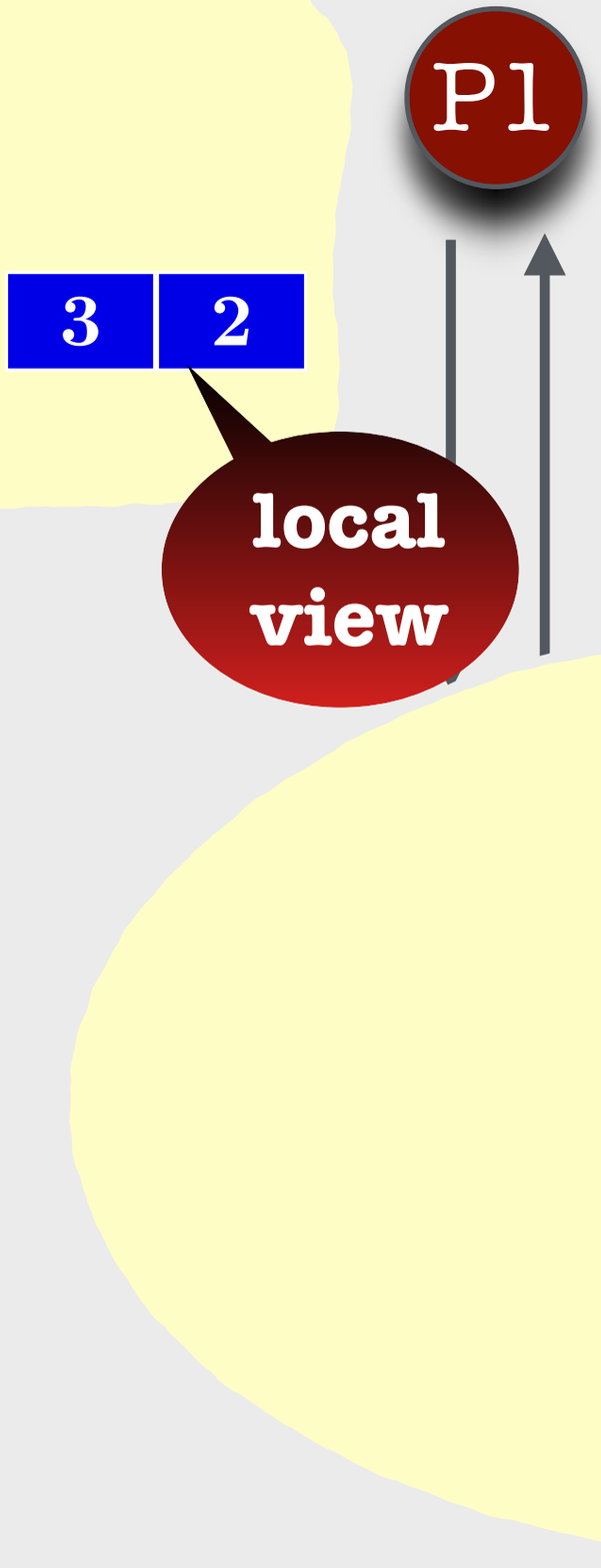


RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$



Read

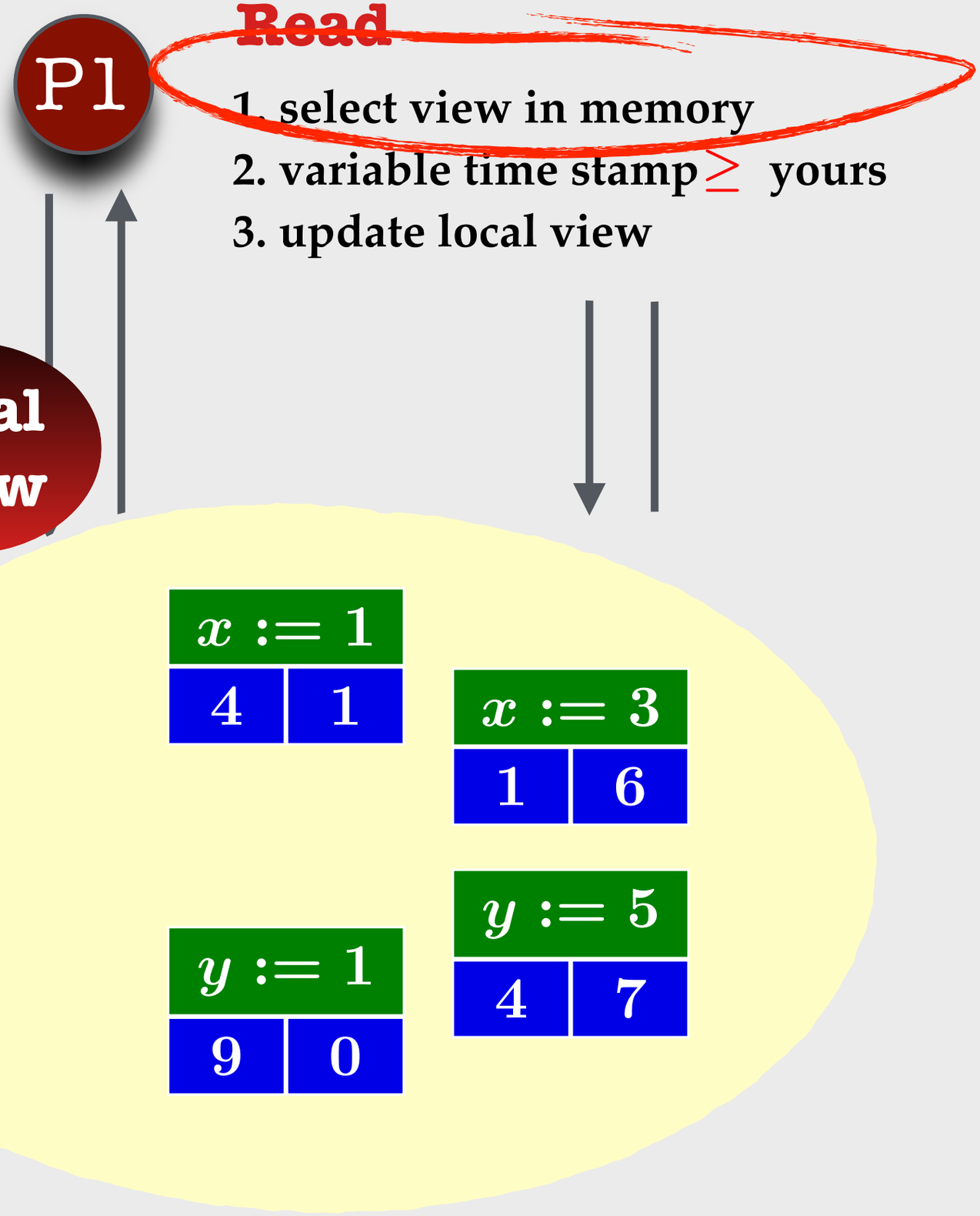
1. select view in memory
2. variable time stamp $\geq$ yours
3. update local view

RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$

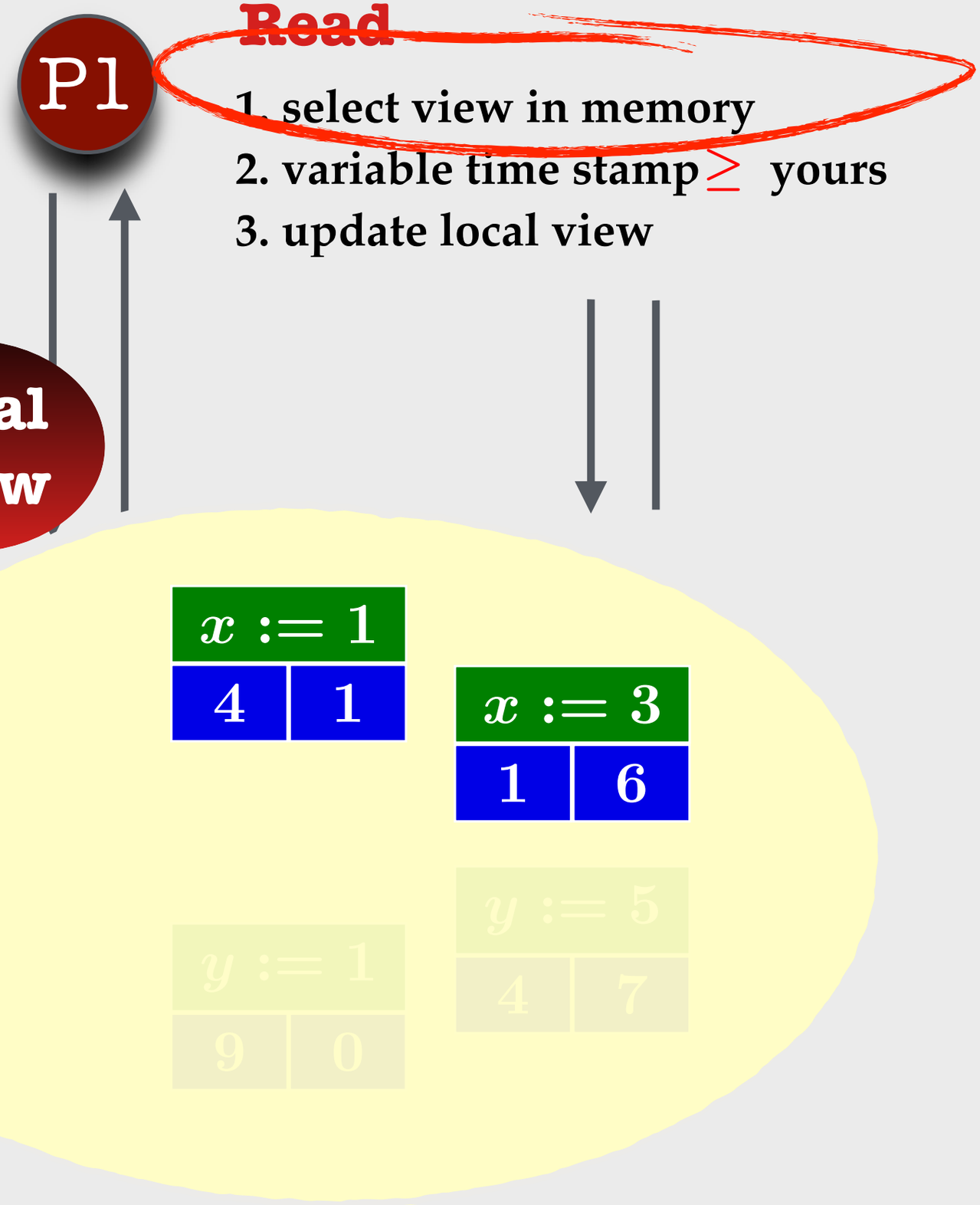


RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$



RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$

P1

Read

1. ~~select view in memory~~
2. variable time stamp $\geq$ yours
3. ~~update local view~~

3	2

local view

$x := 1$	
4	1

$x := 3$	
1	6

$y := 1$	
9	0

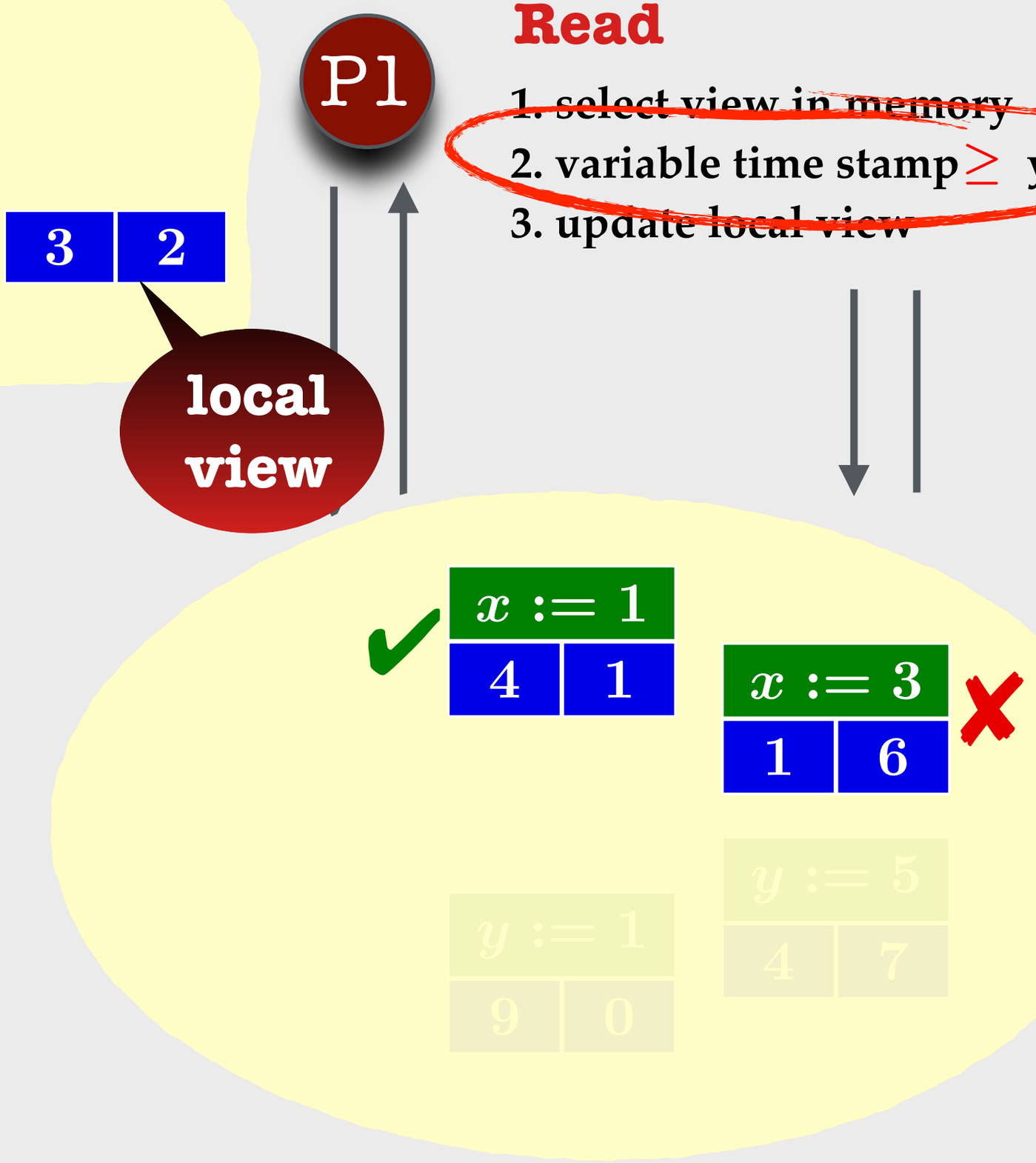
$y := 5$	
4	7

RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$

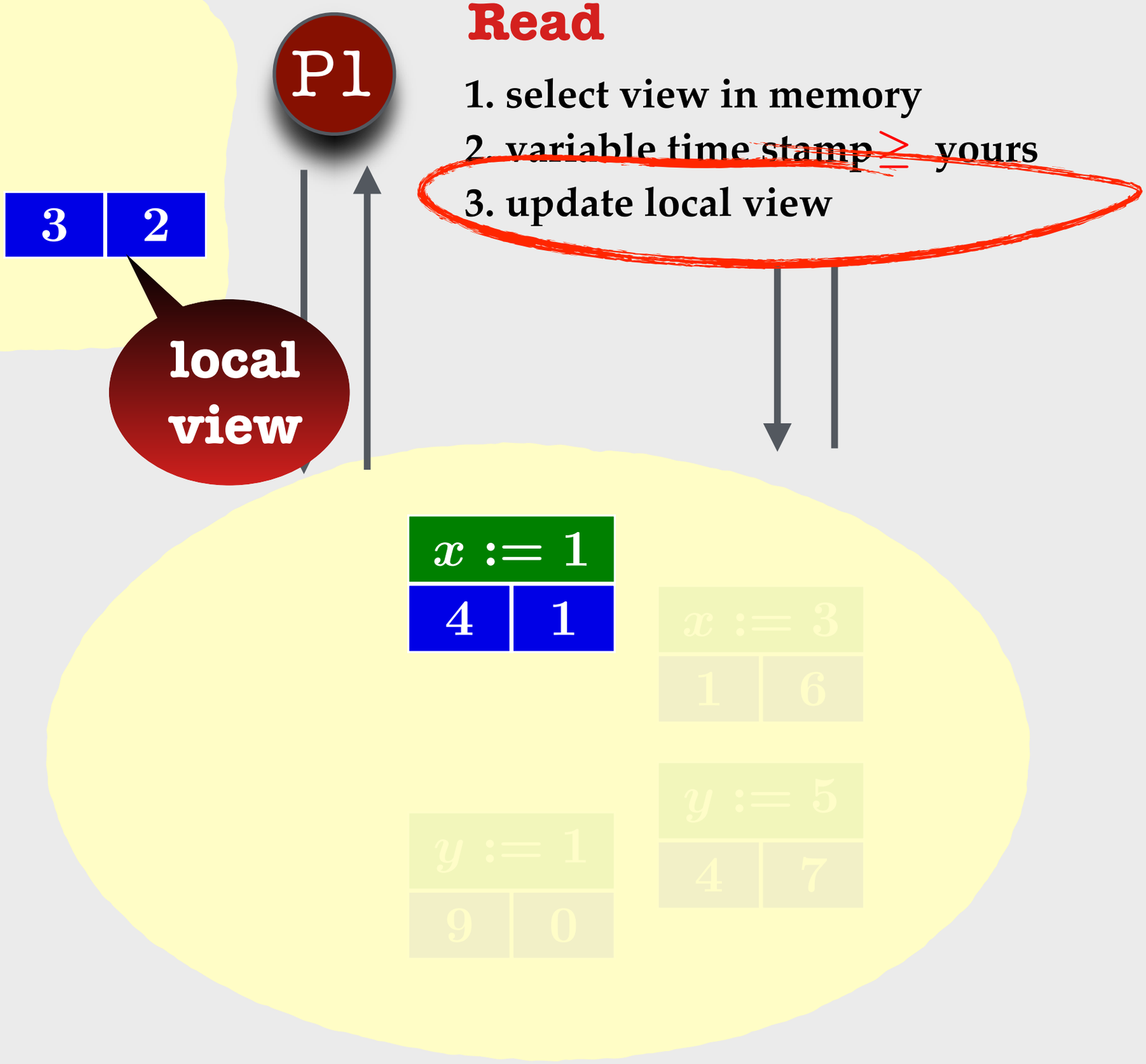


RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$

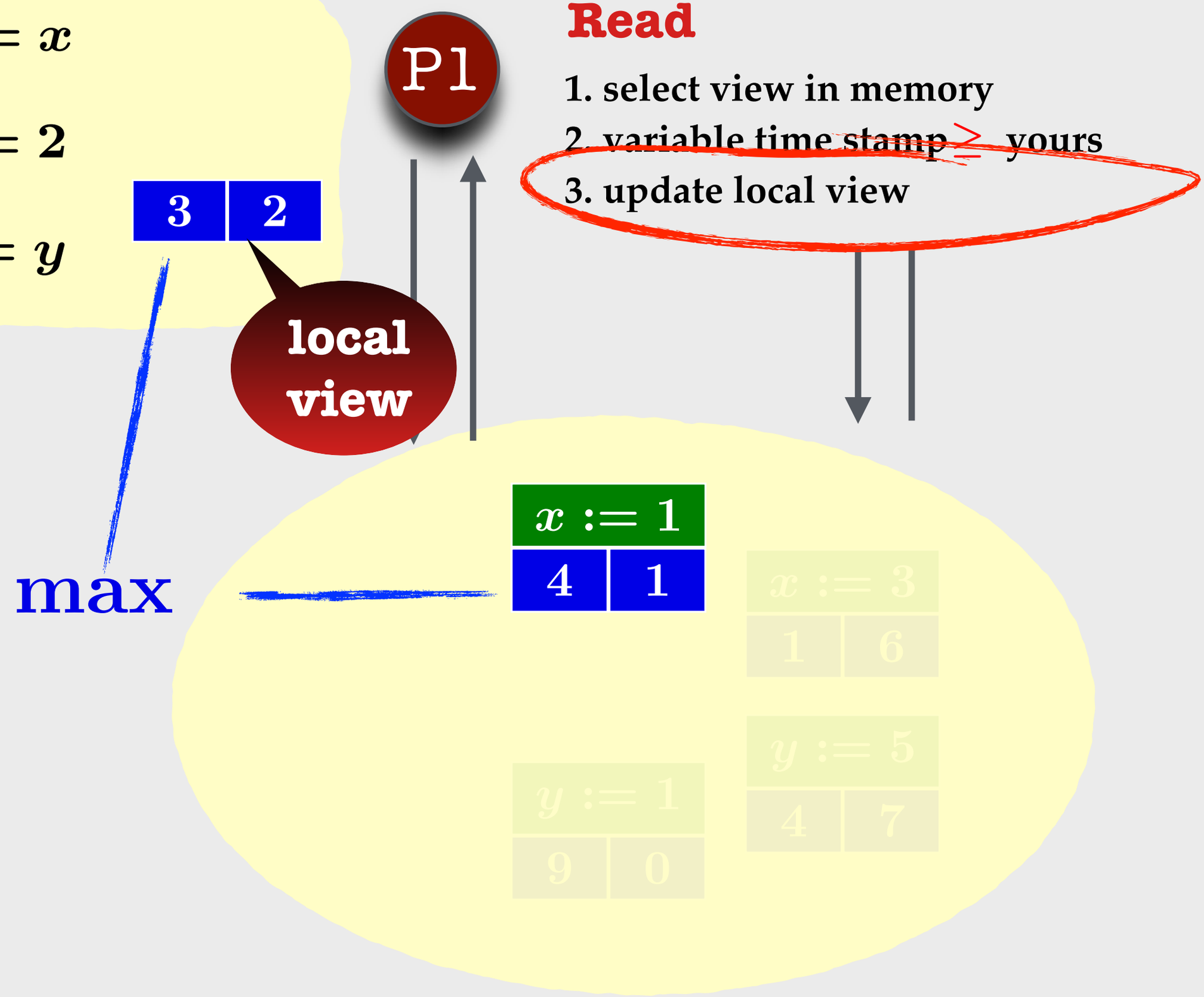


RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$

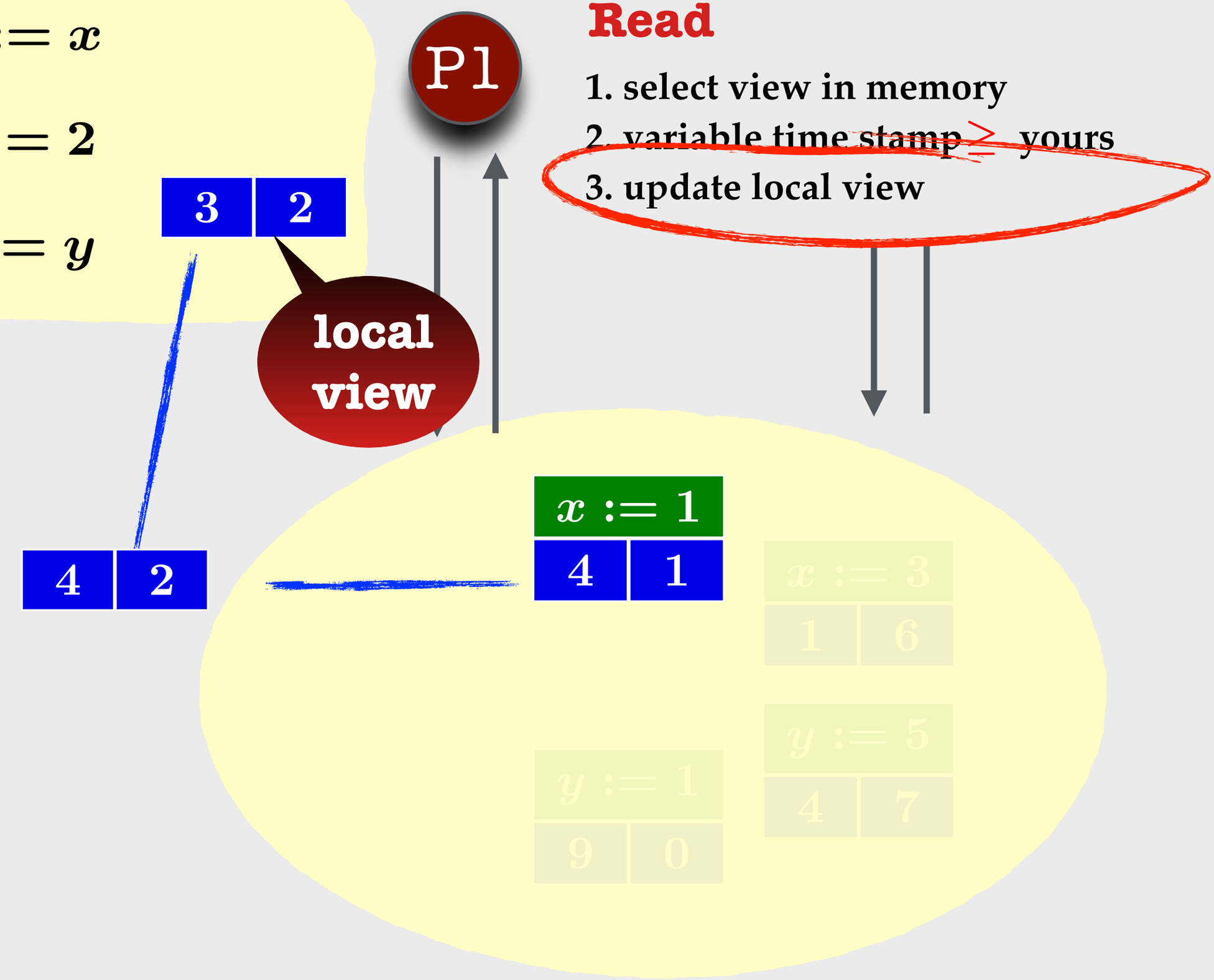


RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$

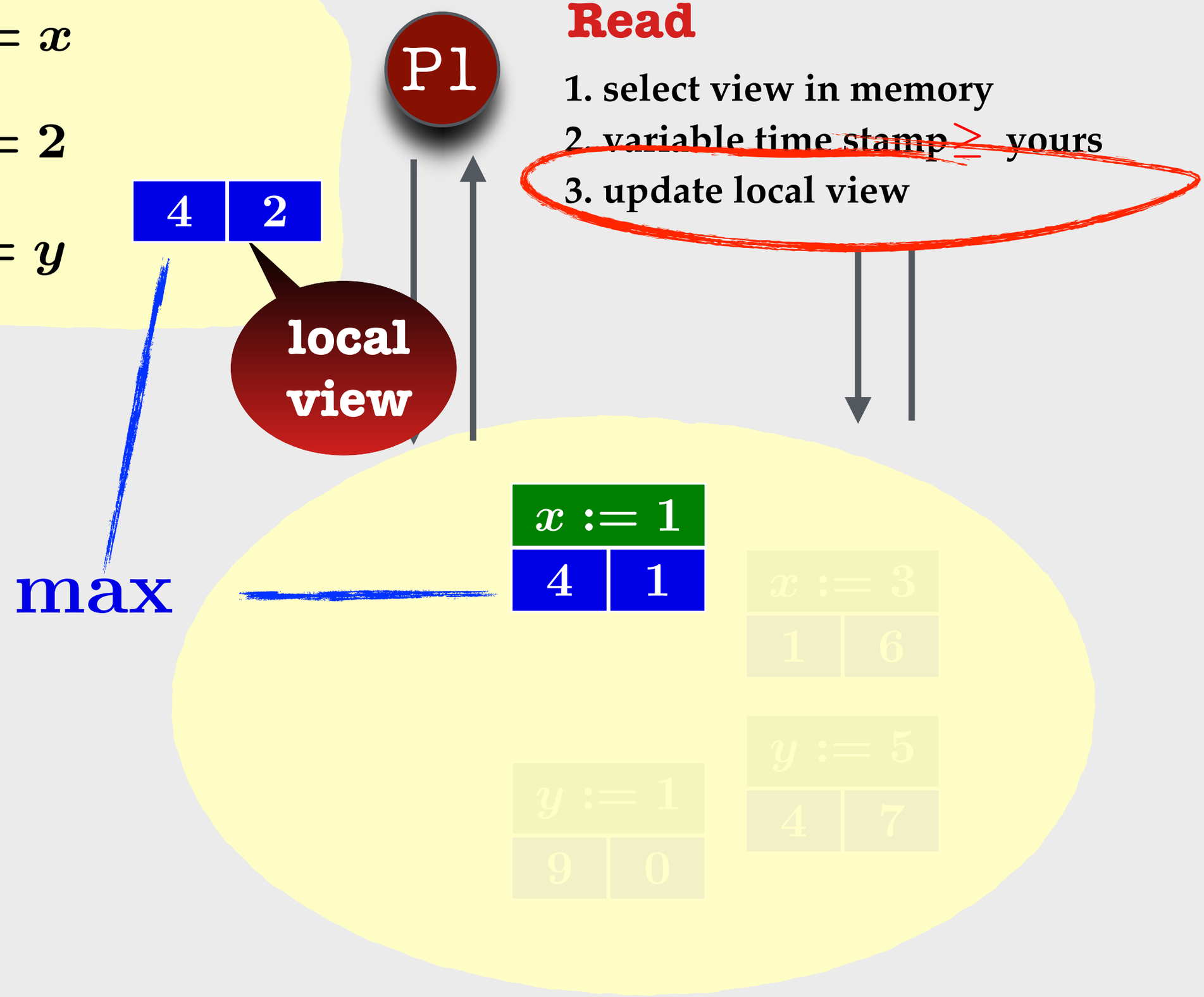


RA: High Level Description

P_1 : $a := x$

P_1 : $x := 2$

P_1 : $b := y$



RA: High Level Description

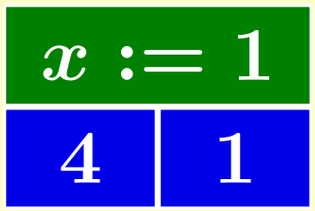
P_1 : $a := x$

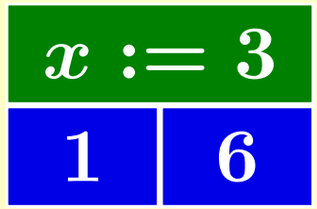
P_1 : $x := 2$

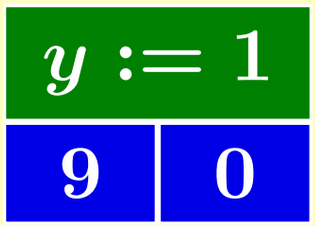
P_1 : $b := y$

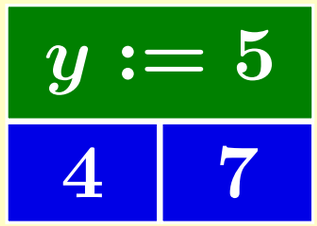
Read

1. select view in memory
2. variable time stamp $\geq$ yours
3. update local view









RA: High Level Description

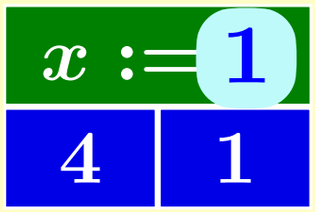
P_1 : $a := x$

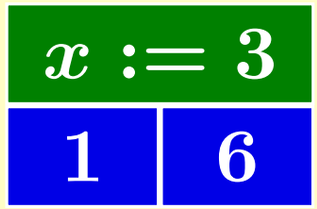
P_1 : $x := 2$

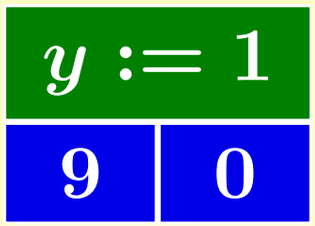
P_1 : $b := y$

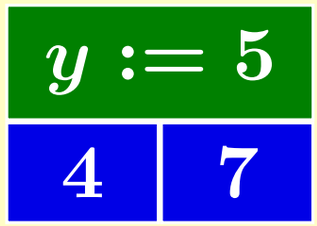
Read

1. select view in memory
2. variable time stamp $\geq$ yours
3. update local view









RA: High Level Description

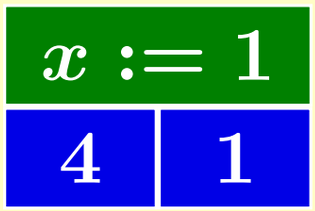
P_1 : $a := 1$

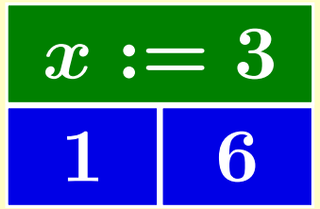
P_1 : $x := 2$

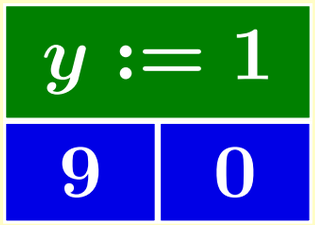
P_1 : $b := y$

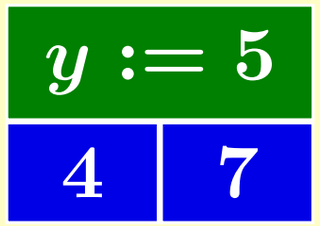
Read

1. select view in memory
2. variable time stamp $\geq$ yours
3. update local view









RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

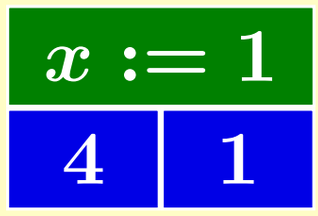
P_1 : $b := y$

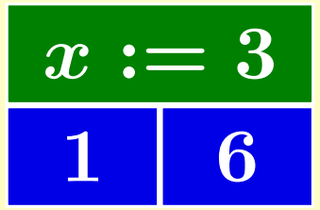
P1

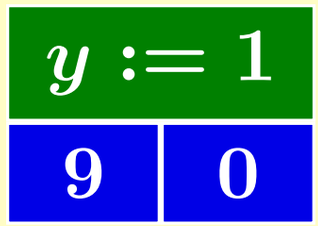
local view

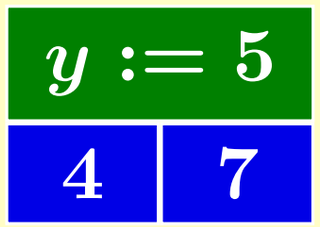
Read

1. select view in memory
2. variable time stamp $\geq$ yours
3. update local view







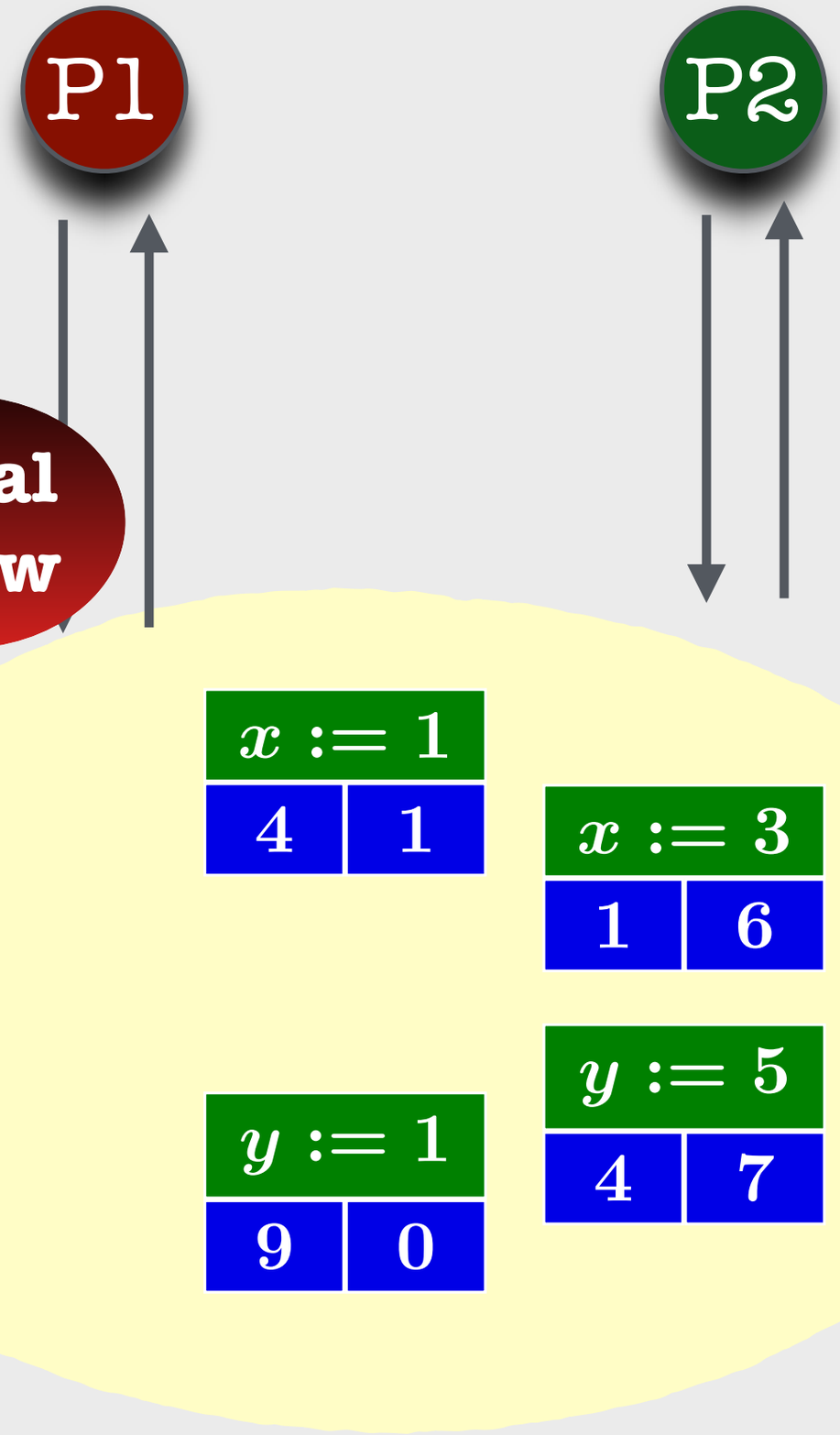


RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

P_1 : $b := y$



RA: High Level Description

P_1 : $a := 1$

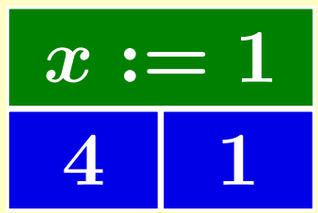
P_1 : $x := 2$

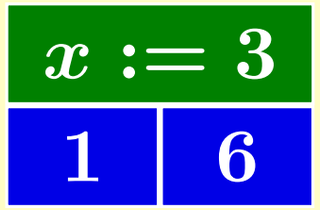
P_1 : $b := y$

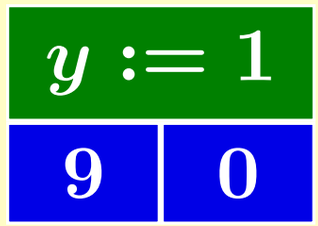
local view

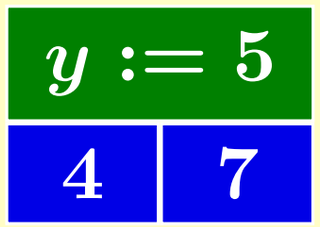
Write

1. create new local view
2. variable time stamp:
   i. newer than yours
   ii. not in memory
3. copy new view to memory







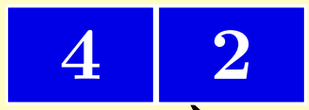


RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

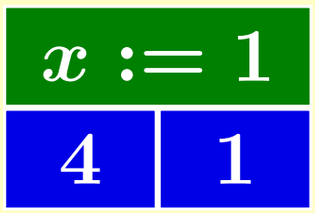
P_1 : $b := y$

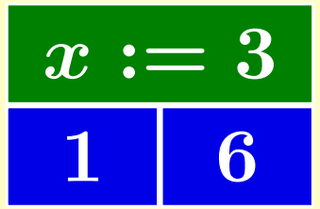


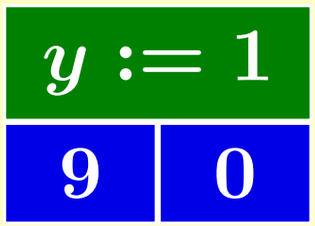
local view

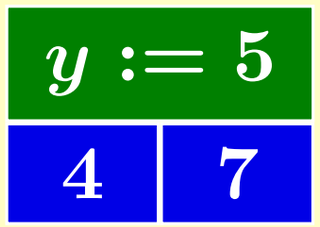
Write

1. create new local view
2. variable time stamp:
   i. newer than yours
   ii. not in memory
3. copy new view to memory







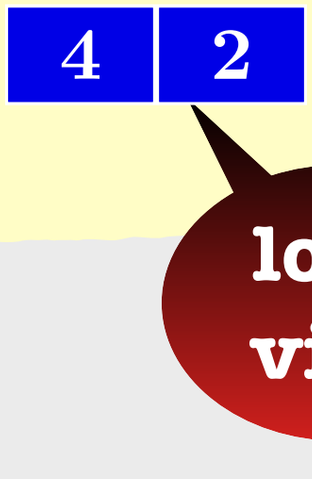


RA: High Level Description

P_1 : $a := 1$

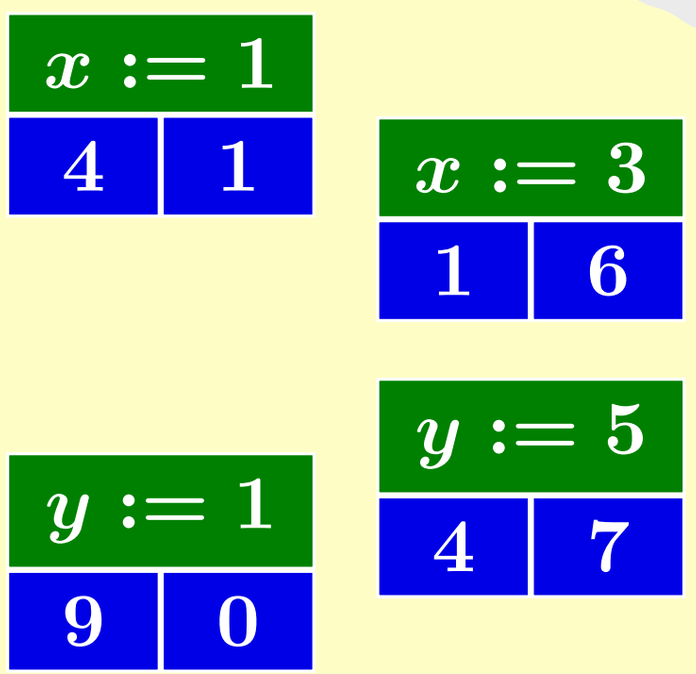
P_1 : $x := 2$

P_1 : $b := y$



Write

1. create new local view
2. variable time stamp:
   i. newer than yours
   ii. not in memory
3. copy new view to memory



RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

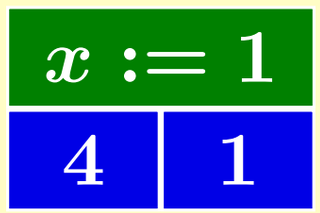
P_1 : $b := y$

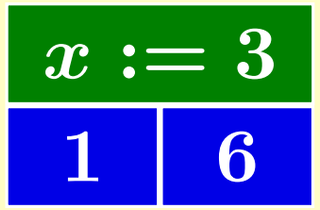
local view

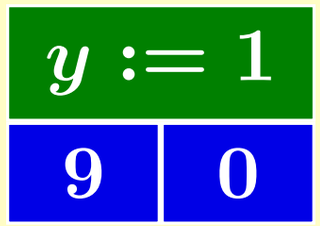
P1

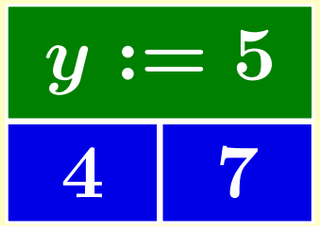
Write

1. create new local view
2. variable time stamp:
   i. newer than yours
   ii. not in memory
3. copy new view to memory









RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

P_1 : $b := y$

$x := 2$	
5	2
4	2

P1

local view

Write

1. create new local view
2. variable time stamp:
   i. newer than yours
   ii. not in memory
3. copy new view to memory

$x := 1$	
4	1

$x := 3$	
1	6

$y := 1$	
9	0

$y := 5$	
4	7

RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

P_1 : $b := y$

$x := 2$	
5	2

5	2

P1

local view

Write

1. create new local view
2. variable time stamp:
   i. newer than yours
   ii. not in memory
3. copy new view to memory

$x := 1$	
4	1

$x := 3$	
1	6

$y := 1$	
9	0

$y := 5$	
4	7

RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

P_1 : $b := y$

$x := 2$	
5	2

5	2

local view

P1

Write

1. create new local view
2. variable time stamp:
   i. newer than yours
   ii. not in memory
3. copy new view to memory

$x := 1$	
4	1

$x := 3$	
1	6

$y := 1$	
9	0

$y := 5$	
4	7

RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

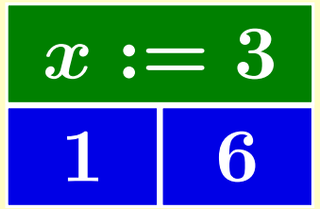
P_1 : $b := y$

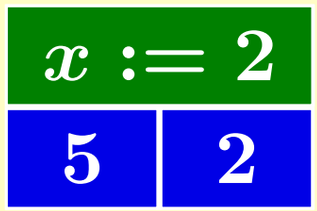
local view

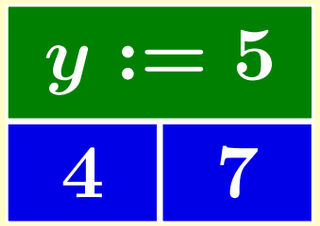
P1

Write

1. create new local view
2. variable time stamp:
   i. newer than yours
   ii. not in memory
3. copy new view to memory





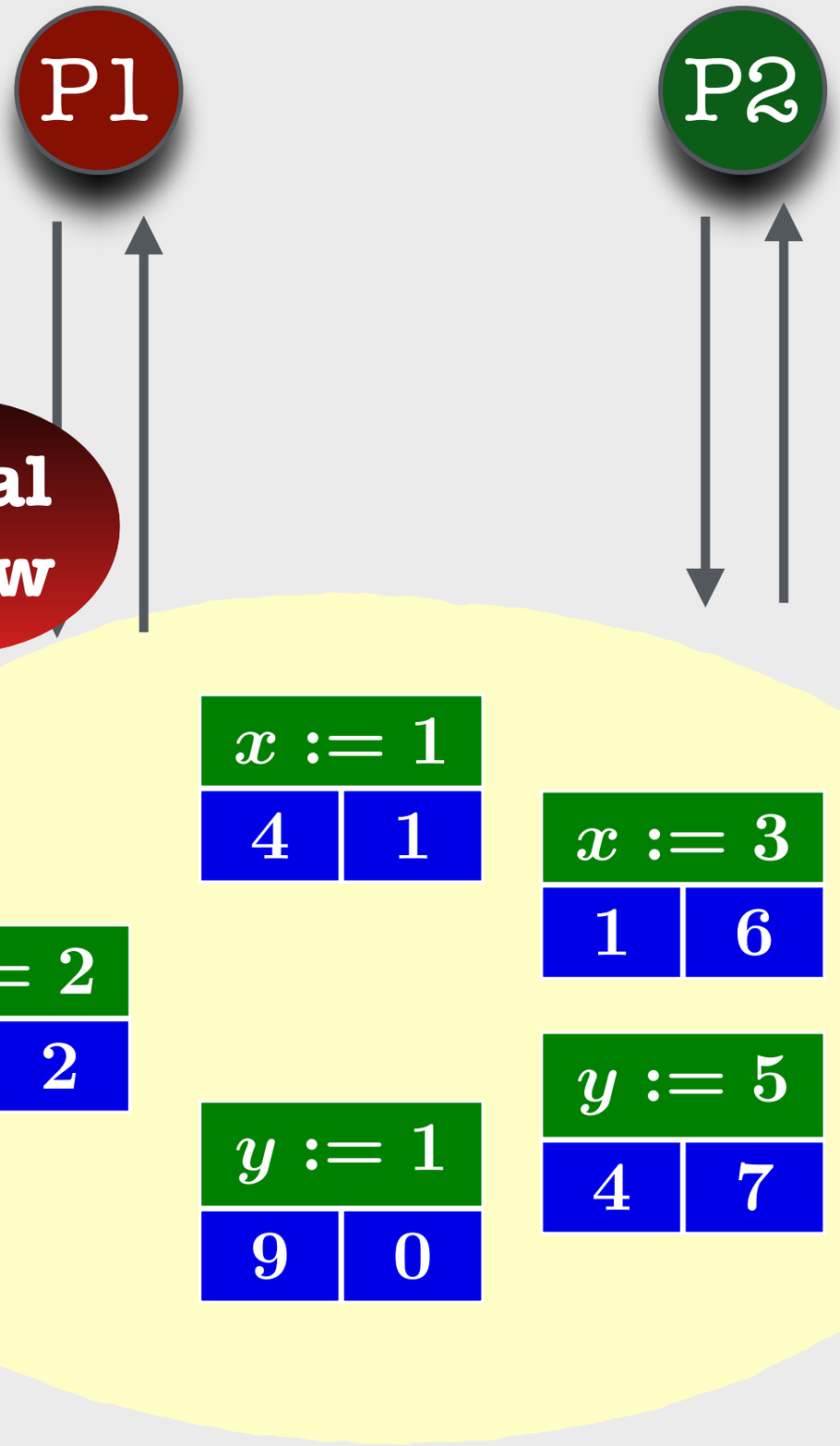


RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

P_1 : $b := y$



RA: High Level Description

P_1 : $a := 1$

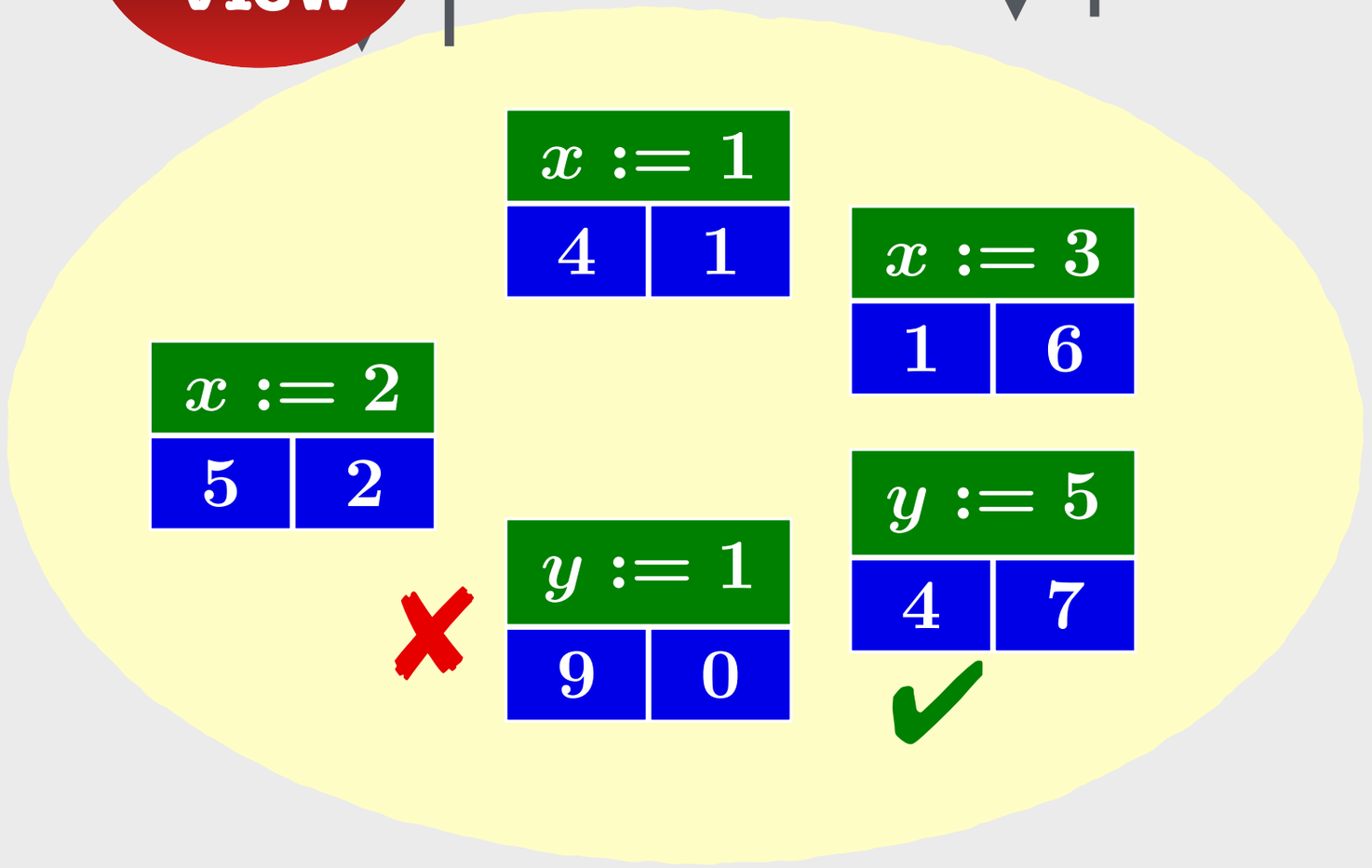
P_1 : $x := 2$

P_1 : $b := y$

Read

1. select view in memory
2. variable time stamp $\geq$ yours
3. update local view

local view



RA: High Level Description

P_1 : $a := 1$

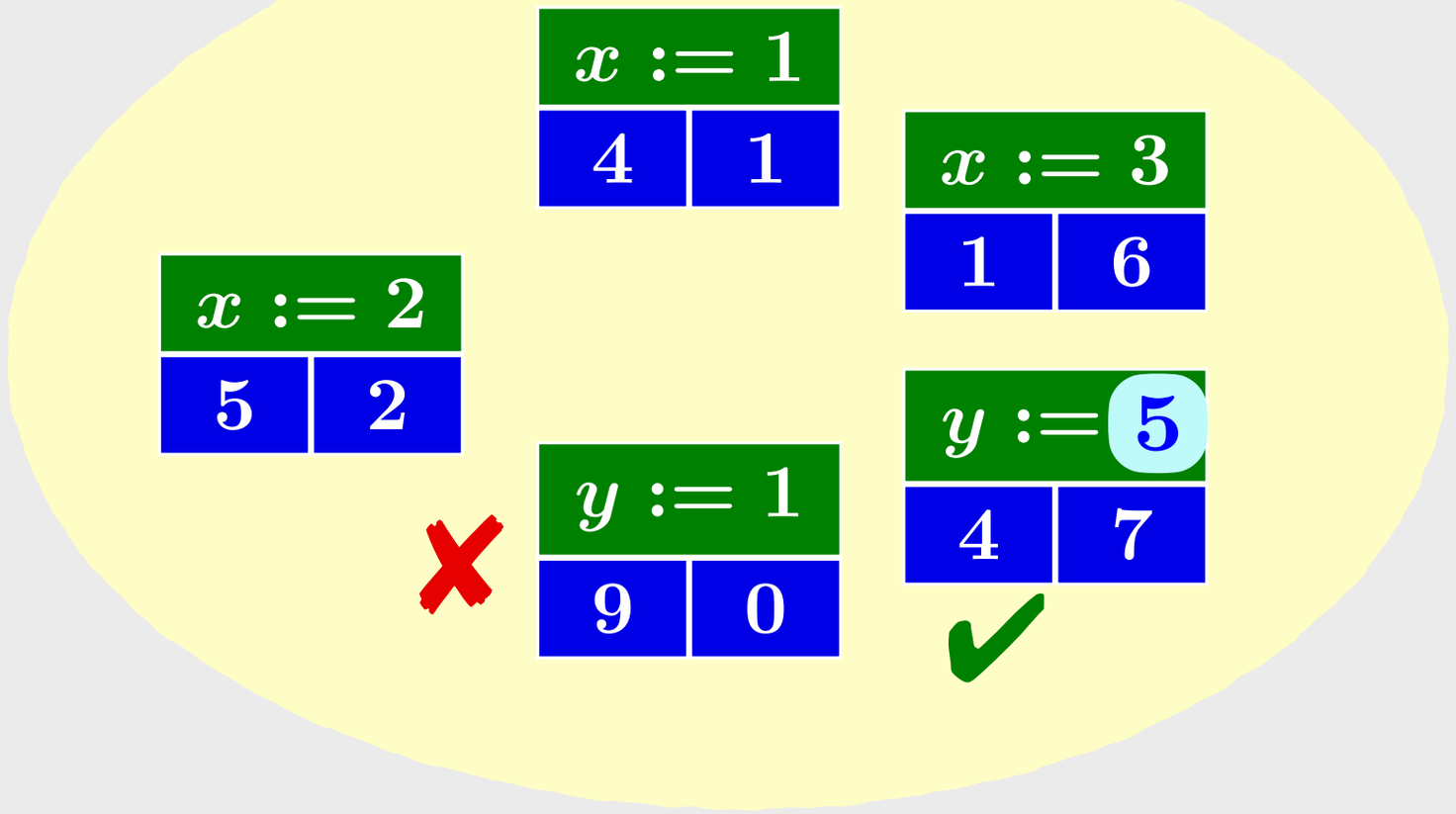
P_1 : $x := 2$

P_1 : $b := y$

Read

1. select view in memory
2. variable time stamp $\geq$ yours
3. update local view

local view

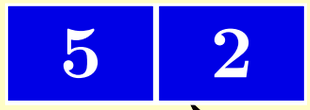


RA: High Level Description

P_1 : $a := 1$

P_1 : $x := 2$

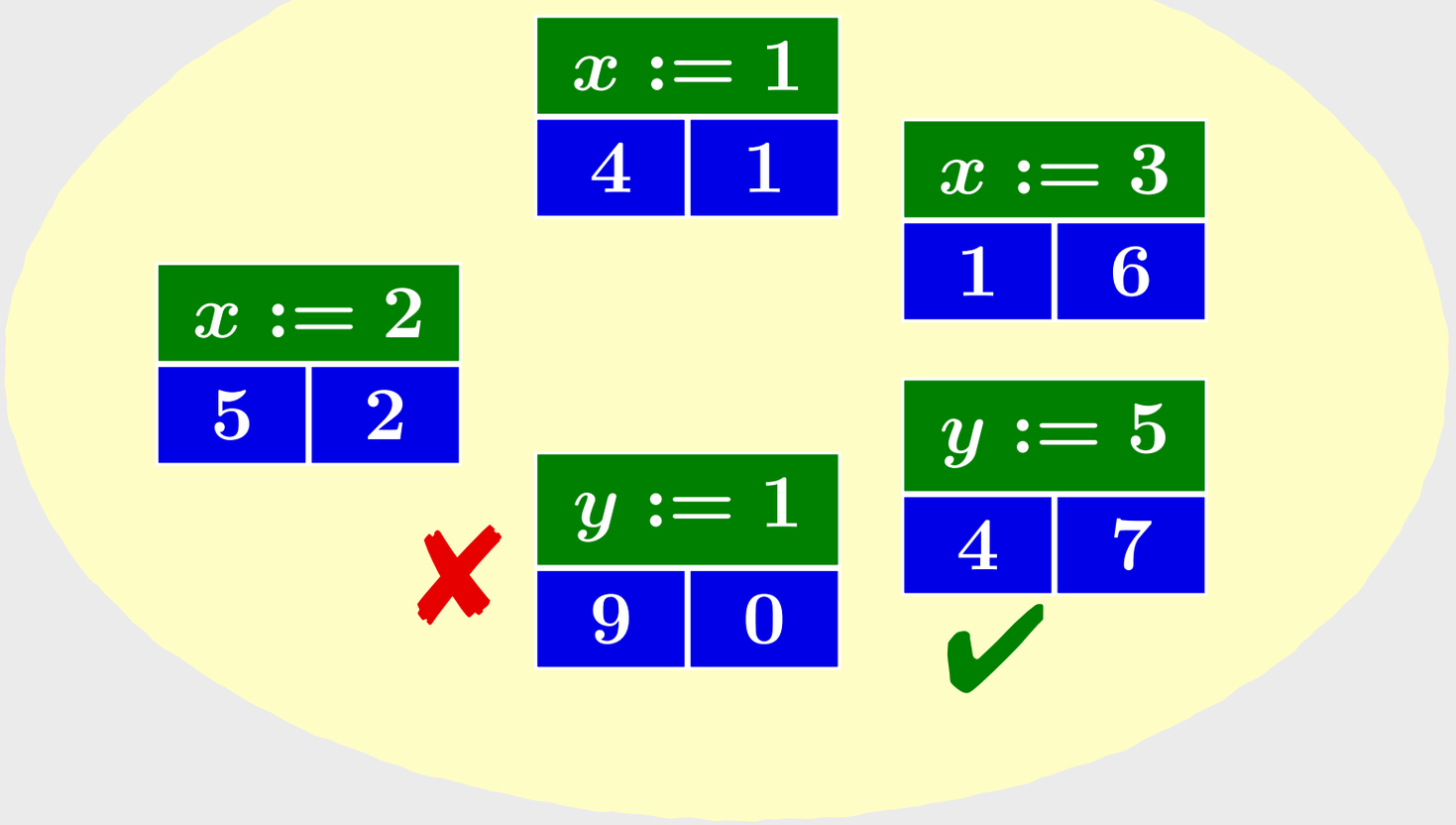
P_1 : $b := 5$



local view

Read

1. select view in memory
2. variable time stamp $\geq$ yours
3. update local view



Init: x=y=0

Store buffer

1. x=1;
2. ry=y;

1. y=1;
2. rx=x;

rx=0 and ry=0?

$y := 0$
0

$x := 0$
0

0

Init: x=y=0

Store buffer

1. x=1;
2. ry=y;

0	0

1. y=1;
2. rx=x;

rx=0 and ry=0?

$y := 0$	
0	0

$x := 0$	
0	0

Init: x=y=0

Store buffer

1. x=1;
2. ry=y;

0	0

1. y=1;
2. rx=x;

0	0

rx=0 and ry=0?

$y := 0$	
0	0

$x := 0$	
0	0

Init: x=y=0

Store buffer

1. x=1;
2. ry=y;

1. y=1;
2. rx=x;

rx=0 and ry=0?

$y := 0$
0

$x := 0$
0

0

Init: x=y=0

Store buffer

1. x=1;
2. ry=y;

3	0

1. y=1;
2. rx=x;

rx=0 and ry=0?

$y := 0$
0 | 0

$x := 0$
0 | 0

$x := 1$
3 | 0

Init: x=y=0

Store buffer

1. x=1;
2. ry=y;

3	0

1. y=1;
2. rx=x;

0	7

rx=0 and ry=0?

$y := 1$

0	7

$y := 0$

0	0

$x := 0$

0	0

$x := 1$

3	0

RA: High Level Description

r := 1
y := 1
r′ := 2
arw(x, r, r′)

0	0

also known as CAS/RMW

$x := 1$	
3	0

$x := 0$	
0	0

$y := 0$	
0	0

RA: High Level Description

r := 1

y := 1

r′ := 2

arw(x, r, r′)

0	0

also known as CAS/RMW

$x := 1$	
3	0

$x := 0$	
0	0

$y := 0$	
0	0

RA: High Level Description

r := 1
y := 1
r′ := 2
arw(x, r, r′)

0	7

also known as CAS/RMW

$x := 1$	
3	0

$x := 0$	
0	0

$y := 1$	
0	7

$y := 0$	
0	0

RA: High Level Description

$\mathbf{r := 1}$
$\mathbf{y := 1}$
$\mathbf{r' := 2}$
$\mathbf{arw(x, r, r')}$

0	7

also known as CAS/RMW

Atomic read write (or CAS)

1. select view in memory with value of r
2. variable time stamp t $\geq$ yours
3. update local view, and write with value of r′
4. choose timestamp t+1 in the new message

$x := 1$	
3	0

$x := 0$	
0	0

$y := 1$	
0	7

$y := 0$	
0	0

RA: High Level Description

$\mathbf{r := 1}$

$\mathbf{y := 1}$

$\mathbf{r' := 2}$

$\mathbf{arw(x, r, r')}$

3	7

also known as CAS/RMW

Atomic read write (or CAS)

1. select view in memory with value of r
2. variable time stamp $t \geq$ yours
3. update local view, and write with value of r′
4. choose timestamp t+1 in the new message

$x := 1$	
3	0

$x := 0$	
0	0

$y := 1$	
0	7

$y := 0$	
0	0

RA: High Level Description

$\mathbf{r := 1}$

$\mathbf{y := 1}$

$\mathbf{r' := 2}$

$\mathbf{arw(x, r, r')}$

4	7

also known as CAS/RMW

Atomic read write (or CAS)

1. select view in memory with value of r
2. variable time stamp $t \geq$ yours
3. update local view, and write with value of r′
4. choose timestamp t+1 in the new message

$x := 2$	
4	7

$x := 1$	
3	0

$x := 0$	
0	0

$y := 1$	
0	7

$y := 0$	
0	0

RA run

Register values: $r1=0 $r2=0

Process 1

1. $r1 = x;
2. y = 1;
3. $r3 = x;

0	0

Process 2

1. $r2 = y;
2. x = 1;
3. x = 2;

0	0

$x := 0$	
0	0

$y := 0$	
0	0

Reachable: $r1 =0, $r2 = 1 and $r3=2?

Register values: $r1=0 $r2=0

Process 1

1. $r1 = 0;
2. y = 1;
3. $r3 = x;

0	0

Process 2

1. $r2 = y;
2. x = 1;
3. x = 2;

0	0

$x := 0$	
0	0

$y := 0$	
0	0

Reachable: $r1 =0, $r2 = 1 and $r3=2?

RA run

r(x,0)

Register values: $r1=0 $r2=0

Process 1

1. $r1 = 0;
2. y = 1;
3. $r3 = x;

0	2

Process 2

1. $r2 = y;
2. x = 1;
3. x = 2;

0	0

$y := 1$	
0	2

$x := 0$	
0	0

$y := 0$	
0	0

Reachable: $r1 =0, $r2 = 1 and $r3=2?

RA run

r(x,0)

w(y,1)

Register values: $r1=0 $r2=0

Process 1

1. $r1 = 0;
2. y = 1;
3. $r3 = x;

0	2

Process 2

1. $r2 = 1;
2. x = 1;
3. x = 2;

0	2

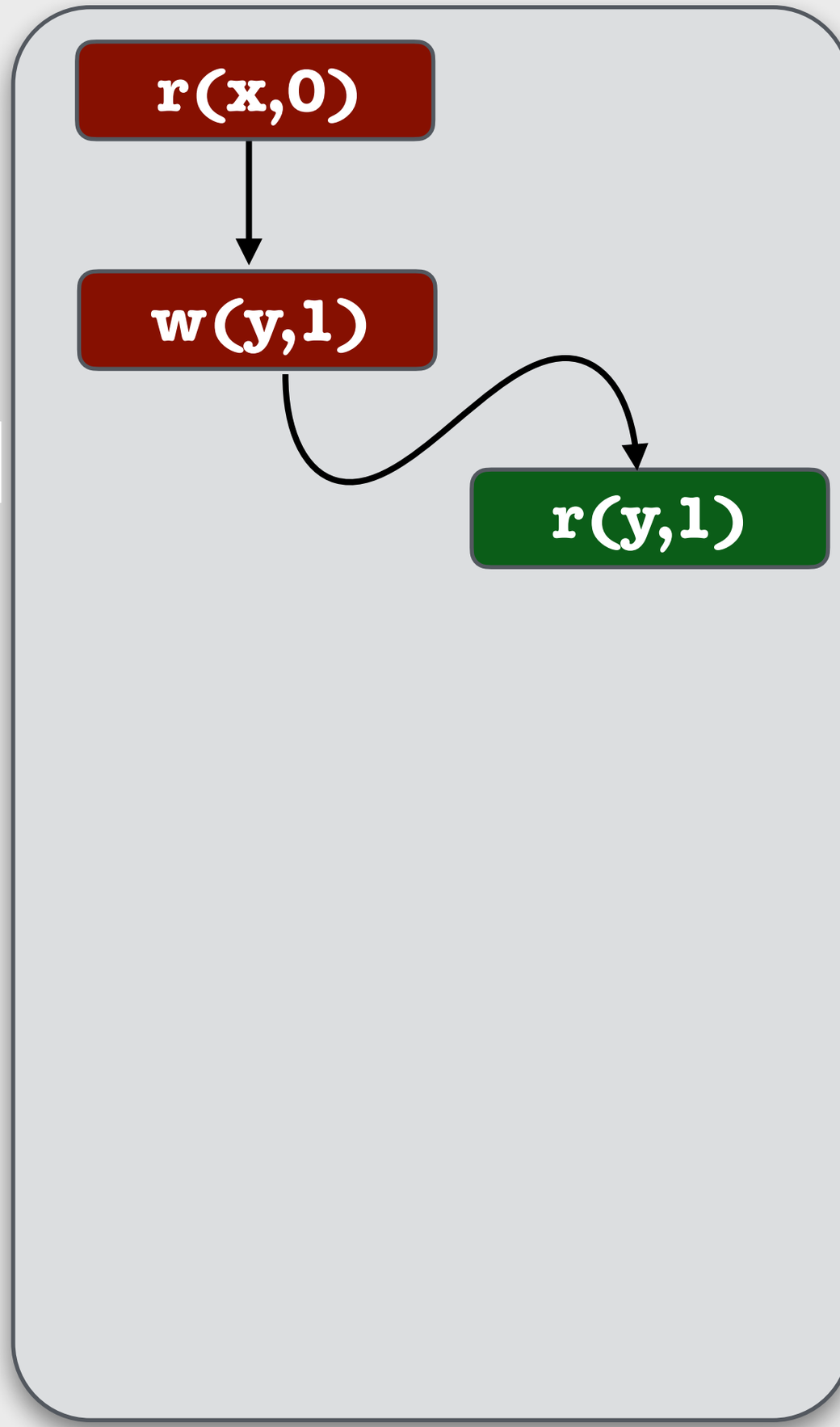
$y := 1$
0 | 2

$x := 0$
0 | 0

$y := 0$
0 | 0

Reachable: $r1 =0, $r2 = 1 and $r3=2?

RA run



Register values: $r1=0 $r2=0

Process 1

1. $r1 = 0;
2. y = 1;
3. $r3 = x;

0	2

Process 2

1. $r2 = 1;
2. x = 1;
3. x = 2;

1	2

$x := 1$	
1	2

$y := 1$	
0	2

$x := 0$	
0	0

$y := 0$	
0	0

Reachable: $r1 =0, $r2 = 1 and $r3=2?

RA run

r(x,0)

w(y,1)

r(y,1)

w(x,1)

Register values: $r1=0 $r2=0

Process 1

1. $r1 = 0;
2. y = 1;
3. $r3 = x;

0	2

Process 2

1. $r2 = 1;
2. x = 1;
3. x = 2;

3	2

$x := 2$	
3	2

$x := 1$	
1	2

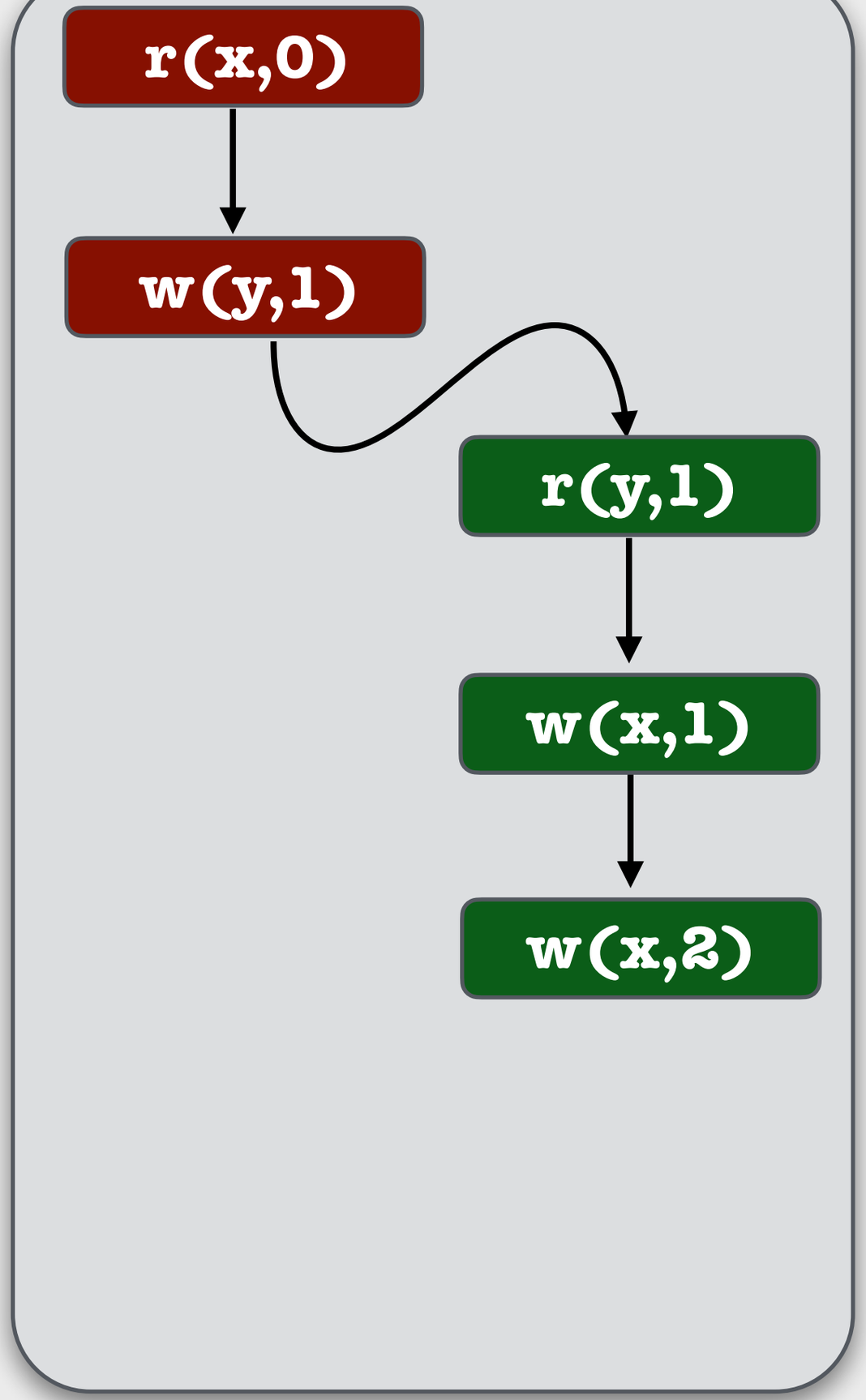
$y := 1$	
0	2

$x := 0$	
0	0

$y := 0$	
0	0

Reachable: $r1 =0, $r2 = 1 and $r3=2?

RA run



Register values: $r1=0 $r2=0

Process 1

```
1. $r1 = 0;
2. y = 1;
3. $r3 = 2;
```

3	2

Process 2

```
1. $r2 = 1;
2. x = 1;
3. x = 2;
```

3	2

$x := 2$	
3	2

$x := 1$	
1	2

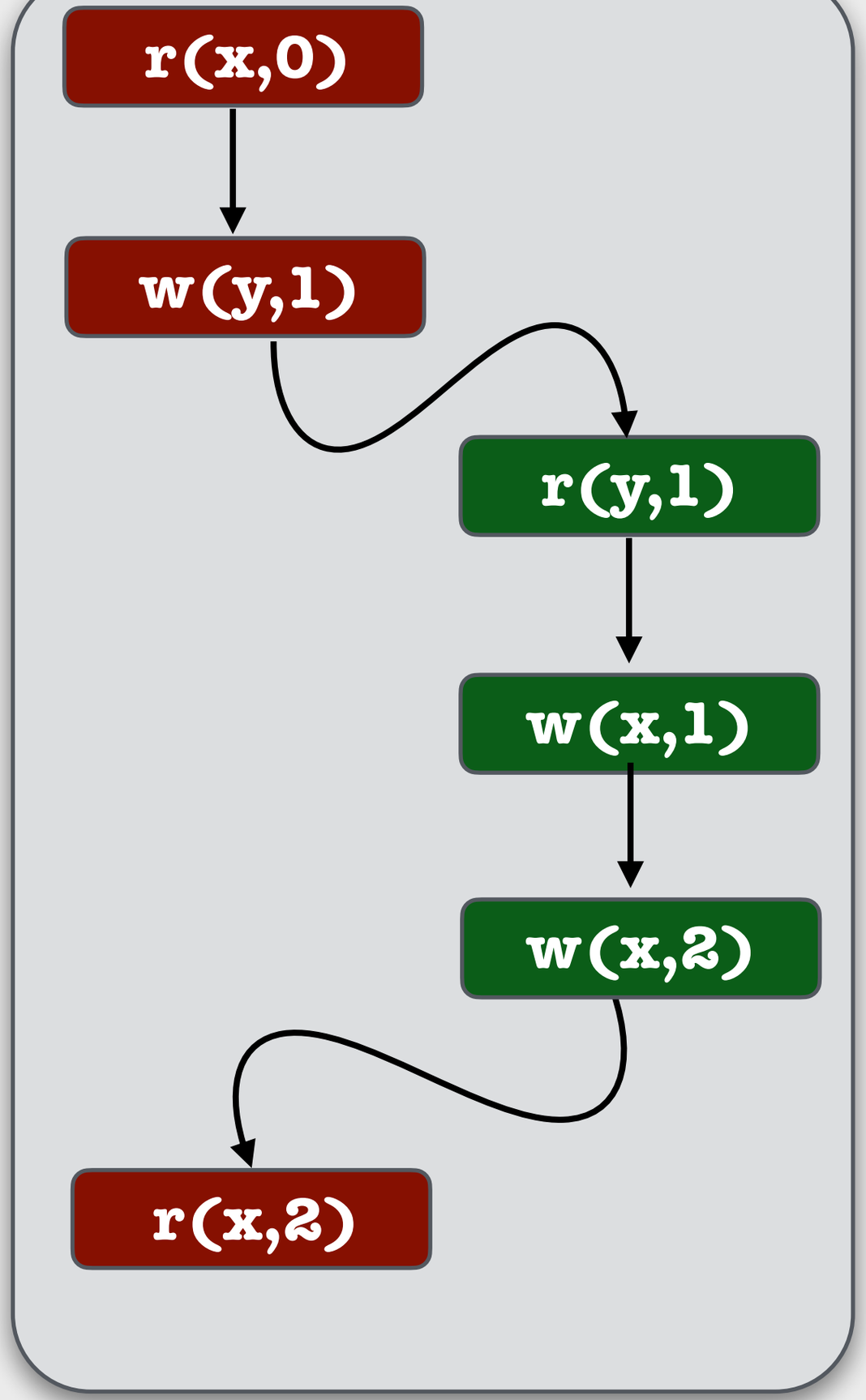
$y := 1$	
0	2

$x := 0$	
0	0

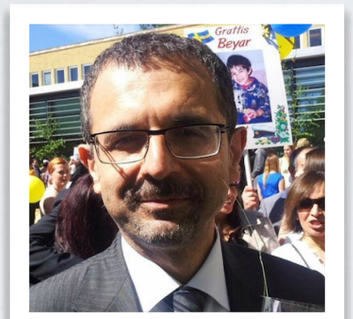
$y := 0$	
0	0

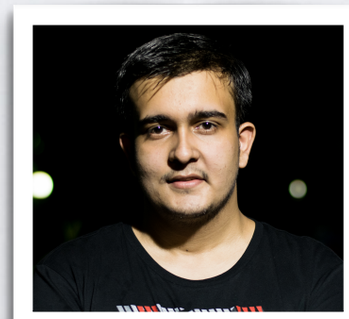
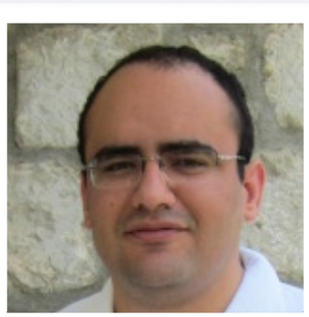
Reachable: $r1 =0, $r2 = 1 and $r3=2?

RA run



(Non parameterized) Reachability under RA




PLDI 2019

The Reachability Problem

Given a **program P** and a (control + memory) **state s**

- State Reachability Problem (Safety)

  Is **s** reachable in **P** under **RA**?

The Reachability Problem

Given a **program P** and a (control + memory) **state s**

- State Reachability Problem (Safety)

  Is **s** reachable in **P** under **RA**?

Decidability/ Complexity?

The Reachability Problem

Given a **program P** and a (control + memory) **state s**

- State Reachability Problem (Safety)

  Is **s** reachable in **P** under **RA**?

Decidability/ Complexity?

Each process is finite-state

The Reachability Problem

Given a **program P** and a (control + memory) **state s**

- State Reachability Problem (Safety)

  Is **s** reachable in **P** under **RA**?

Decidability/ Complexity?

Each process is finite-state

- For **SC**, the reachability problem is PSPACE-complete

The Reachability Problem

Given a **program P** and a (control + memory) **state s**

- State Reachability Problem (Safety)

  Is **s** reachable in **P** under **RA**?

Decidability/ Complexity?

Each process is finite-state

- For **SC**, the reachability problem is PSPACE-complete
- Nontrivial for **RA** since the set of paths is nonregular

The state reachability problem is undecidable for RA

- By reduction from the Post's correspondence Problem

	u	v
1.	b	bb
2.	$abba$	a
3.	ba	aa
4.	b	abb

214 is a solution

The state reachability problem is undecidable for RA

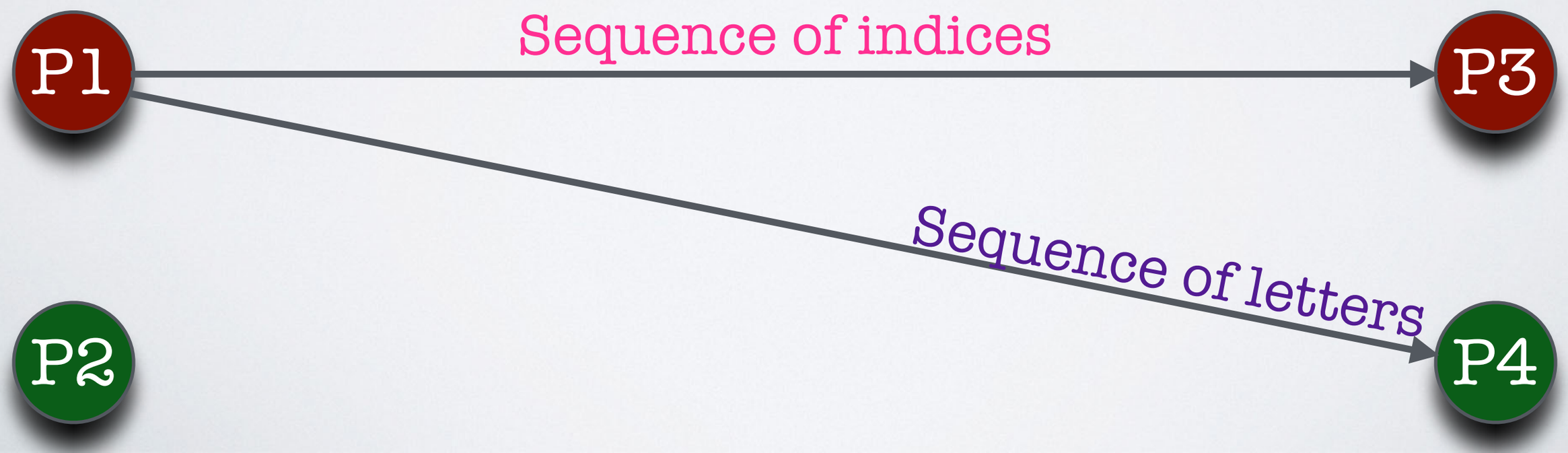
- By reduction from the Post's correspondence Problem

The state reachability problem is undecidable for RA

- By reduction from the Post's correspondence Problem

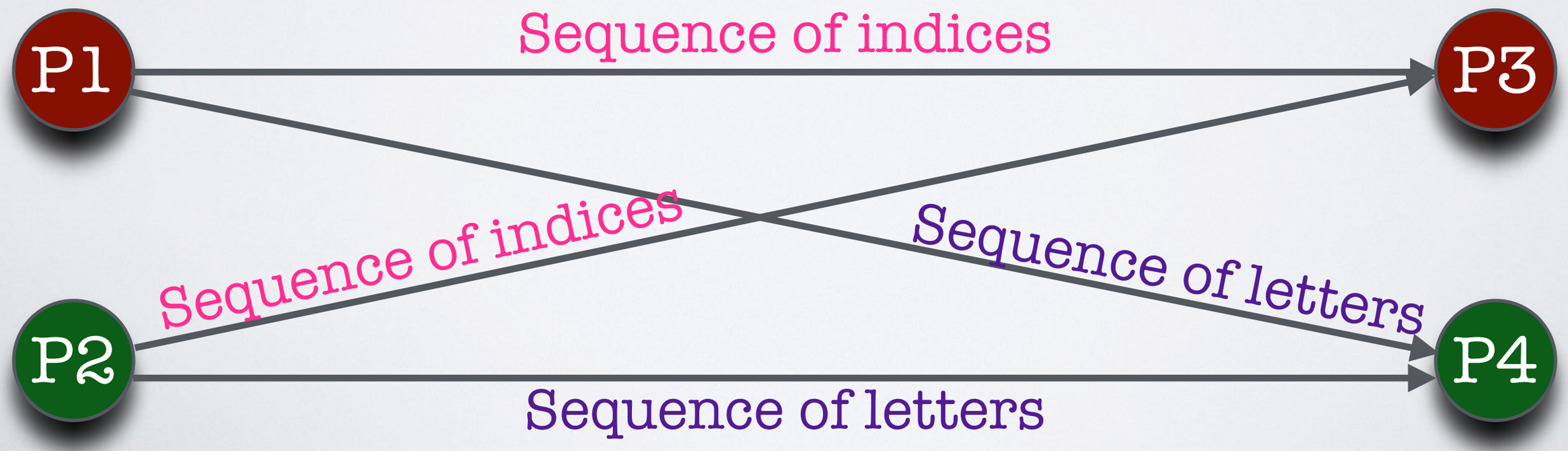
The state reachability problem is undecidable for RA

- By reduction from the Post's correspondence Problem

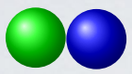
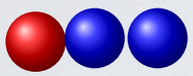
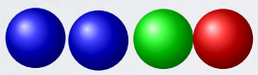
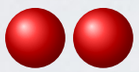


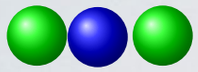
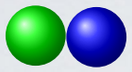
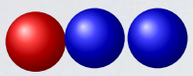
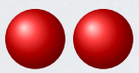
The state reachability problem is undecidable for RA

- By reduction from the Post's correspondence Problem



P1 P2 P4

	A	B
1		
2		
3		
4		

	A	B
1		
2		
3		
4		

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$y_1 = 1$

$x_2 =$ 

$x_1 =$ 

$x_2 =$ 

$y_2 = 2$

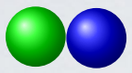
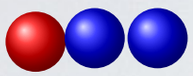
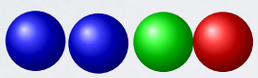
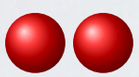
$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$y_1 = 3$

P1 P2 P4

	A	B
1		
2		
3		
4		

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$y_1 = 1$

$x_2 =$ 

$x_1 =$ 

$x_2 =$ 

$y_2 = 2$

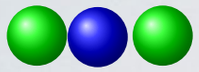
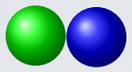
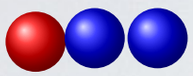
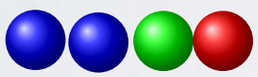
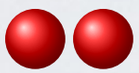
$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$y_1 = 3$

P1 P2 P4

	A	B
1		
2		
3		
4		

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$y_1 = 1$

$x_2 =$ 

$x_1 =$ 

$x_2 =$ 

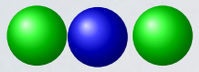
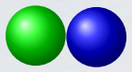
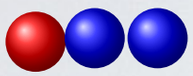
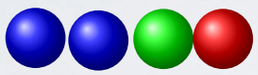
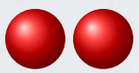
$y_2 = 2$

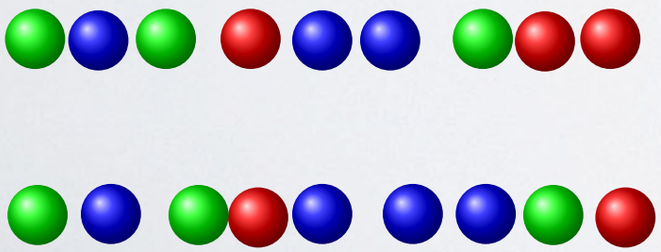
$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$y_1 = 3$

	A	B
1		
2		
3		
4		



P1

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$y_1 = 1$

$x_2 =$ 

$x_1 =$ 

$x_2 =$ 

$y_2 = 2$

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$y_1 = 3$

P2

$x_3 =$ 

$x_4 =$ 

$y_3 = 1$

$x_3 =$ 

$x_4 =$ 

$x_3 =$ 

$y_4 = 2$

$x_4 =$ 

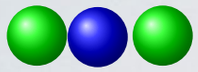
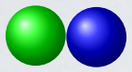
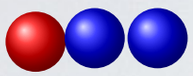
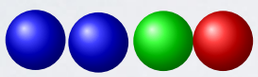
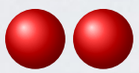
$x_3 =$ 

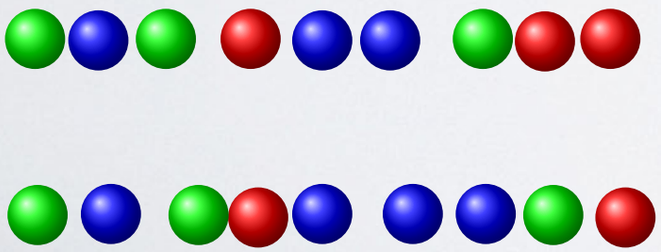
$x_4 =$ 

$x_3 =$ 

$y_3 = 3$

P4

	A	B
1		
2		
3		
4		



P1

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$x_2 =$ 

$x_1 =$

$x_2 =$

P2

$x_3 =$ 

$x_4 =$ 

$x_3 =$ 

$x_4 =$

$x_3 =$

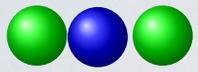
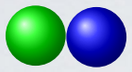
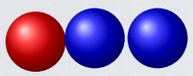
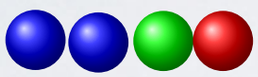
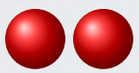
P4

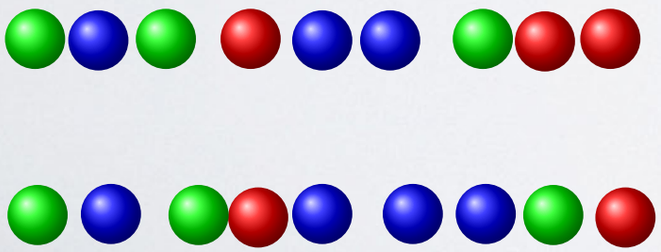
$x_4 =$

$x_3 =$

$x_4 =$

$x_3 =$

	A	B
1		
2		
3		
4		



P1

$x_1 =$ ✓

$x_2 =$

$x_1 =$

$x_2 =$

$x_1 =$

$x_2 =$

$x_1 =$

P2

$x_3 =$ ✓

$x_4 =$

$x_3 =$

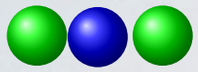
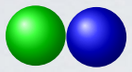
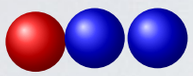
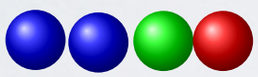
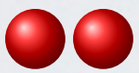
$x_4 =$

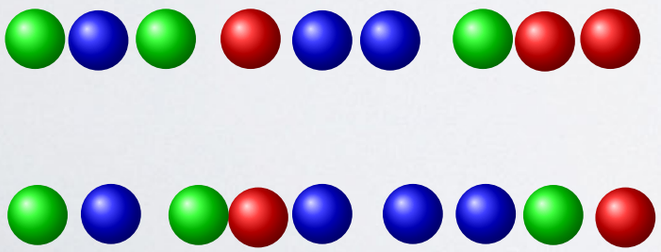
$x_3 =$

$x_4 =$

$x_3 =$

P4

	A	B
1		
2		
3		
4		



P1

$x_1 =$ ✓

$x_2 =$ ✓

$x_1 =$

$x_2 =$

$x_1 =$

$x_2 =$

P2

$x_3 =$ ✓

$x_4 =$ ✓

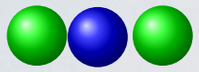
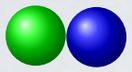
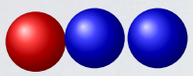
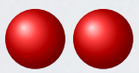
$x_3 =$

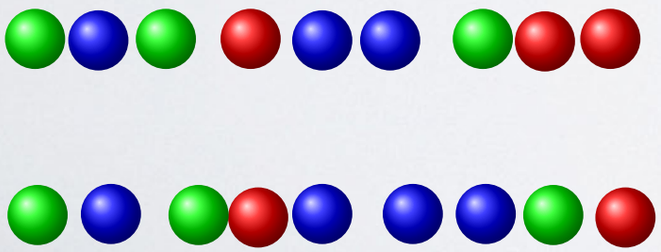
$x_4 =$

$x_4 =$

$x_3 =$

P4

	A	B
1		
2		
3		
4		



P1

$x_1 =$ ✓

$x_2 =$ ✓

$x_1 =$ ✓

$x_2 =$

$x_1 =$

$x_2 =$

$x_1 =$

$x_2 =$

$x_1 =$

P2

$x_3 =$ ✓

$x_4 =$ ✓

$x_3 =$ ✓

$x_4 =$

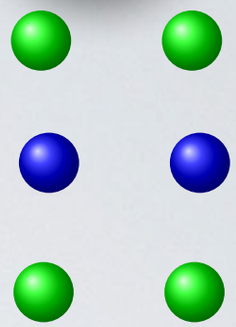
$x_3 =$

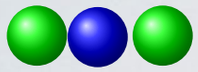
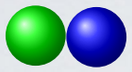
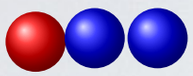
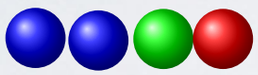
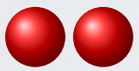
$x_4 =$

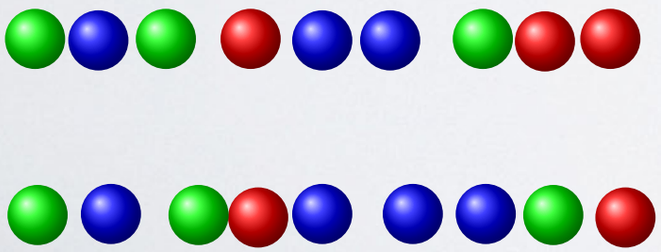
$x_3 =$

$x_4 =$

P4



	A	B
1		
2		
3		
4		



P1

$x_1 =$  ✓

$x_2 =$  ✓

$x_1 =$  ✓

$x_2 =$  ✓

$x_1 =$

$x_2 =$

P2

$x_3 =$  ✓

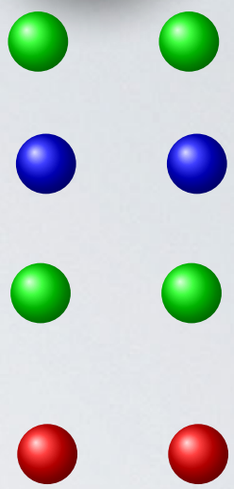
$x_4 =$  ✓

$x_3 =$  ✓

$x_4 =$ ✓

$x_3 =$

P4

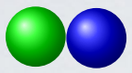
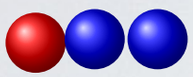
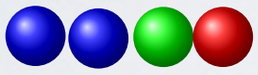


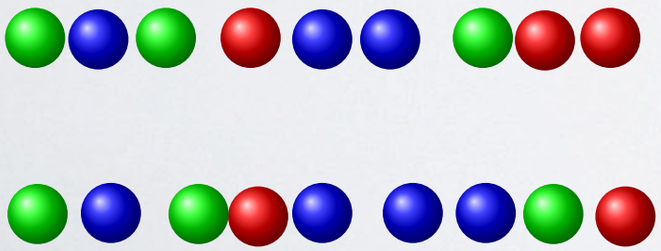
$x_4 =$

$x_3 =$

$x_4 =$

$x_3 =$

	A	B
1		
2		
3		
4		



P1

$x_1 =$ ✓
$x_2 =$ ✓
$x_1 =$ ✓

$x_2 =$ ✓
$x_1 =$ ✓
$x_2 =$

$x_1 =$
$x_2 =$
$x_1 =$

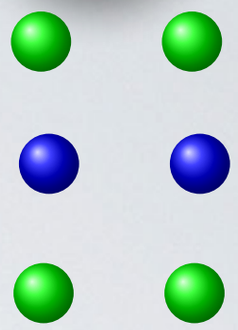
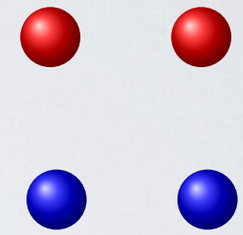
P2

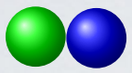
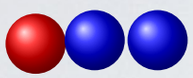
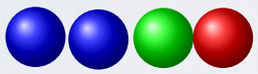
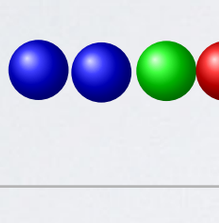
$x_3 =$ ✓
$x_4 =$ ✓

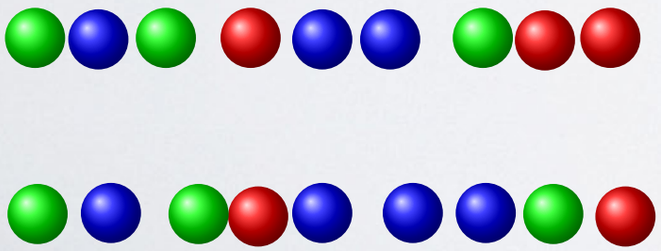
$x_3 =$ ✓
$x_4 =$ ✓
$x_3 =$ ✓

$x_4 =$
$x_3 =$
$x_4 =$
$x_3 =$

P4

	A	B
1		
2		
3		
4		



P1

$x_1 =$ ● ✓

$x_2 =$ ● ✓

$x_1 =$ ● ✓

$x_2 =$ ● ✓

$x_1 =$ ● ✓

$x_2 =$ ●

P2

$x_3 =$ ● ✓

$x_4 =$ ● ✓

$x_3 =$ ● ✓

$x_4 =$ ● ✓

$x_3 =$ ● ✓

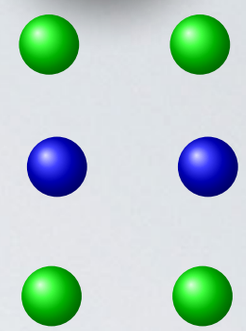
$x_4 =$ ●

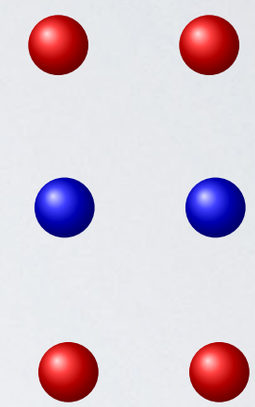
$x_3 =$ ●

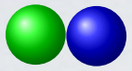
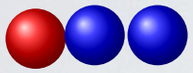
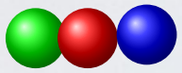
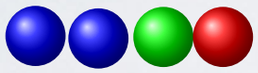
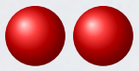
$x_4 =$ ●

$x_3 =$ ●

P4





	A	B
1		
2		
3		
4		

P1

$x_1 =$  ✓

$x_2 =$  ✓

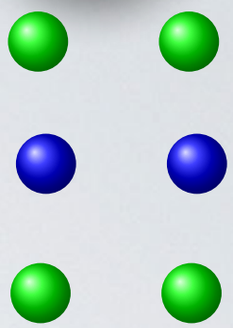
$x_1 =$  ✓

P2

$x_3 =$  ✓

$x_4 =$  ✓

P4



$x_2 =$

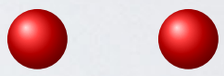
$x_1 =$

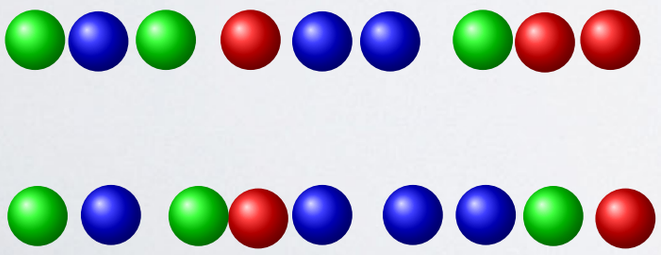
$x_2 =$ 

P4 jumps and cheats!

$=$  ✓

✓





$x_1 =$ 

$x_2 =$ 

$x_1 =$  ✓

$x_4 =$ 

$x_3 =$ 

$x_4 =$ 

$x_3 =$ 

✓

P1

P2

P4

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$x_2 =$ 

$x_1 =$ 

$x_3 =$ 

$x_4 =$ 

$x_3 =$ 

$x_4 =$ 

$x_3 =$ 

$x_4 =$ 

$x_3 =$ 

P1 P2 P4

t_1 $x_1 =$ $x_3 =$

t_2 $x_2 =$ $x_4 =$

t_3 $x_1 =$ $x_3 =$

t_4 $x_2 =$ $x_4 =$

t_5 $x_1 =$ $x_3 =$

t_6 $x_2 =$ $x_4 =$

t_7 $x_1 =$ $x_3 =$

P1 P2 P4

t_1 $x_1 =$ t_1' $x_3 =$

t_2 $x_2 =$ t_2' $x_4 =$

t_3 $x_1 =$ t_3' $x_3 =$

t_4 $x_2 =$ t_4' $x_4 =$

t_5 $x_1 =$ t_5' $x_3 =$

t_6 $x_2 =$ t_6' $x_4 =$

t_7 $x_1 =$ t_7' $x_3 =$

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ 

t_2 $x_2 =$ 

t_3 $x_1 =$ 

t_4 $x_2 =$ 

t_5 $x_1 =$ 

t_6 $x_2 =$ 

t_7 $x_1 =$ 

t'_1 $x_3 =$ 

t'_2 $x_4 =$ 

t'_3 $x_3 =$ 

t'_4 $x_4 =$ 

t'_5 $x_3 =$ 

t'_6 $x_4 =$ 

t'_7 $x_3 =$ 

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ●

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t_7 $x_1 =$ ●

t'_1 $x_3 =$ ●

t'_2 $x_4 =$ ●

t'_3 $x_3 =$ ●

t'_4 $x_4 =$ ●

t'_5 $x_3 =$ ●

t'_6 $x_4 =$ ●

t'_7 $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ●

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t_7 $x_1 =$ ●

t'_1 $x_3 =$ ● ✓

t'_2 $x_4 =$ ●

t'_3 $x_3 =$ ●

t'_4 $x_4 =$ ●

t'_5 $x_3 =$ ●

t'_6 $x_4 =$ ●

t'_7 $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0
t_1	0	t_1'	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ●

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t_7 $x_1 =$ ●

t_1' $x_3 =$ ● ✓

t_2' $x_4 =$ ●

t_3' $x_3 =$ ●

t_4' $x_4 =$ ●

t_5' $x_3 =$ ●

t_6' $x_4 =$ ●

t_7' $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0
t_1	0	t_1'	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ● ✓

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t_7 $x_1 =$ ●

t_1' $x_3 =$ ● ✓

t_2' $x_4 =$ ●

t_3' $x_3 =$ ●

t_4' $x_4 =$ ●

t_5' $x_3 =$ ●

t_6' $x_4 =$ ●

t_7' $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0
t_1	0	t_1'	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ● ✓

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t_7 $x_1 =$ ●

t_1' $x_3 =$ ● ✓

t_2' $x_4 =$ ● ✓

t_3' $x_3 =$ ●

t_4' $x_4 =$ ●

t_5' $x_3 =$ ●

t_6' $x_4 =$ ●

t_7' $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t'_1 $x_3 =$ ● ✓

● ●

t_1	0	t'_1	0

t_2 $x_2 =$ ● ✓

t'_2 $x_4 =$ ● ✓

● ●

t_1	t_2	t'_1	t'_2

t_3 $x_1 =$ ●

t'_3 $x_3 =$ ●

t_4 $x_2 =$ ●

t'_4 $x_4 =$ ●

t_5 $x_1 =$ ●

t'_5 $x_3 =$ ●

t_6 $x_2 =$ ●

t'_6 $x_4 =$ ●

t_7 $x_1 =$ ●

t'_7 $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0
t_1	0	t'_1	0
t_1	t_2	t'_1	t'_2

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ● ✓

t_3 $x_1 =$ ●

t_4 $x_2 =$ ● ✓

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t_7 $x_1 =$ ●

t'_1 $x_3 =$ ● ✓

t'_2 $x_4 =$ ● ✓

t'_3 $x_3 =$ ●

t'_4 $x_4 =$ ●

t'_5 $x_3 =$ ●

t'_6 $x_4 =$ ●

t'_7 $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t'_1 $x_3 =$ ● ✓

● ●

t_1	0	t'_1	0

t_2 $x_2 =$ ● ✓

t'_2 $x_4 =$ ● ✓

● ●

t_1	t_2	t'_1	t'_2

t_3 $x_1 =$ ●

t'_3 $x_3 =$ ●

t_4 $x_2 =$ ● ✓

t'_4 $x_4 =$ ● ✓

t_5 $x_1 =$ ●

t'_5 $x_3 =$ ●

t_6 $x_2 =$ ●

t'_6 $x_4 =$ ●

t_7 $x_1 =$ ●

t'_7 $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t'_1 $x_3 =$ ● ✓

● ●

t_1	0	t'_1	0

t_2 $x_2 =$ ● ✓

t'_2 $x_4 =$ ● ✓

● ●

t_1	t_2	t'_1	t'_2

t_3 $x_1 =$ ●

t'_3 $x_3 =$ ●

● ●

	t_4		t'_4

t_4 $x_2 =$ ● ✓

t'_4 $x_4 =$ ● ✓

t_5 $x_1 =$ ●

t'_5 $x_3 =$ ●

t_6 $x_2 =$ ●

t'_6 $x_4 =$ ●

t_7 $x_1 =$ ●

t'_7 $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t'_1 $x_3 =$ ● ✓

● ●

t_1	0	t'_1	0

t_2 $x_2 =$ ● ✓

t'_2 $x_4 =$ ● ✓

● ●

t_1	t_2	t'_1	t'_2

t_3 $x_1 =$ ●

t'_3 $x_3 =$ ●

● ●

t_3	t_4	t'_3	t'_4

t_4 $x_2 =$ ● ✓

t'_4 $x_4 =$ ● ✓

t_5 $x_1 =$ ●

t'_5 $x_3 =$ ●

t_6 $x_2 =$ ●

t'_6 $x_4 =$ ●

t_7 $x_1 =$ ●

t'_7 $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t'_1 $x_3 =$ ● ✓

● ●

t_1	0	t'_1	0

t_2 $x_2 =$ ● ✓

t'_2 $x_4 =$ ● ✓

● ●

t_1	t_2	t'_1	t'_2

t_3 $x_1 =$ ●

t'_3 $x_3 =$ ●

● ●

t_3	t_4	t'_3	t'_4

t_4 $x_2 =$ ● ✓

t'_4 $x_4 =$ ● ✓

t_5 $x_1 =$ ●

t'_5 $x_3 =$ ●

t_6 $x_2 =$ ●

t'_6 $x_4 =$ ●

t_7 $x_1 =$ ● ✓

t'_7 $x_3 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t'_1 $x_3 =$ ● ✓

● ●

t_1	0	t'_1	0

t_2 $x_2 =$ ● ✓

t'_2 $x_4 =$ ● ✓

● ●

t_1	t_2	t'_1	t'_2

t_3 $x_1 =$ ●

t'_3 $x_3 =$ ●

● ●

t_3	t_4	t'_3	t'_4

t_4 $x_2 =$ ● ✓

t'_4 $x_4 =$ ● ✓

t_5 $x_1 =$ ●

t'_5 $x_3 =$ ●

t_6 $x_2 =$ ●

t'_6 $x_4 =$ ●

t_7 $x_1 =$ ● ✓

t'_7 $x_3 =$ ● ✓

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t'_1 $x_3 =$ ● ✓

● ●

t_1	0	t'_1	0

t_2 $x_2 =$ ● ✓

t'_2 $x_4 =$ ● ✓

● ●

t_1	t_2	t'_1	t'_2

t_3 $x_1 =$ ●

t'_3 $x_3 =$ ●

● ●

t_3	t_4	t'_3	t'_4

t_4 $x_2 =$ ● ✓

t'_4 $x_4 =$ ● ✓

t_5 $x_1 =$ ●

t'_5 $x_3 =$ ●

t_6 $x_2 =$ ●

t'_6 $x_4 =$ ●

● ●

t_7		t'_7	

t_7 $x_1 =$ ● ✓

t'_7 $x_3 =$ ● ✓

P1 P2 P4

x_1 x_2 x_3 x_4

0	0	0	0

t_1 $x_1 =$ ● ✓ t'_1 $x_3 =$ ● ✓

● ●

t_1	0	t'_1	0

t_2 $x_2 =$ ● ✓ t'_2 $x_4 =$ ● ✓

● ●

t_1	t_2	t'_1	t'_2

t_3 $x_1 =$ ● t'_3 $x_3 =$ ●

● ●

t_3	t_4	t'_3	t'_4

t_4 $x_2 =$ ● ✓ t'_4 $x_4 =$ ● ✓

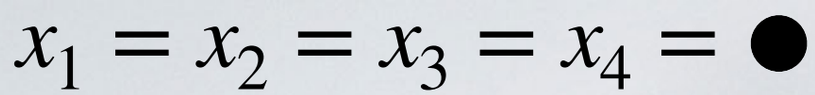
t_5 $x_1 =$ ● t'_5 $x_3 =$ ●

t_6 $x_2 =$ ● t'_6 $x_4 =$ ●

● ●

t_7	t_6	t'_7	t'_6

t_7 $x_1 =$ ● ✓ t'_7 $x_3 =$ ● ✓

$x_1 = x_2 = x_3 = x_4 =$ 

P4

P1

P2

t_1 $x_1 =$ 

t_1' $x_3 =$ 

t_2 $x_2 =$ 

t_2' $x_4 =$ 

t_3 $x_1 =$ 

t_3' $x_3 =$ 

t_4 $x_2 =$ 

t_4' $x_4 =$ 

t_5 $x_1 =$ 

t_5' $x_3 =$ 

t_6 $x_2 =$ 

t_6' $x_4 =$ 

$x_1 = x_2 = x_3 = x_4 =$ ●

P4

x_1	x_2	x_3	x_4
0	0	0	0

P1

P2

t_1 $x_1 =$ ●

t_2 $x_2 =$ ●

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t'_1 $x_3 =$ ●

t'_2 $x_4 =$ ●

t'_3 $x_3 =$ ●

t'_4 $x_4 =$ ●

t'_5 $x_3 =$ ●

t'_6 $x_4 =$ ●

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

arw(x_1, ● ●)

t_1	0	0	0

t_1 $x_1 =$ ● ✓

t_1' $x_3 =$ ●

t_2 $x_2 =$ ●

t_2' $x_4 =$ ●

t_3 $x_1 =$ ●

t_3' $x_3 =$ ●

t_4 $x_2 =$ ●

t_4' $x_4 =$ ●

t_5 $x_1 =$ ●

t_5' $x_3 =$ ●

t_6 $x_2 =$ ●

t_6' $x_4 =$ ●

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

arw(x_1, ● ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ●

t_2 $x_2 =$ ● t_2' $x_4 =$ ●

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ●

t_2 $x_2 =$ ● t_2' $x_4 =$ ●

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ●

t_2 $x_2 =$ ● t_2' $x_4 =$ ●

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ●

t_2 $x_2 =$ ● t_2' $x_4 =$ ●

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ●

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t'_1 $x_3 =$ ● ✓

t'_2 $x_4 =$ ●

t'_3 $x_3 =$ ●

t'_4 $x_4 =$ ●

t'_5 $x_3 =$ ●

t'_6 $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

arw(x_3, ● ●)

t_1	0	0	0
$t_1 + 1$	0	0	0
$t_1 + 1$	0	t'_1	0

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ●

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t'_1 $x_3 =$ ● ✓

t'_2 $x_4 =$ ●

t'_3 $x_3 =$ ●

t'_4 $x_4 =$ ●

t'_5 $x_3 =$ ●

t'_6 $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

arw(x_3, ● ●)

t_1	0	0	0
$t_1 + 1$	0	0	0
$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ●

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t'_1 $x_3 =$ ● ✓

t'_2 $x_4 =$ ●

t'_3 $x_3 =$ ●

t'_4 $x_4 =$ ●

t'_5 $x_3 =$ ●

t'_6 $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

arw(x_3, ● ●)
r(x_4, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ●

t_1' $x_3 =$ ● ✓

t_2' $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

arw(x_3, ● ●)
r(x_4, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0
$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t_3' $x_3 =$ ●

t_4' $x_4 =$ ●

t_5' $x_3 =$ ●

t_6' $x_4 =$ ●

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓

t_2 $x_2 =$ ●

t_3 $x_1 =$ ●

t_4 $x_2 =$ ●

t_5 $x_1 =$ ●

t_6 $x_2 =$ ●

t'_1 $x_3 =$ ● ✓

t'_2 $x_4 =$ ●

t'_3 $x_3 =$ ●

t'_4 $x_4 =$ ●

t'_5 $x_3 =$ ●

t'_6 $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

arw(x_3, ● ●)
r(x_4, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ $\quad t_1'$ $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ $\quad t_2'$ $x_4 =$ ●

t_3 $x_1 =$ ● $\quad t_3'$ $x_3 =$ ●

t_4 $x_2 =$ ● $\quad t_4'$ $x_4 =$ ●

t_5 $x_1 =$ ● $\quad t_5'$ $x_3 =$ ●

t_6 $x_2 =$ ● $\quad t_6'$ $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

arw(x_2, ● ●)

$t_1 + 1$	t_2	$t_1' + 1$	0

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t'_1 $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t'_2 $x_4 =$ ●

t_3 $x_1 =$ ● t'_3 $x_3 =$ ●

t_4 $x_2 =$ ● t'_4 $x_4 =$ ●

t_5 $x_1 =$ ● t'_5 $x_3 =$ ●

t_6 $x_2 =$ ● t'_6 $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

arw(x_2, ● ●)

$t_1 + 1$	t_2	$t'_1 + 1$	0
$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	0

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ $\quad t'_1$ $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ $\quad t'_2$ $x_4 =$ ●

t_3 $x_1 =$ ● $\quad t'_3$ $x_3 =$ ●

t_4 $x_2 =$ ● $\quad t'_4$ $x_4 =$ ●

t_5 $x_1 =$ ● $\quad t'_5$ $x_3 =$ ●

t_6 $x_2 =$ ● $\quad t'_6$ $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t'_1 + 1$	0
$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	0

$x_1 = x_2 = x_3 = x_4 = \bullet$

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 = \bullet$ ✓ t_1' $x_3 = \bullet$ ✓

t_2 $x_2 = \bullet$ ✓ t_2' $x_4 = \bullet$

t_3 $x_1 = \bullet$ t_3' $x_3 = \bullet$

t_4 $x_2 = \bullet$ t_4' $x_4 = \bullet$

t_5 $x_1 = \bullet$ t_5' $x_3 = \bullet$

t_6 $x_2 = \bullet$ t_6' $x_4 = \bullet$

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t_1' + 1$	0
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	0

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ $\quad t'_1$ $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ $\quad t'_2$ $x_4 =$ ●

t_3 $x_1 =$ ● $\quad t'_3$ $x_3 =$ ●

t_4 $x_2 =$ ● $\quad t'_4$ $x_4 =$ ●

t_5 $x_1 =$ ● $\quad t'_5$ $x_3 =$ ●

t_6 $x_2 =$ ● $\quad t'_6$ $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t'_1 + 1$	0
$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	0

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t'_1 $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t'_2 $x_4 =$ ● ✓

t_3 $x_1 =$ ● t'_3 $x_3 =$ ●

t_4 $x_2 =$ ● t'_4 $x_4 =$ ●

t_5 $x_1 =$ ● t'_5 $x_3 =$ ●

t_6 $x_2 =$ ● t'_6 $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t'_1 + 1$	0
$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	0

arw(x_4, ● ●)

$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	t'_2

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t_2' $x_4 =$ ● ✓

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t_1' + 1$	0
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	0

arw(x_4, ● ●)

$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	t_2'
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	$t_2' + 1$

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t'_1 $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t'_2 $x_4 =$ ● ✓

t_3 $x_1 =$ ● t'_3 $x_3 =$ ●

t_4 $x_2 =$ ● t'_4 $x_4 =$ ●

t_5 $x_1 =$ ● t'_5 $x_3 =$ ●

t_6 $x_2 =$ ● t'_6 $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t'_1 + 1$	0
$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	0

arw(x_4, ● ●)
r(x_3, ●)

$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	t'_2
$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	$t'_2 + 1$

$x_1 = x_2 = x_3 = x_4 = \bullet$

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 = \bullet$ ✓ t_1' $x_3 = \bullet$ ✓

t_2 $x_2 = \bullet$ ✓ t_2' $x_4 = \bullet$ ✓

t_3 $x_1 = \bullet$ t_3' $x_3 = \bullet$

t_4 $x_2 = \bullet$ t_4' $x_4 = \bullet$

t_5 $x_1 = \bullet$ t_5' $x_3 = \bullet$

t_6 $x_2 = \bullet$ t_6' $x_4 = \bullet$

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t_1' + 1$	0
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	0

arw(x_4, ● ●)
r(x_3, ●)

$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	t_2'
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	$t_2' + 1$

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t'_1 $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t'_2 $x_4 =$ ● ✓

t_3 $x_1 =$ ● t'_3 $x_3 =$ ●

t_4 $x_2 =$ ● t'_4 $x_4 =$ ●

t_5 $x_1 =$ ● t'_5 $x_3 =$ ●

t_6 $x_2 =$ ● t'_6 $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t'_1	0
$t_1 + 1$	0	$t'_1 + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t'_1 + 1$	0
$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	0

arw(x_4, ● ●)
r(x_3, ●)

$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	t'_2
$t_1 + 1$	$t_2 + 1$	$t'_1 + 1$	$t'_2 + 1$

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t_2' $x_4 =$ ● ✓

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● ✓ t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t_1' + 1$	0
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	0

arw(x_4, ● ●)
r(x_3, ●)

$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	t_2'
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	$t_2' + 1$

arw(x_1, ● ●)

t_5	t_4	$t_1' + 1$	$t_2' + 1$

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t_2' $x_4 =$ ● ✓

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● ✓ t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t_1' + 1$	0
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	0

arw(x_4, ● ●)
r(x_3, ●)

$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	t_2'
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	$t_2' + 1$

arw(x_1, ● ●)

t_5	t_4	$t_1' + 1$	$t_2' + 1$
$t_5 + 1$	t_4	$t_1' + 1$	$t_2' + 1$

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t_2' $x_4 =$ ● ✓

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● ✓ t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t_1' + 1$	0
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	0

arw(x_4, ● ●)
r(x_3, ●)

$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	t_2'
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	$t_2' + 1$

arw(x_1, ● ●)

t_5	t_4	$t_1' + 1$	$t_2' + 1$
$t_5 + 1$	t_4	$t_1' + 1$	$t_2' + 1$

r(x_2, ●)

$x_1 = x_2 = x_3 = x_4 =$ ●

P1

P2

P4

x_1	x_2	x_3	x_4
0	0	0	0

t_1 $x_1 =$ ● ✓ t_1' $x_3 =$ ● ✓

t_2 $x_2 =$ ● ✓ t_2' $x_4 =$ ● ✓

t_3 $x_1 =$ ● t_3' $x_3 =$ ●

t_4 $x_2 =$ ● t_4' $x_4 =$ ●

t_5 $x_1 =$ ● ✓ t_5' $x_3 =$ ●

t_6 $x_2 =$ ● t_6' $x_4 =$ ●

arw(x_1, ● ●)
r(x_2, ●)

t_1	0	0	0
$t_1 + 1$	0	0	0

arw(x_3, ● ●)
r(x_4, ●)

$t_1 + 1$	0	t_1'	0
$t_1 + 1$	0	$t_1' + 1$	0

arw(x_2, ● ●)
r(x_1, ●)

$t_1 + 1$	t_2	$t_1' + 1$	0
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	0

arw(x_4, ● ●)
r(x_3, ●)

$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	t_2'
$t_1 + 1$	$t_2 + 1$	$t_1' + 1$	$t_2' + 1$

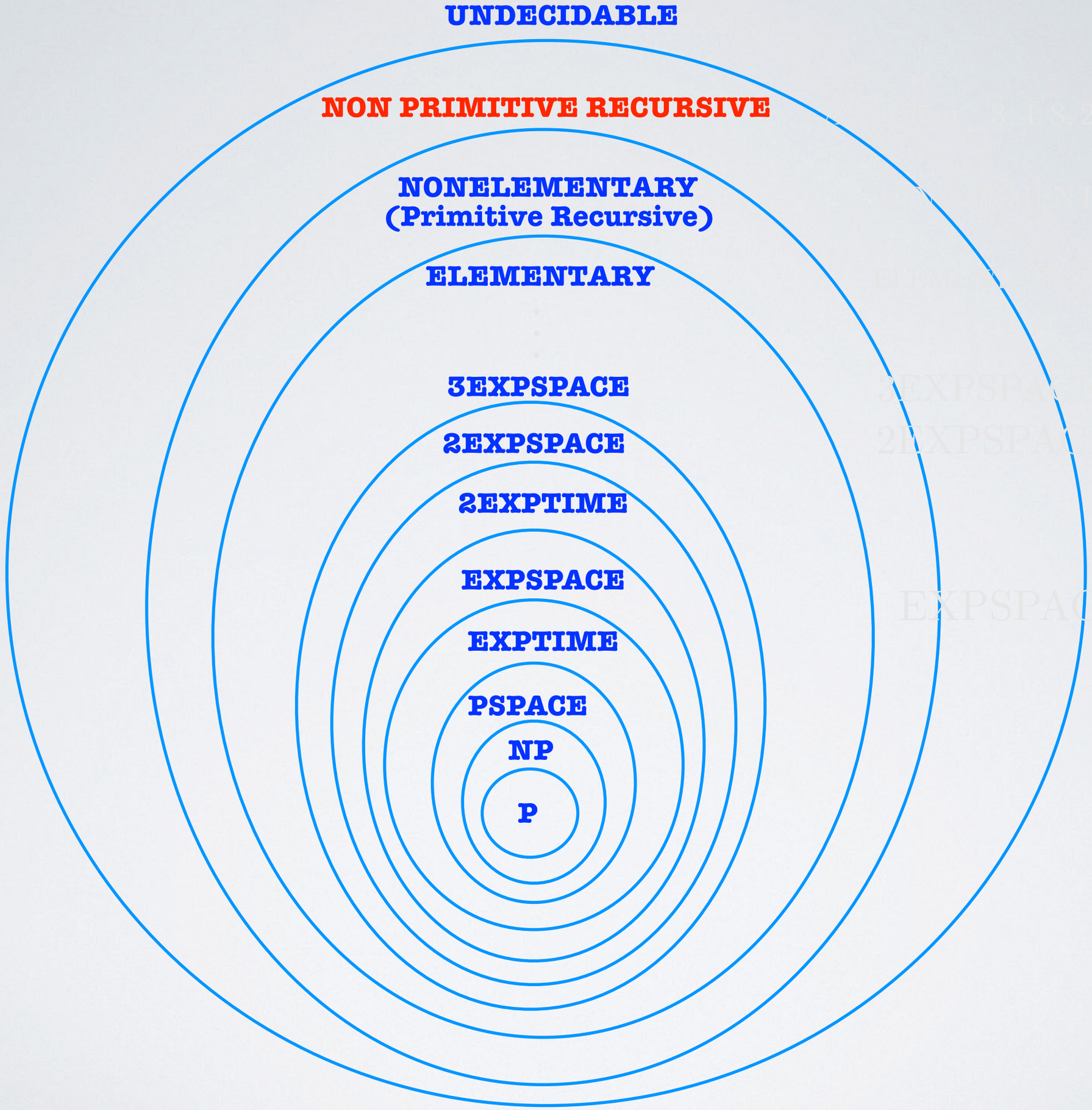
arw(x_1, ● ●)

t_5	t_4	$t_1' + 1$	$t_2' + 1$
$t_5 + 1$	t_4	$t_1' + 1$	$t_2' + 1$

r(x_2, ●)

$t_4 > t_2 + 1$

RA without arw


UNDECIDABLE
NON PRIMITIVE RECURSIVE
NONELEMENTARY
(Primitive Recursive)
ELEMENTARY
3EXPSPACE
2EXPSPACE
2EXPTIME
EXPSPACE
EXPTIME
PSPACE
NP
P

UNDECIDABLE

NON PRIMITIVE RECURSIVE

NONELEMENTARY
(Primit… …rsive)

Reachability of RA without arw

OPEN

P

Context-bounded Analysis (CBA)

Context-bounded Analysis (CBA)

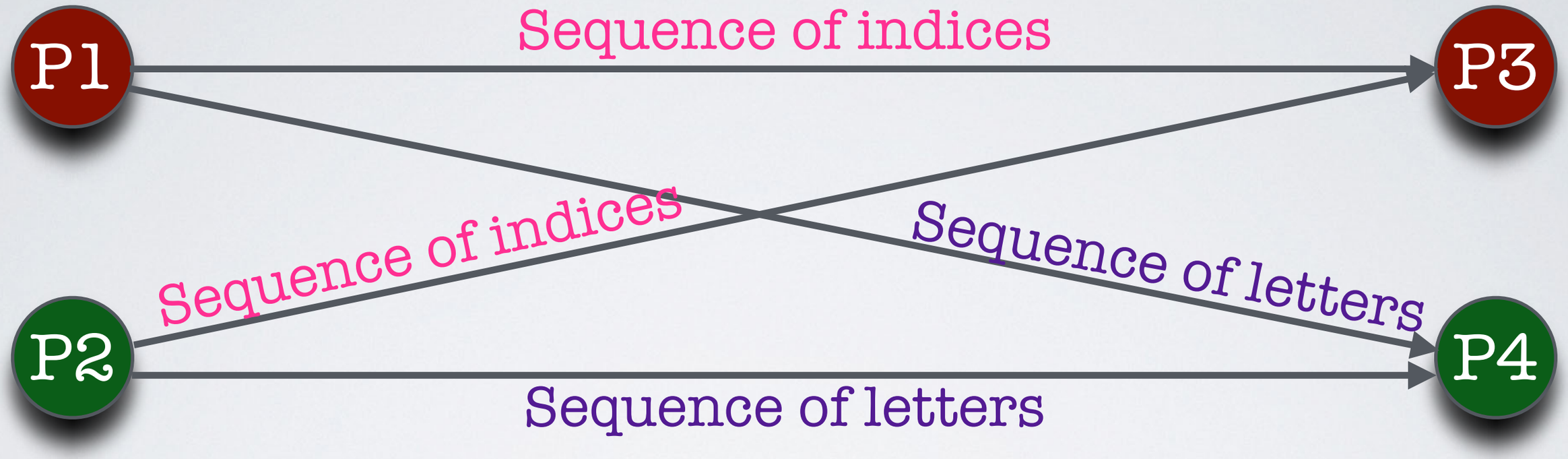
- ✦ Efficient under-approximation technique for SC [Qadeer et al. 2005, Lal et al. 2009, Torre et al. 2009]

- ✦ A context denotes one active process in a run

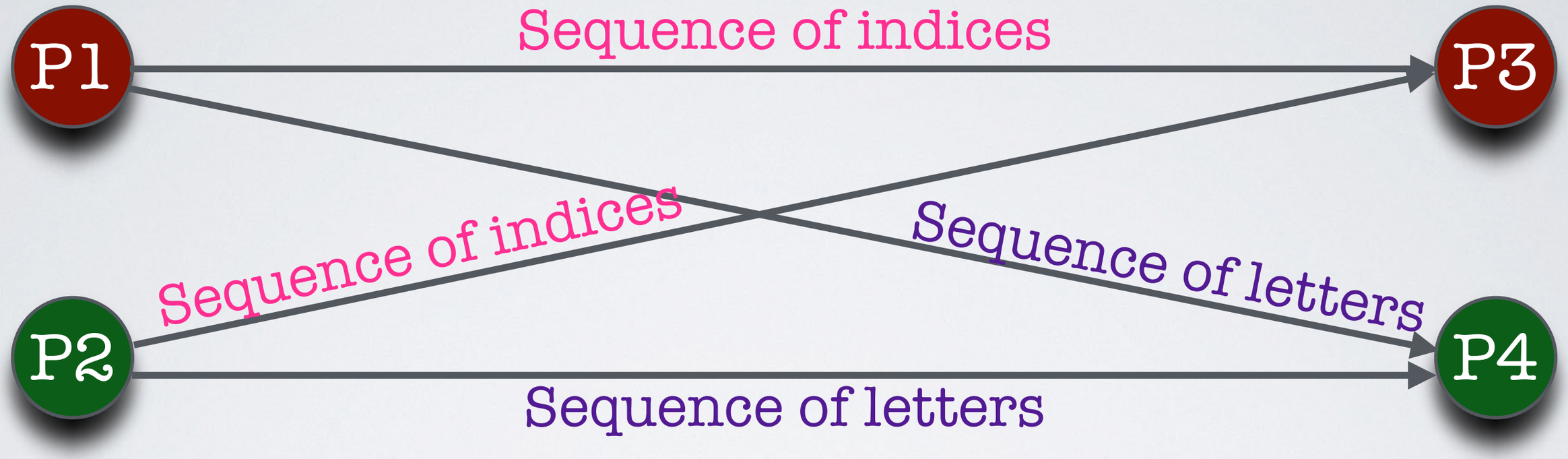
- ✦ For instance, $(P_1 P_2)^*$ has an unbounded context switch, while $P_1 P_2^*$ is 1-context bounded

  - Several tools: CHESS, Corral, CSeq, etc

Context-bounded Analysis (CBA)



Context-bounded Analysis (CBA)

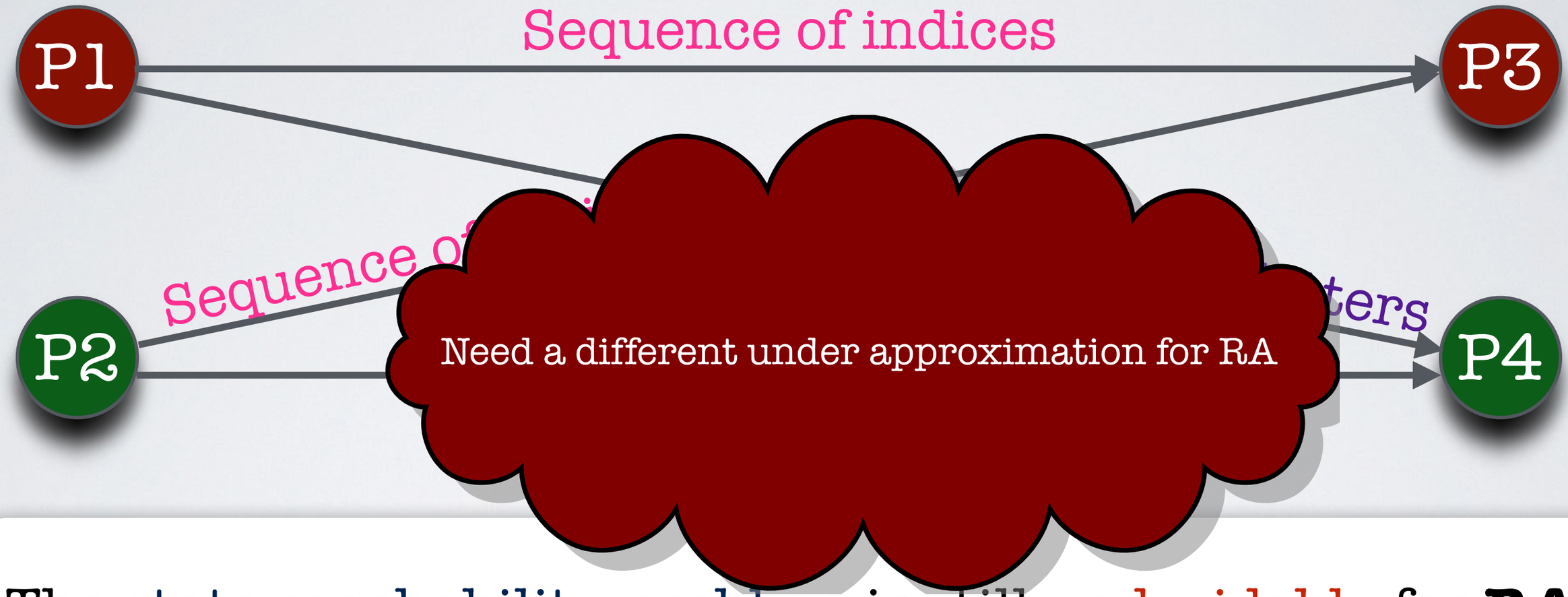


The state reachability problem is still undecidable for **RA** with a bounded number (3) of context switches

(context: only one "**active**" process)

P1 runs; P2 runs; P3 runs; P4 runs

Context-bounded Analysis (CBA)

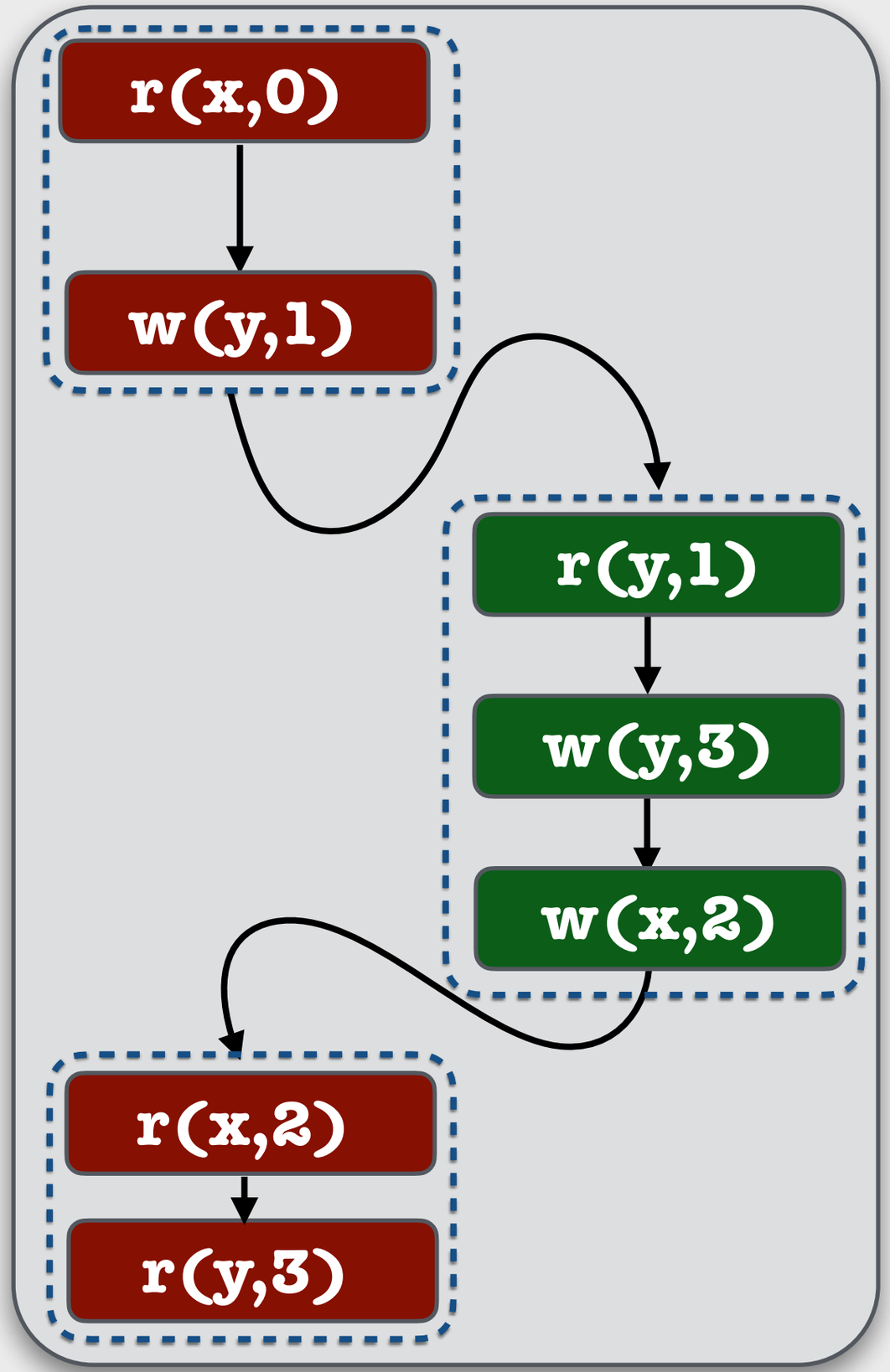


The state reachability problem is still undecidable for **RA** with a bounded number (3) of context switches

(context: only one "**active**" process)

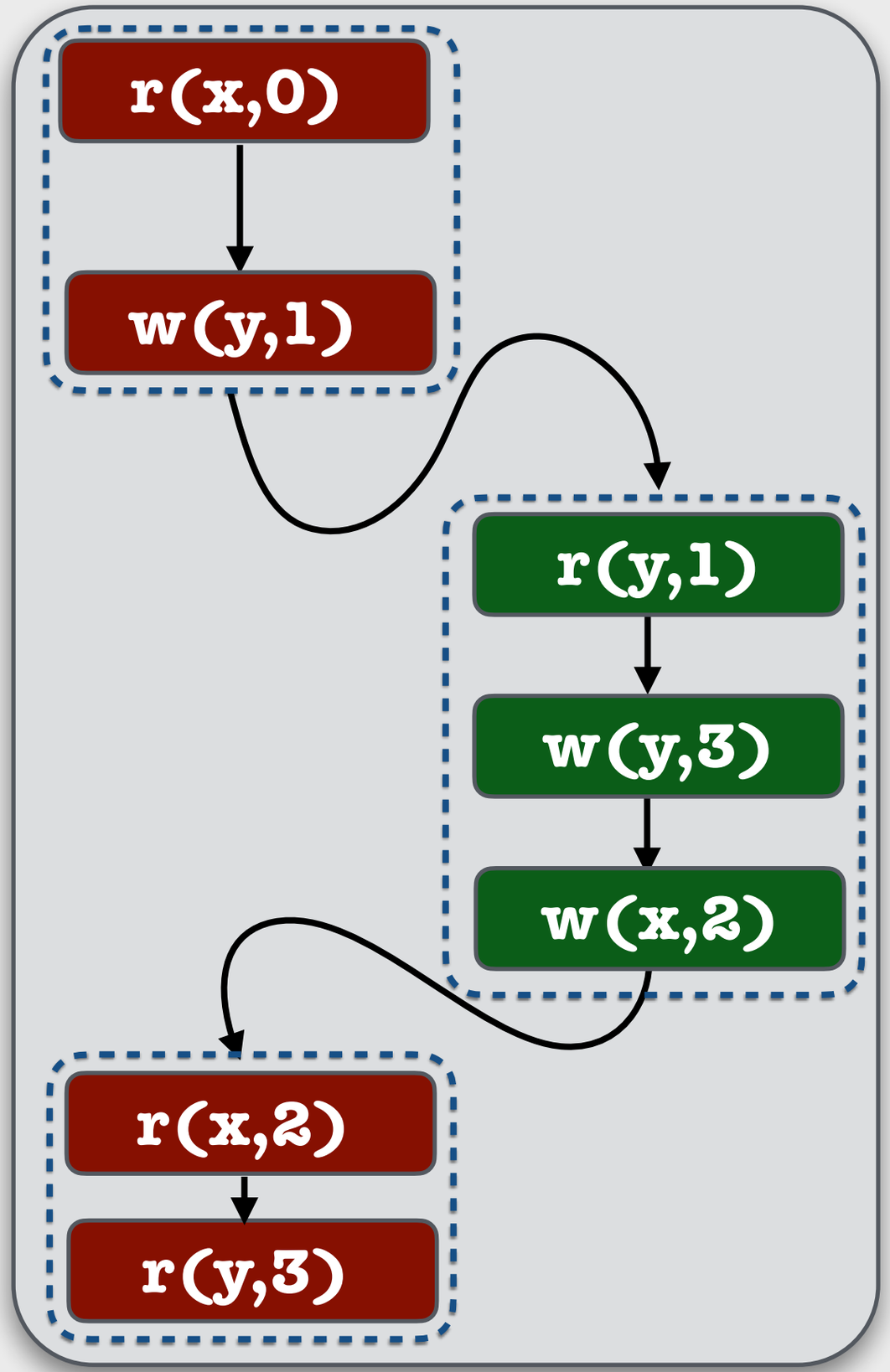
P1 runs; P2 runs; P3 runs; P4 runs

View Switch



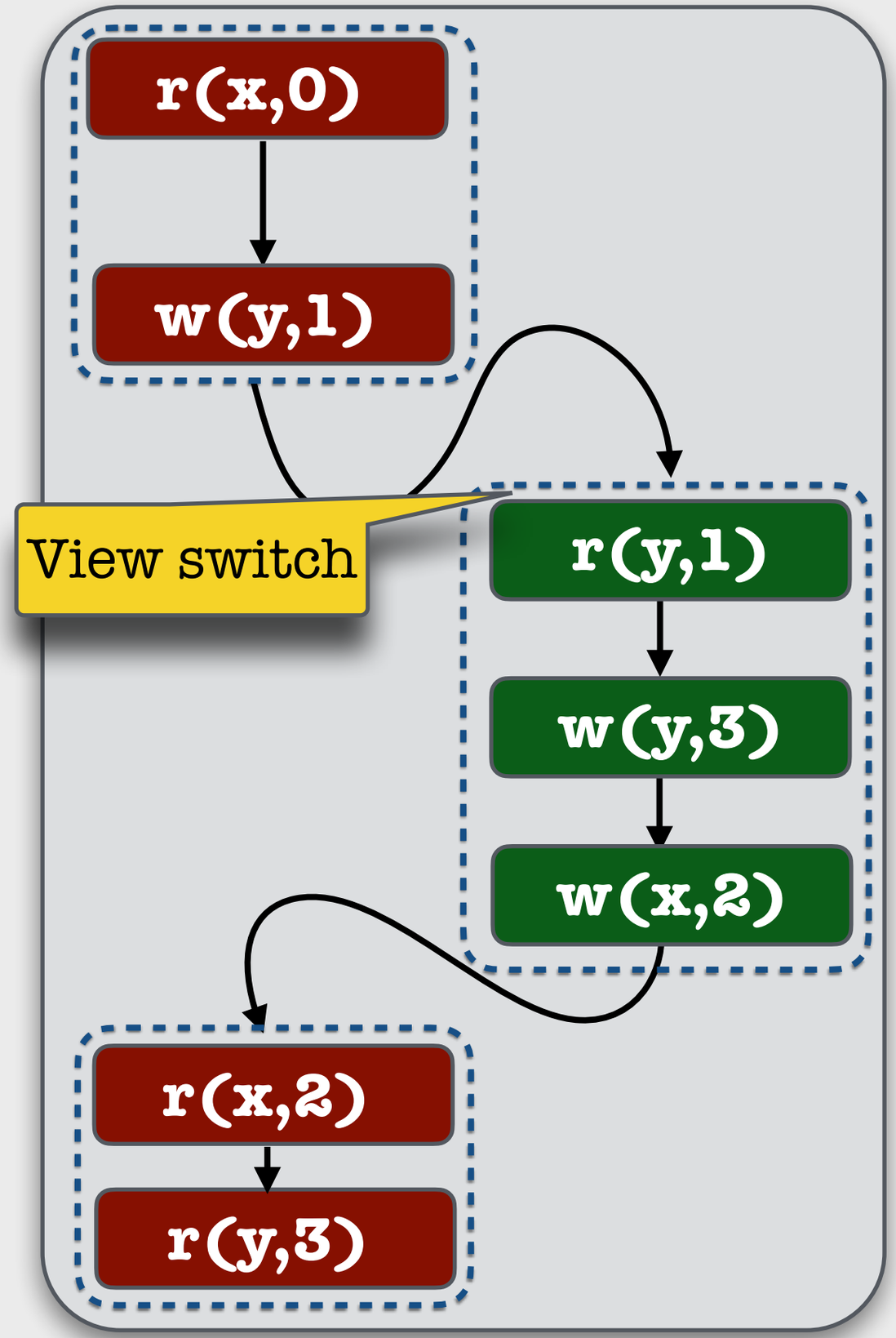
View Switch

A view-switch happens when a process reads a value written by another process, and changes its view



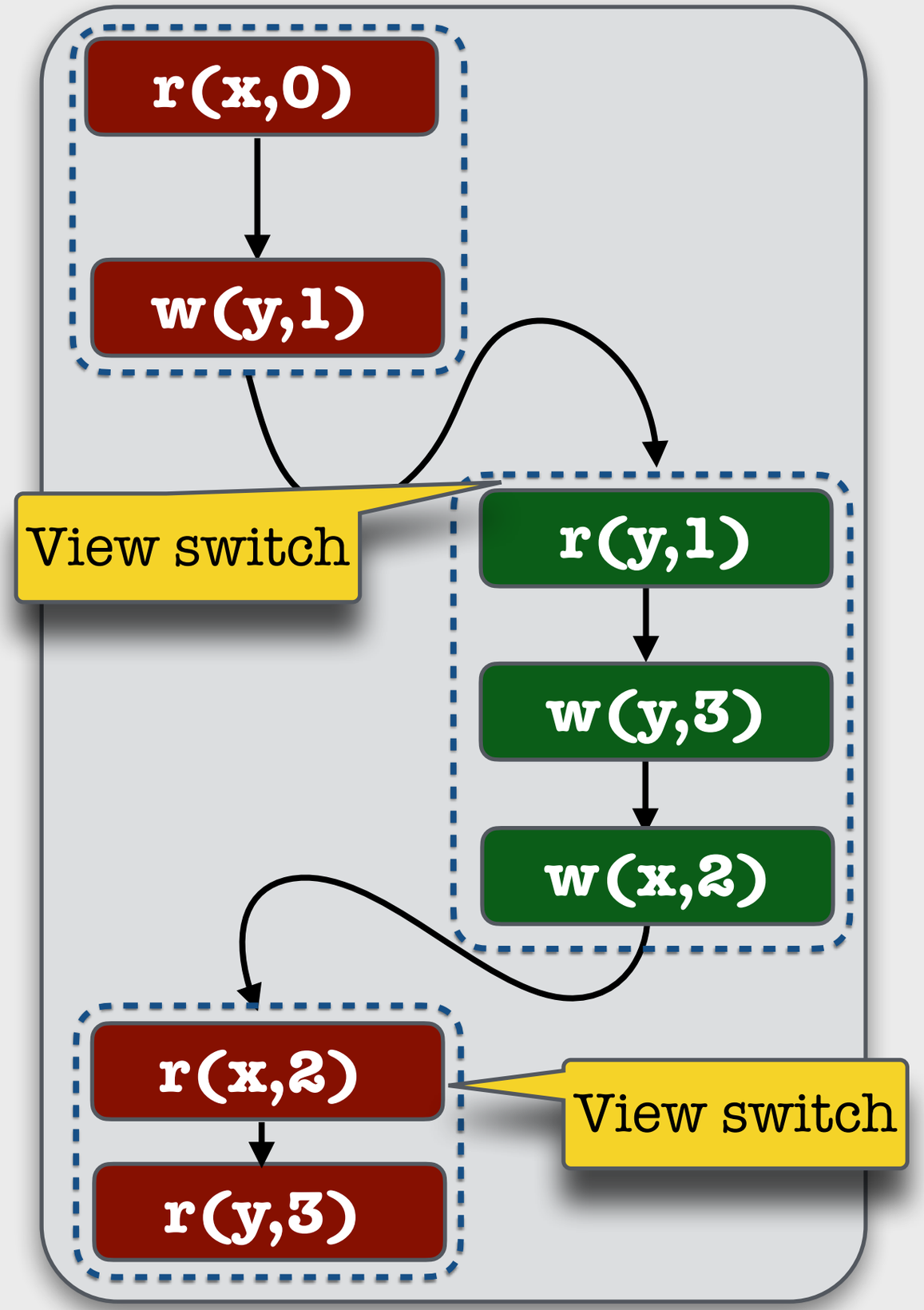
View Switch

A view-switch happens when a process reads a value written by another process, and changes its view



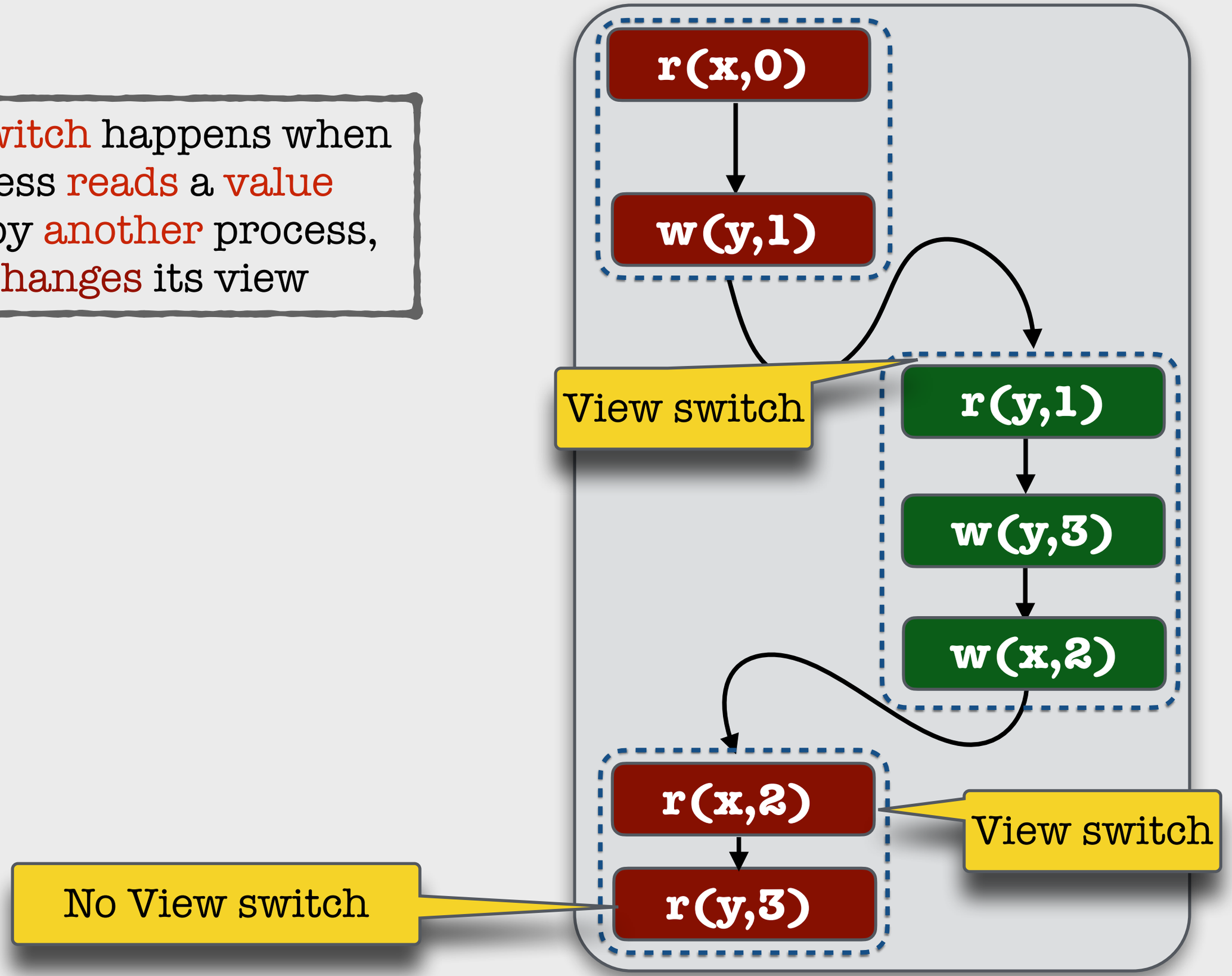
View Switch

A view-switch happens when a process reads a value written by another process, and changes its view



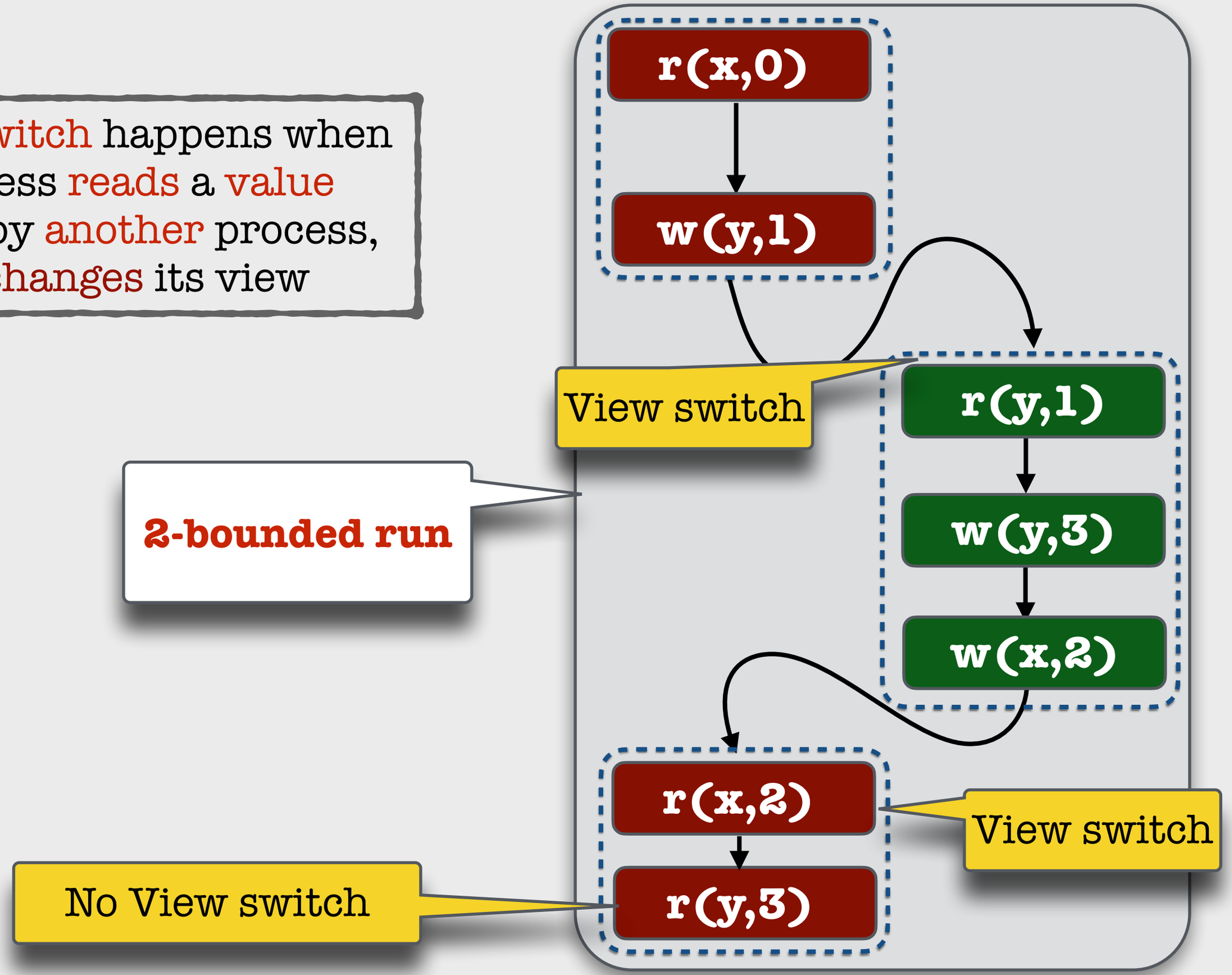
View Switch

A view-switch happens when a process reads a value written by another process, and changes its view



View Switch

A view-switch happens when a process reads a value written by another process, and changes its view



$x := 2$	
5	3

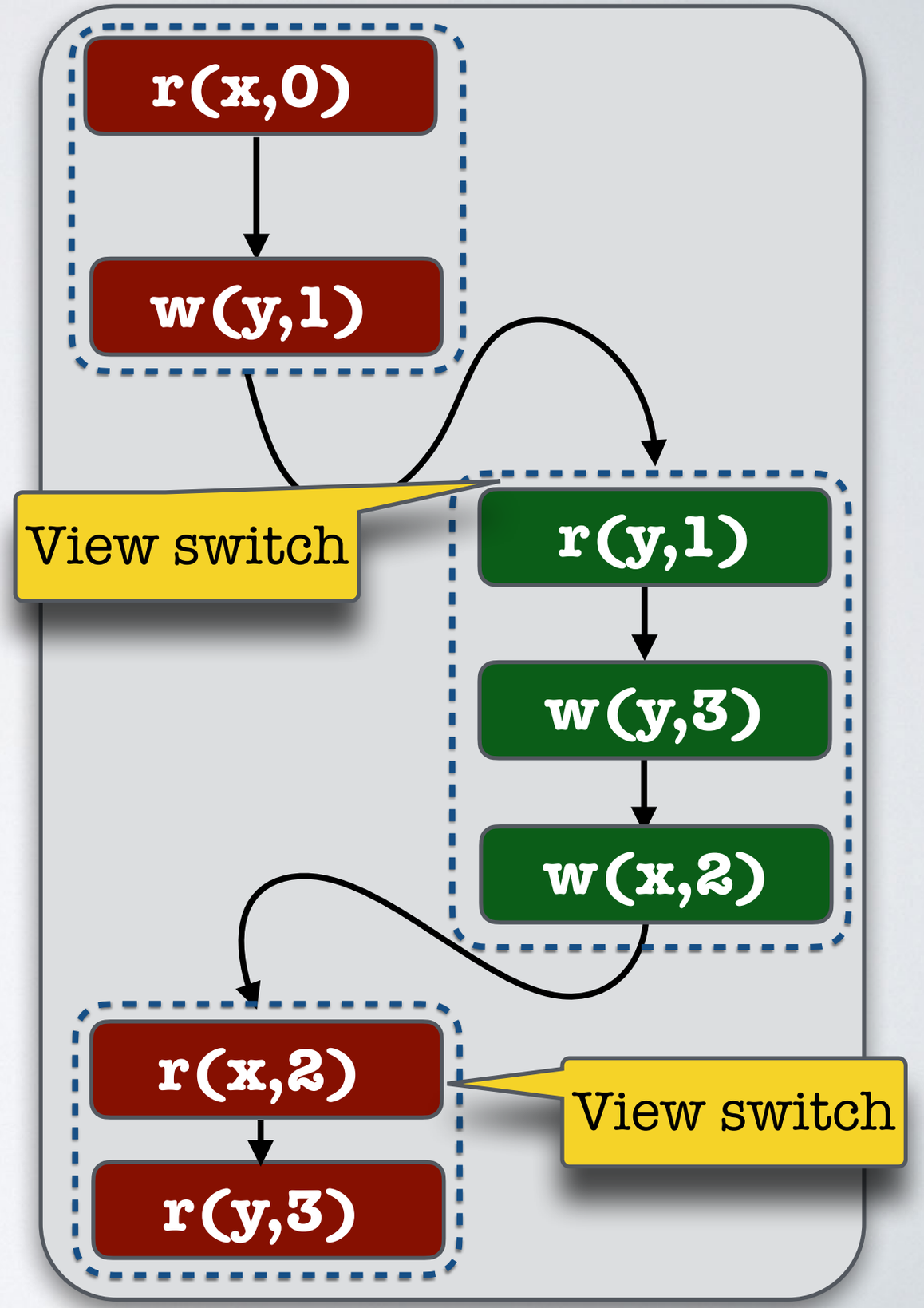
$x := 1$	
1	2

$y := 3$	
0	3

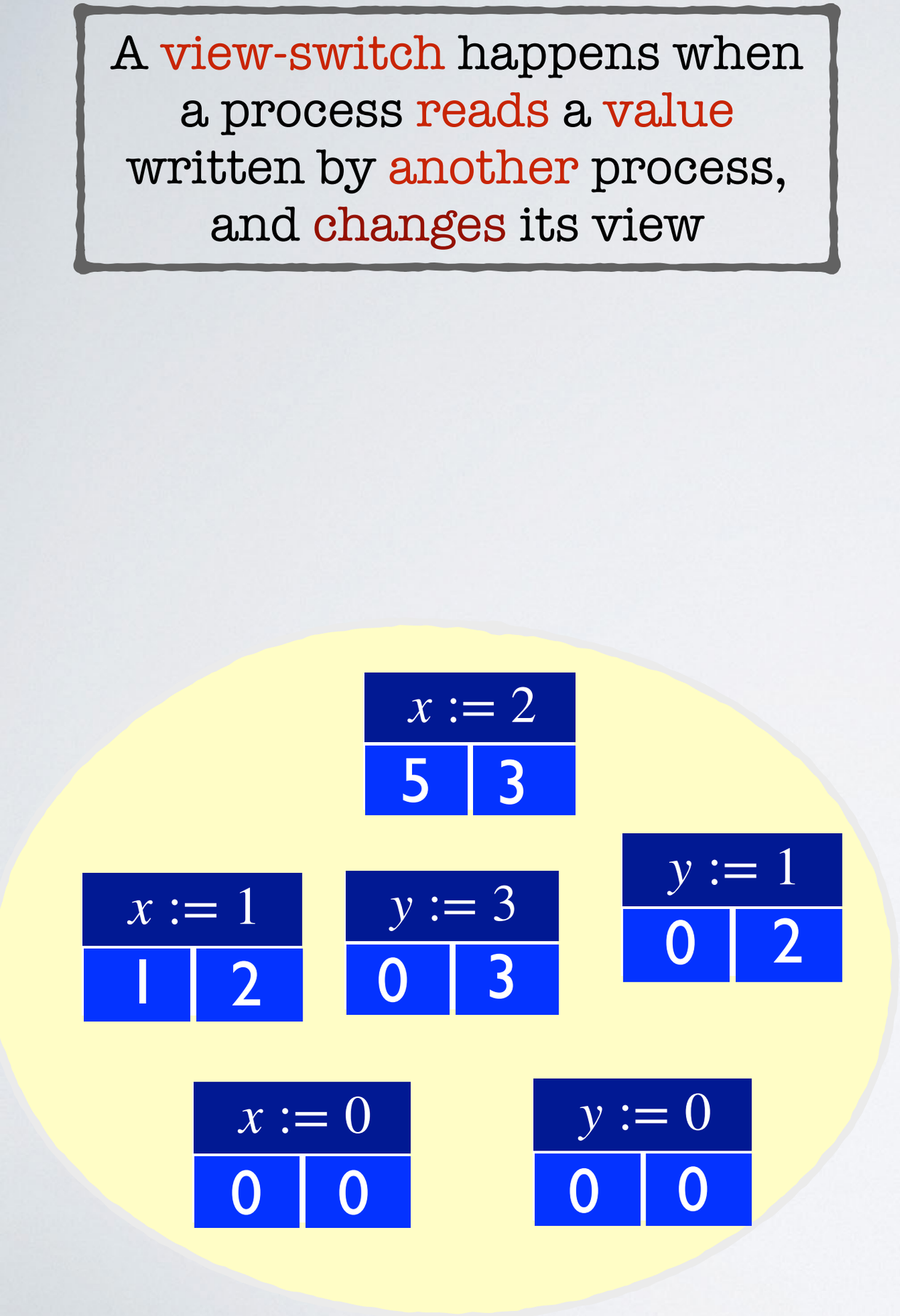
$y := 1$	
0	2

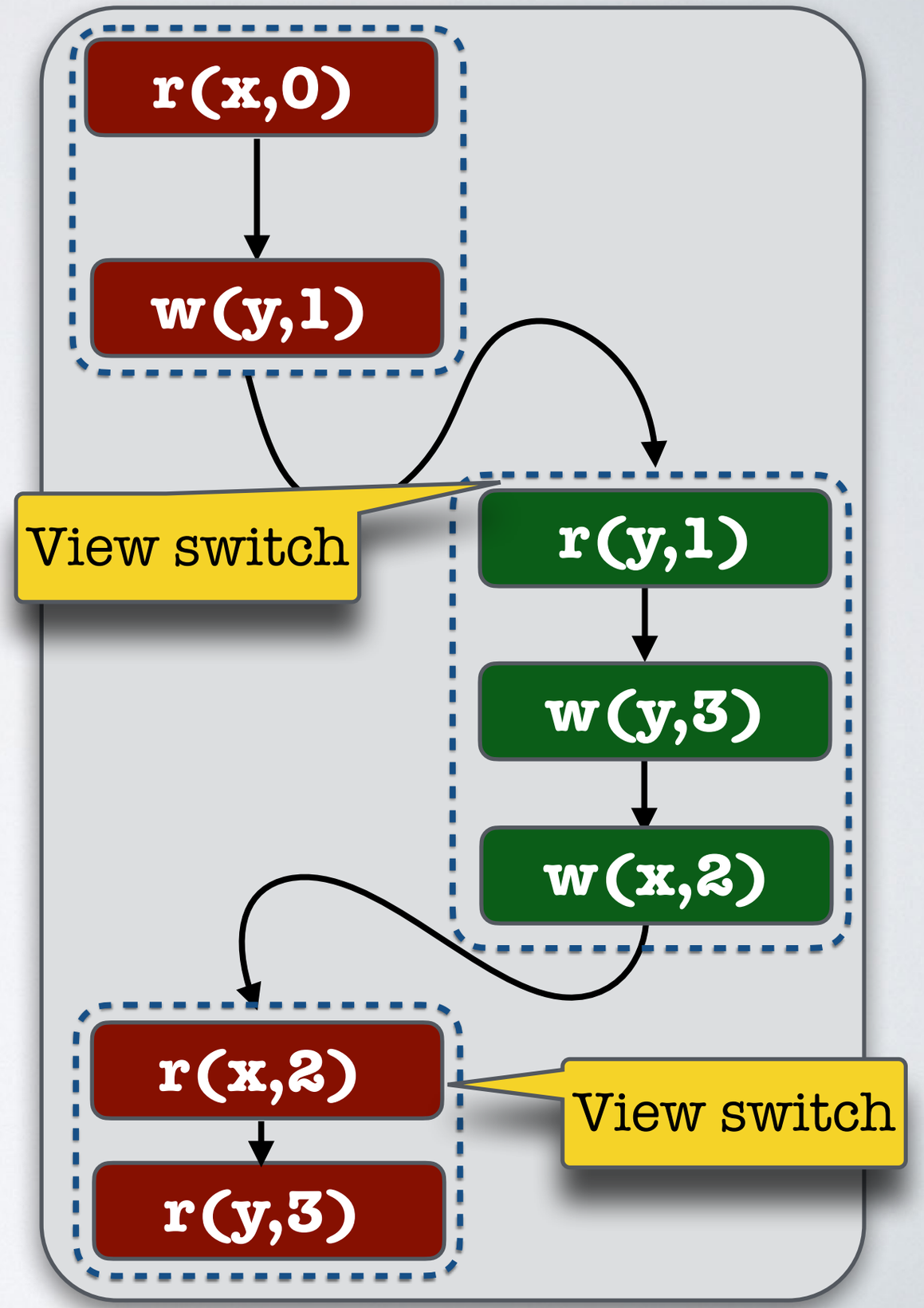
$x := 0$	
0	0

$y := 0$	
0	0



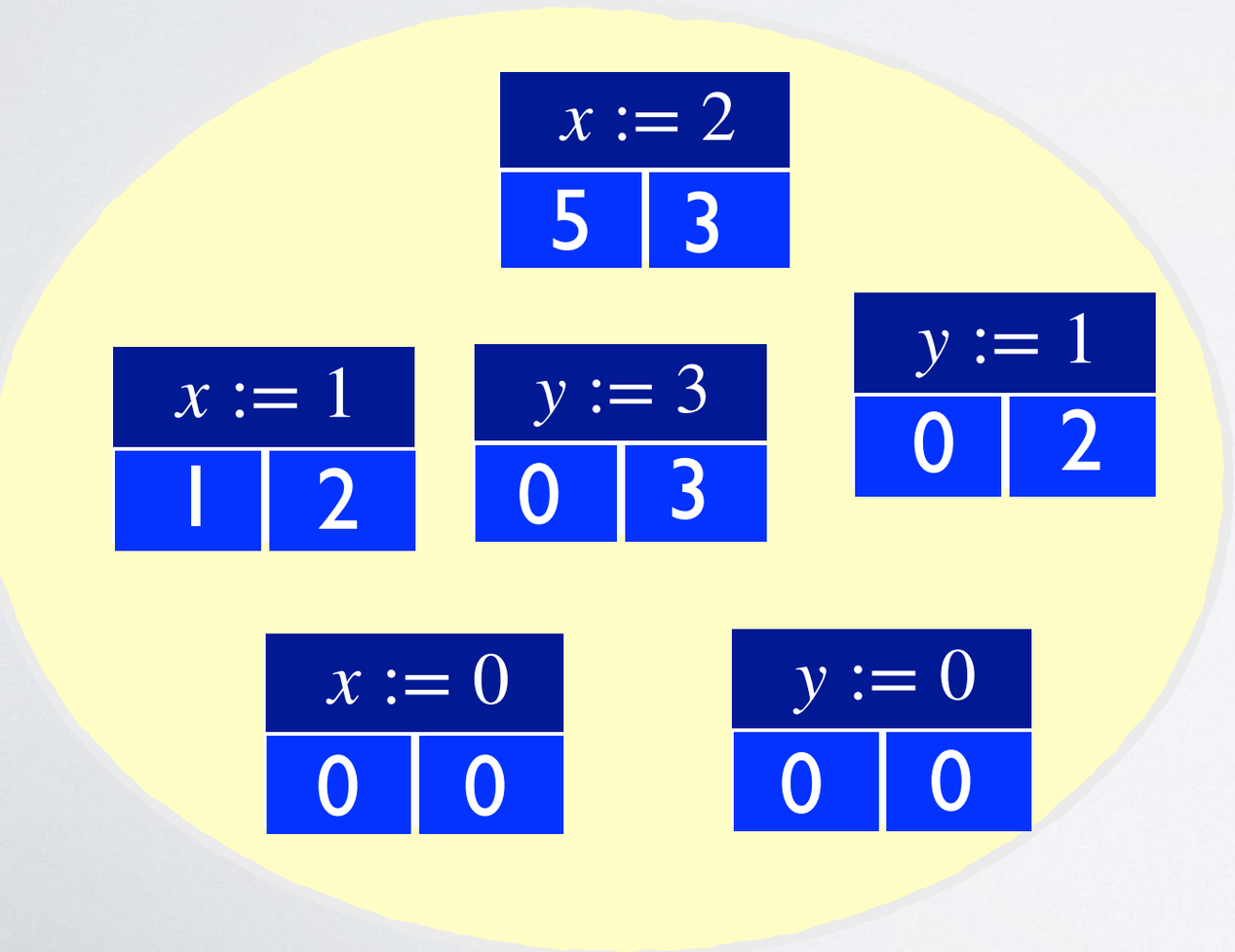
A view-switch happens when a process reads a value written by another process, and changes its view

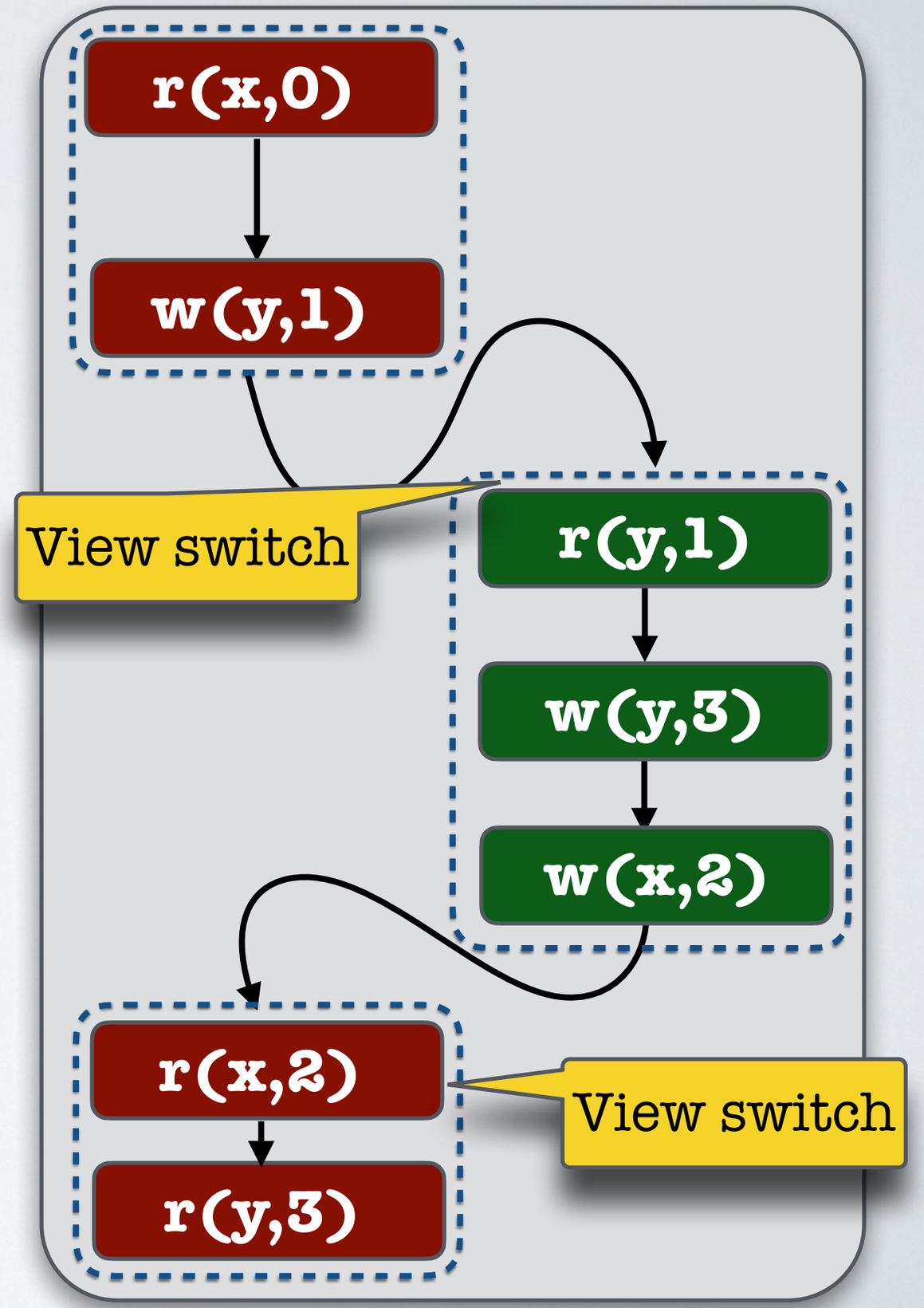




A view-switch happens when a process reads a value written by another process, and changes its view

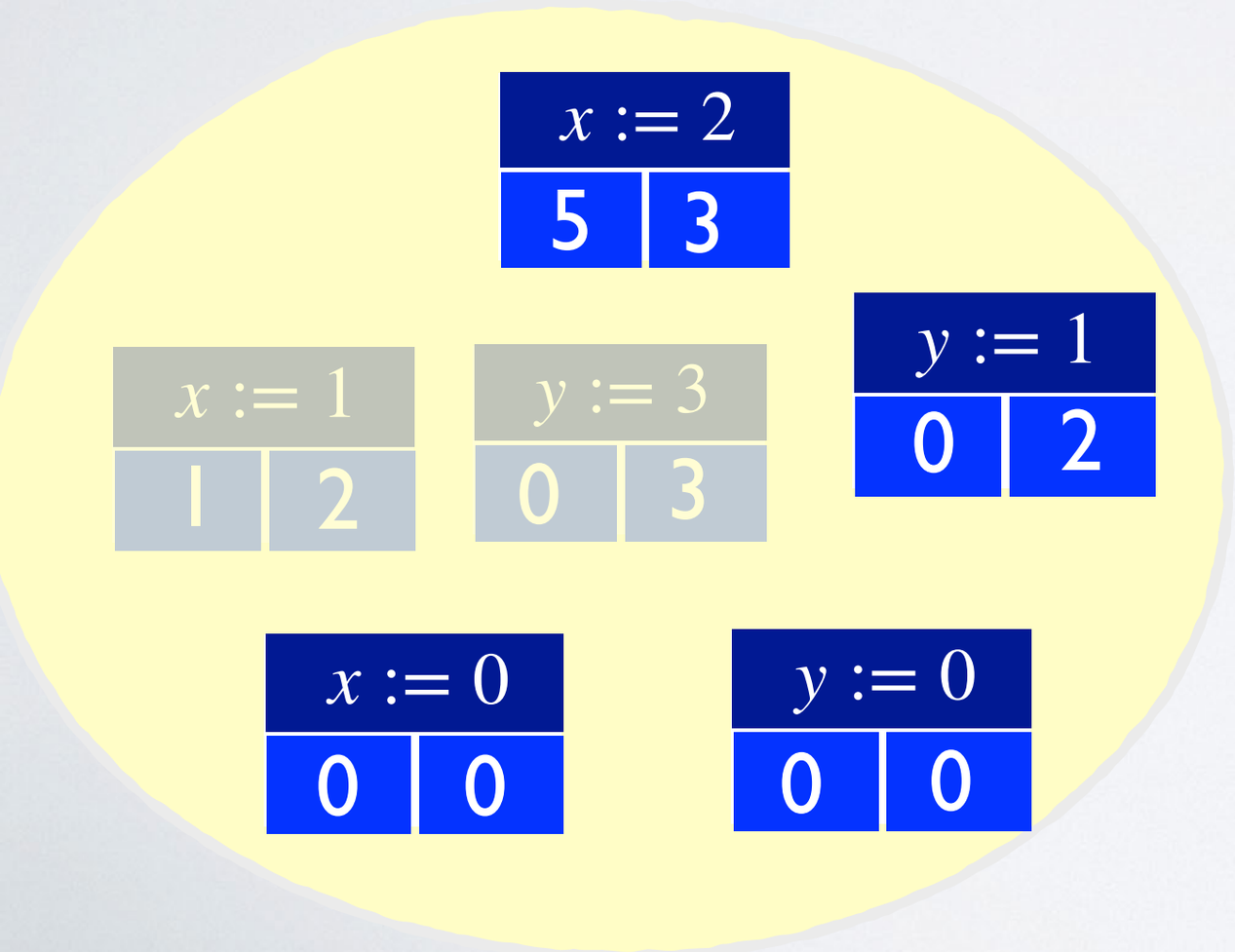
Bounding the number of **essential** views in the memory

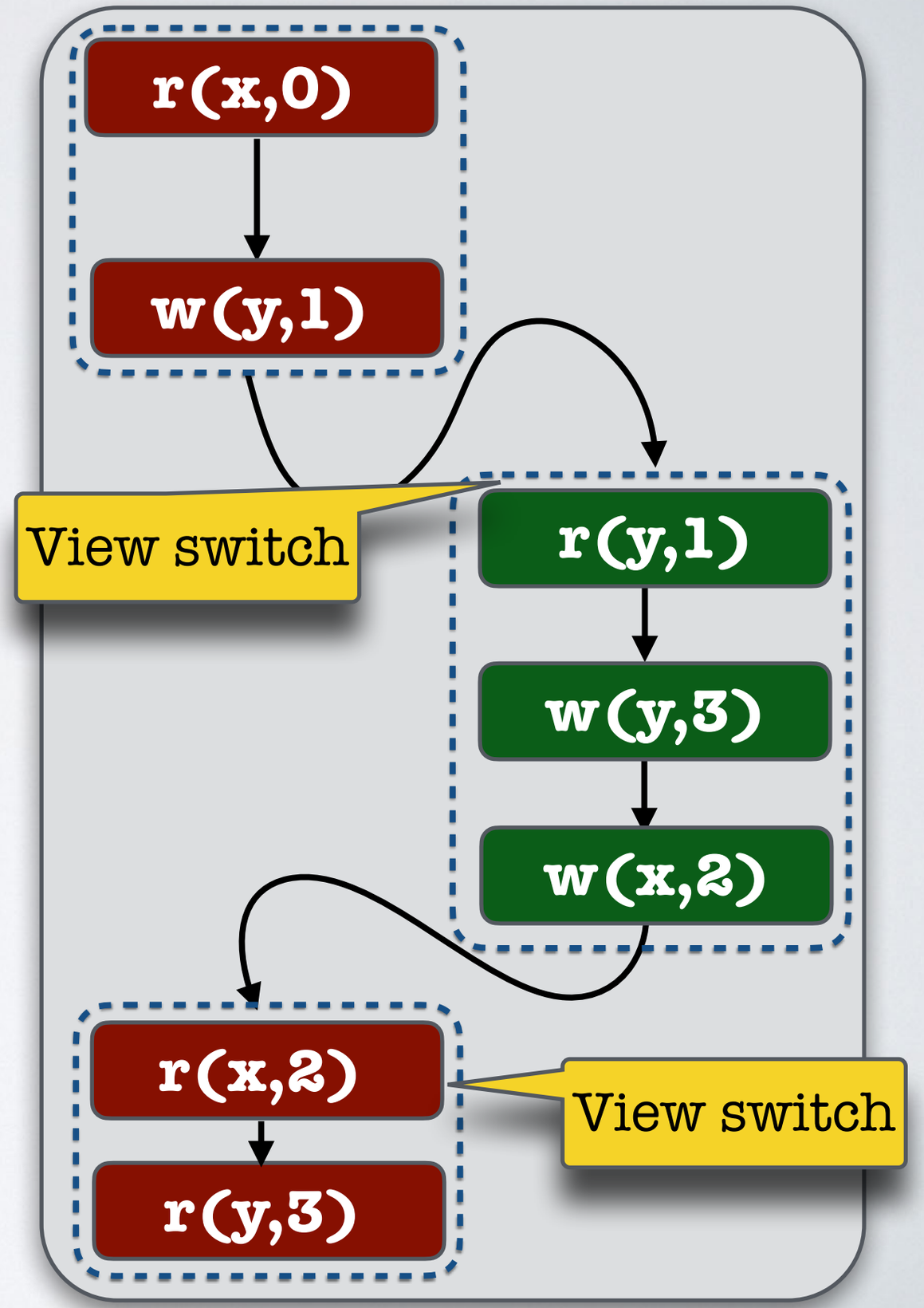




A view-switch happens when a process reads a value written by another process, and changes its view

Bounding the number of **essential** views in the memory





A view-switch happens when a process reads a value written by another process, and changes its view

Bound
essenti

The undecidability proof needed unbounded number of view switches

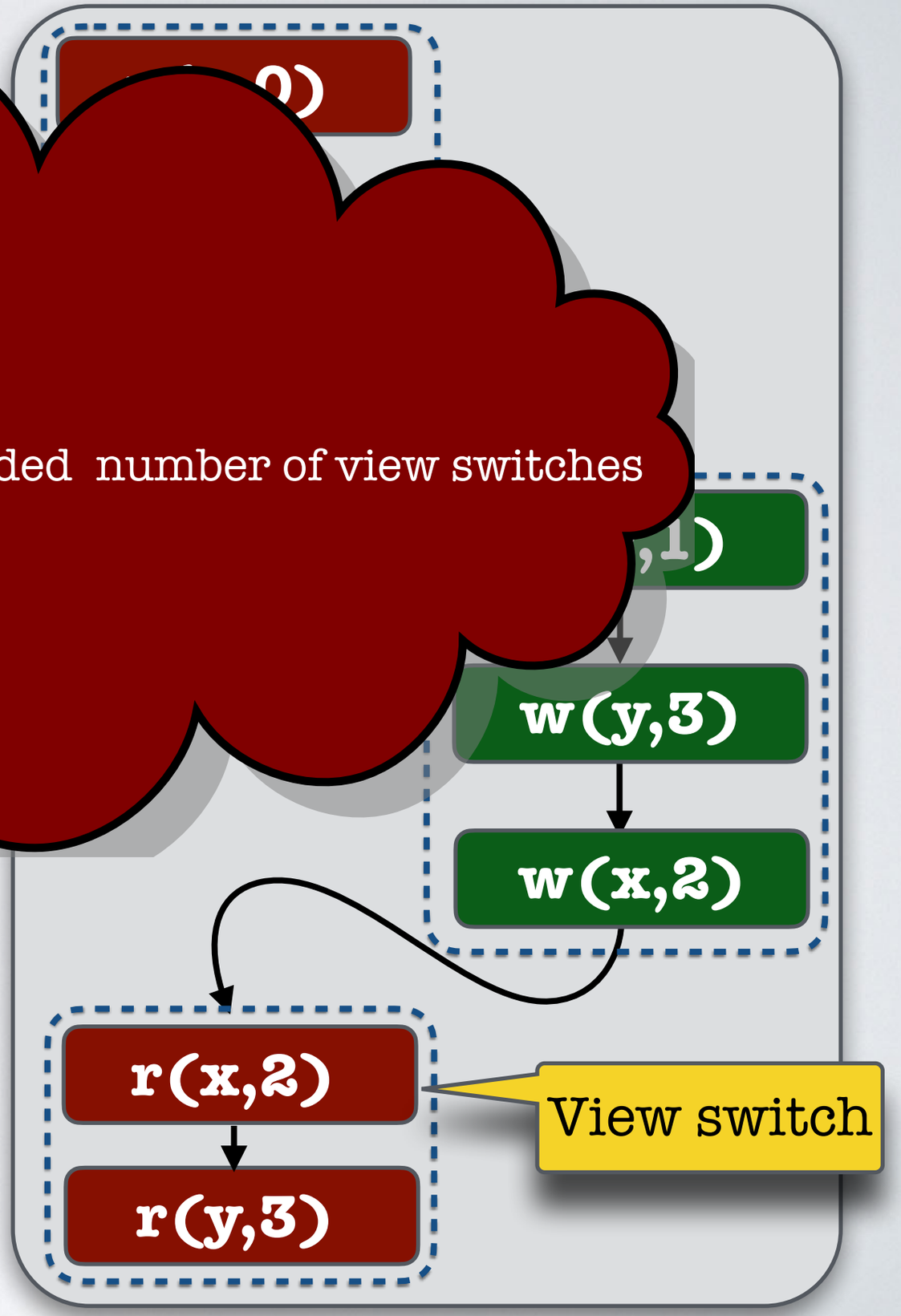
$x := 1$	
1	2

$y := 3$	
0	3

0	2

$x := 0$	
0	0

$y := 0$	
0	0



K-bounded Reachability Problem

K-bounded Reachability Problem

Definition

Reachability problem restricted to K-bounded runs

K-bounded Reachability Problem

Definition

Reachability problem restricted to K-bounded runs

Theorem

The K-bounded reachability for RA is reducible to K+n bounded context reachability under SC

Code-to-code translation

K-bounded Reachability Problem

Definition

Reachability problem restricted to K-bounded runs

Code-to-code translation

Theorem

The K-bounded reachability for RA is reducible to K+n bounded context reachability under SC

Corollary

The K-bounded reachability for RA is decidable for finite-state programs

View bounding under RA to Context bounding under SC

View bounding under RA to Context bounding under SC

- View bound K implies K essential messages

View bounding under RA to Context bounding under SC

- View bound K implies K essential messages
- K time stamps, one per essential message

View bounding under RA to Context bounding under SC

- View bound K implies K essential messages
- K time stamps, one per essential message
- Another K time stamps for comparison corresponding to a view change

View bounding under RA to Context bounding under SC

- View bound K implies K essential messages
- K time stamps, one per essential message
- Another K time stamps for comparison corresponding to a view change
- 2K time stamps

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

Globally, maintain a set of **Messages**, each of type (var, View)

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

Globally, maintain a set of **Messages**, each of type (var, View)

- On each write x:=v in process p, update pval(x) in pview_x

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

Globally, maintain a set of **Messages**, each of type (var, View)

- On each write x:=v in process p, update pval(x) in pview_x
- Guess if this will be used in a time stamp comparison. If yes, pUseful(x)=true, and update ptime(x)

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

Globally, maintain a set of **Messages**, each of type (var, View)

- On each write x:=v in process p, update pval(x) in pview_x
- Guess if this will be used in a time stamp comparison. If yes, pUseful(x)=true, and update ptime(x)
- Guess if this will result in an essential message. If yes, add to **Messages** if pUseful(x)=true for all pview_x in pView

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

Globally, maintain a set of **Messages**, each of type (var, View)

- On each write x:=v in process p, update pval(x) in pview_x
- Guess if this will be used in a time stamp comparison. If yes, pUseful(x)=true, and update ptime(x)
- Guess if this will result in an essential message. If yes, add to **Messages** if pUseful(x)=true for all pview_x in pView
- Once all essential messages are generated, **Messages** has all necessary data, simply run all processes to completion (**K+n context** switches).

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

Globally, maintain a set of messages, each of type (var, View)

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

Globally, maintain a set of messages, each of type (var, View)

- On each read r:=x, guess if this read is view altering. If yes, check if pUseful(x) is true in pview_x, and pUseful(var) is true in pView for all var

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

Globally, maintain a set of messages, each of type (var, View)

- On each read r:=x, guess if this read is view altering. If yes, check if pUseful(x) is true in pview_x, and pUseful(var) is true in pView for all var

- Pick up a suitable m=(x, View) from **Messages**

Per process p, maintain pView=(pview_x,....pview_z),
pview_x=(ptime(x), pval(x), pUseful(x))

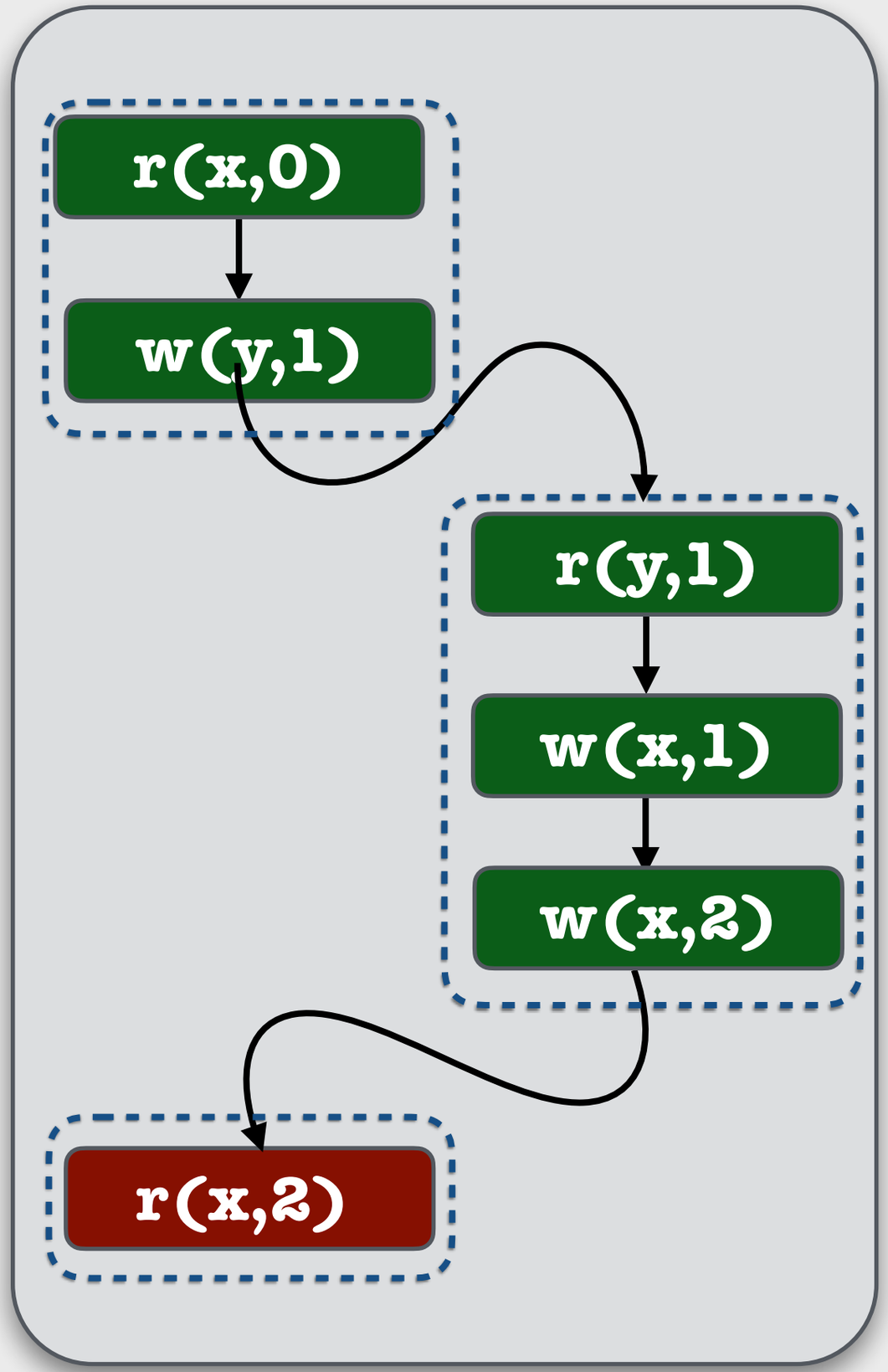
Globally, maintain a set of messages, each of type (var, View)

- On each read r:=x, guess if this read is view altering. If yes, check if pUseful(x) is true in pview_x, and pUseful(var) is true in pView for all var

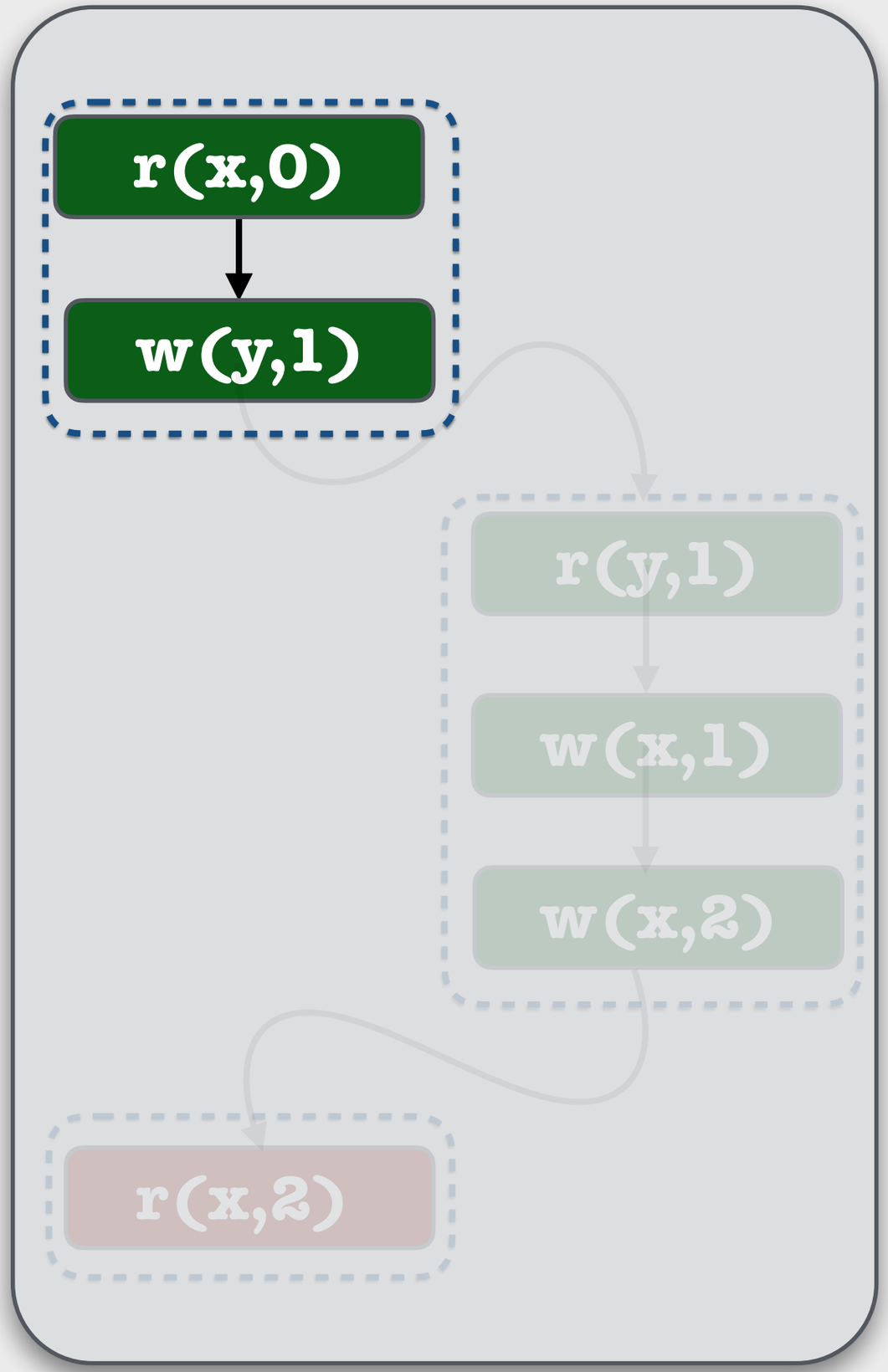
- Pick up a suitable m=(x, View) from **Messages**

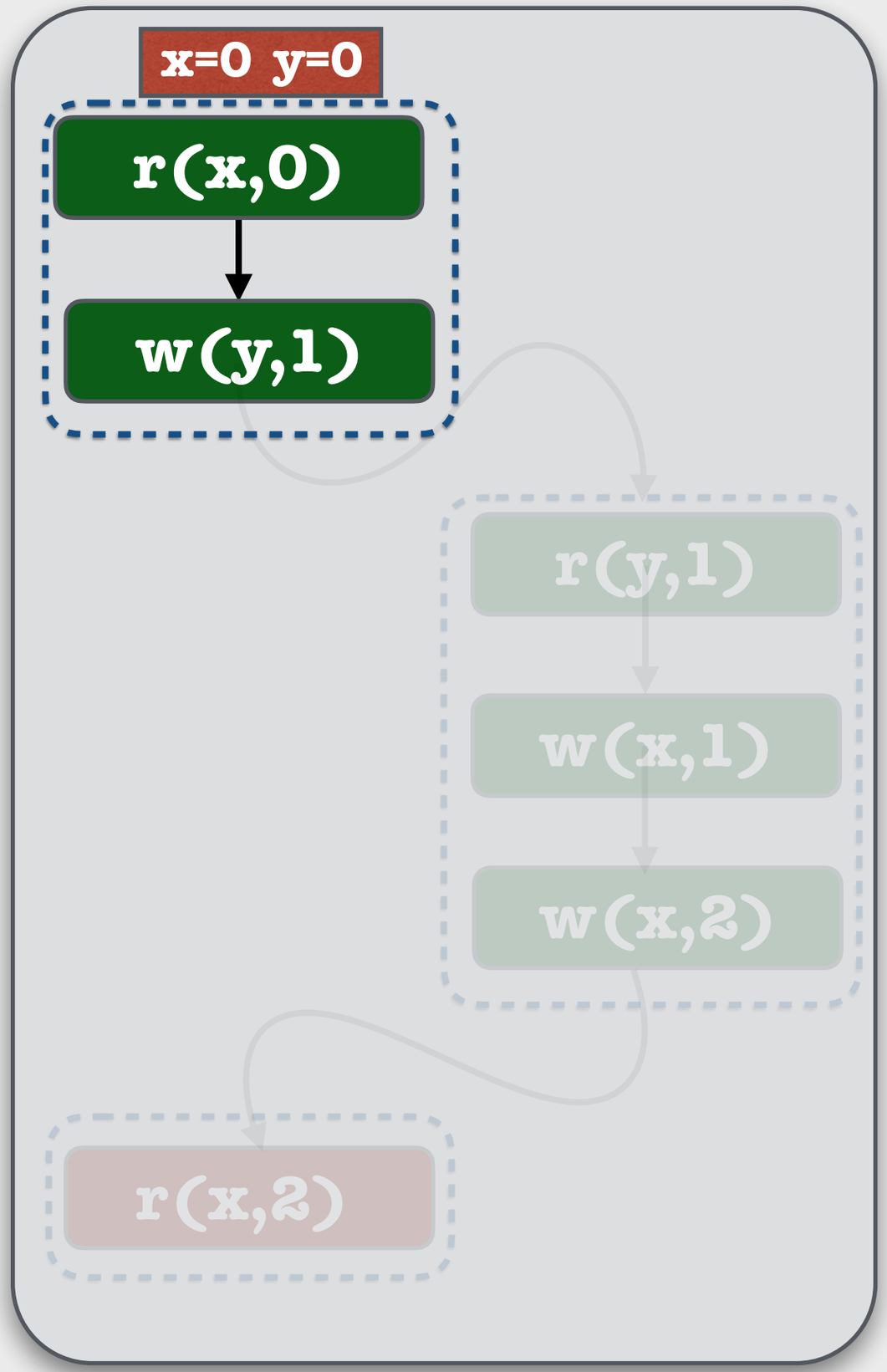
- Update ptime(var), pval(var) in pView if ptime(var) is at most mtime(var)

Guess essential messages, run processes generating them in order



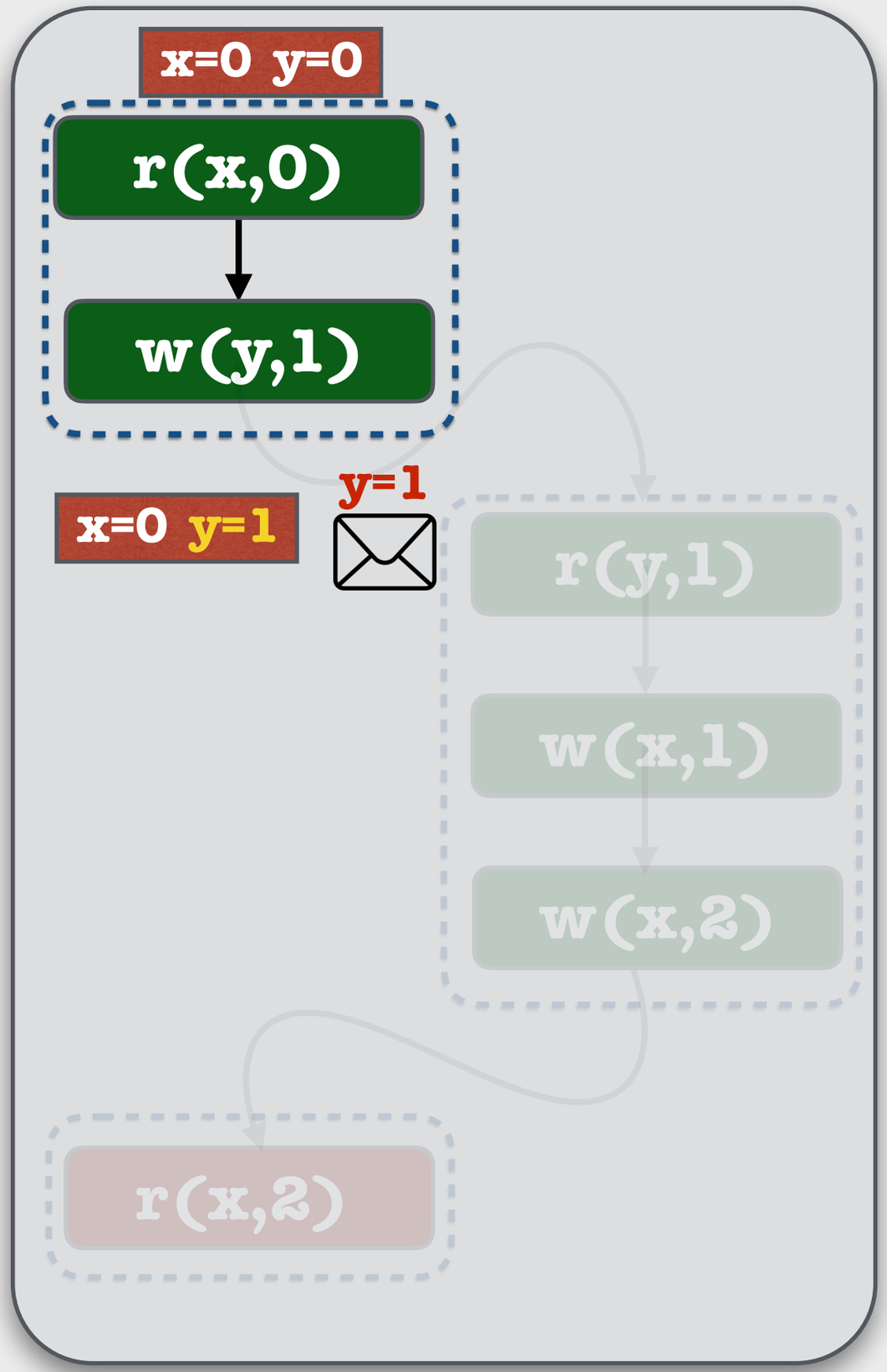
Guess essential messages, run processes generating them in order



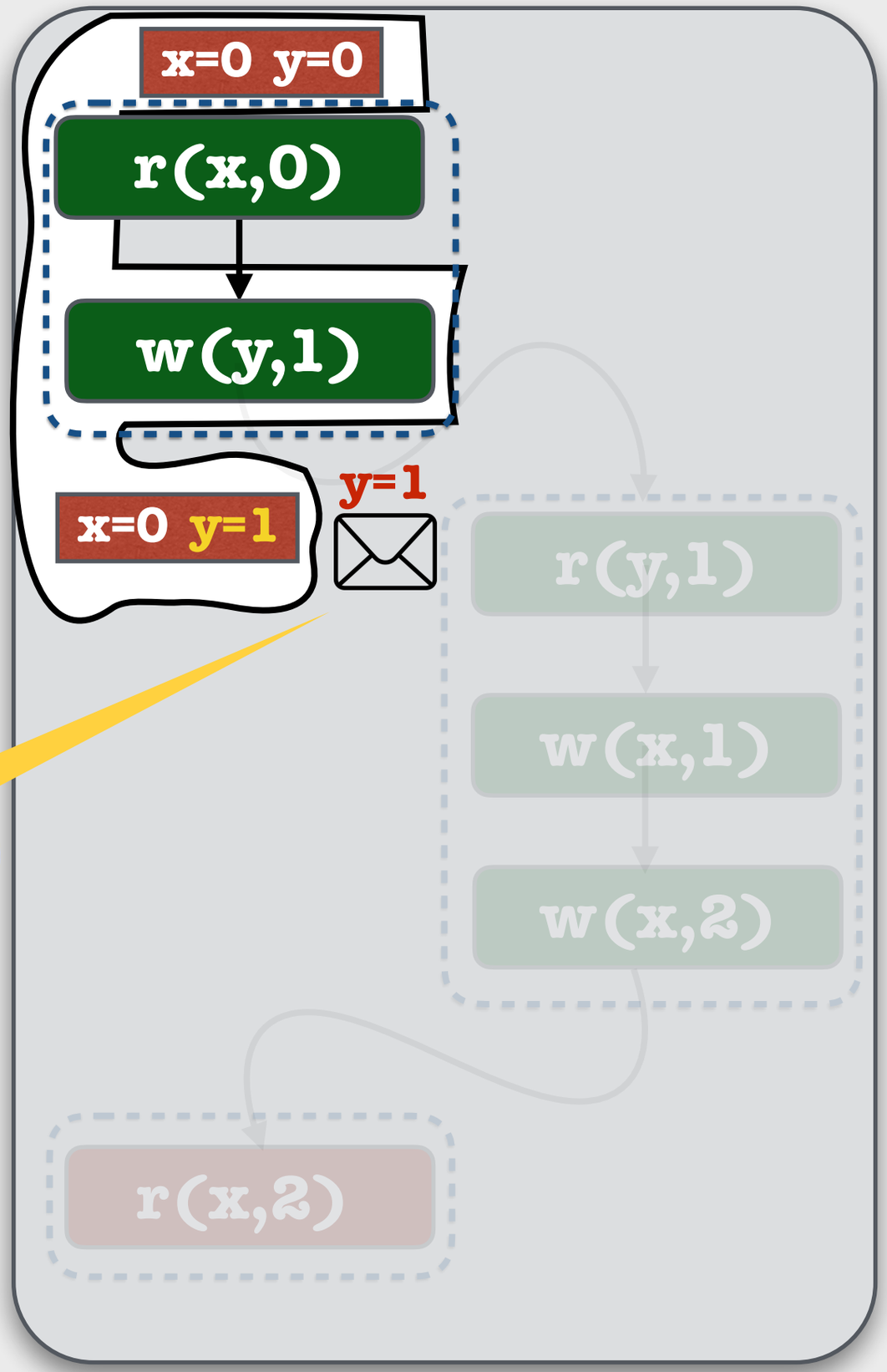


Guess essential messages, run processes generating them in order

Guess essential messages, run processes generating them in order

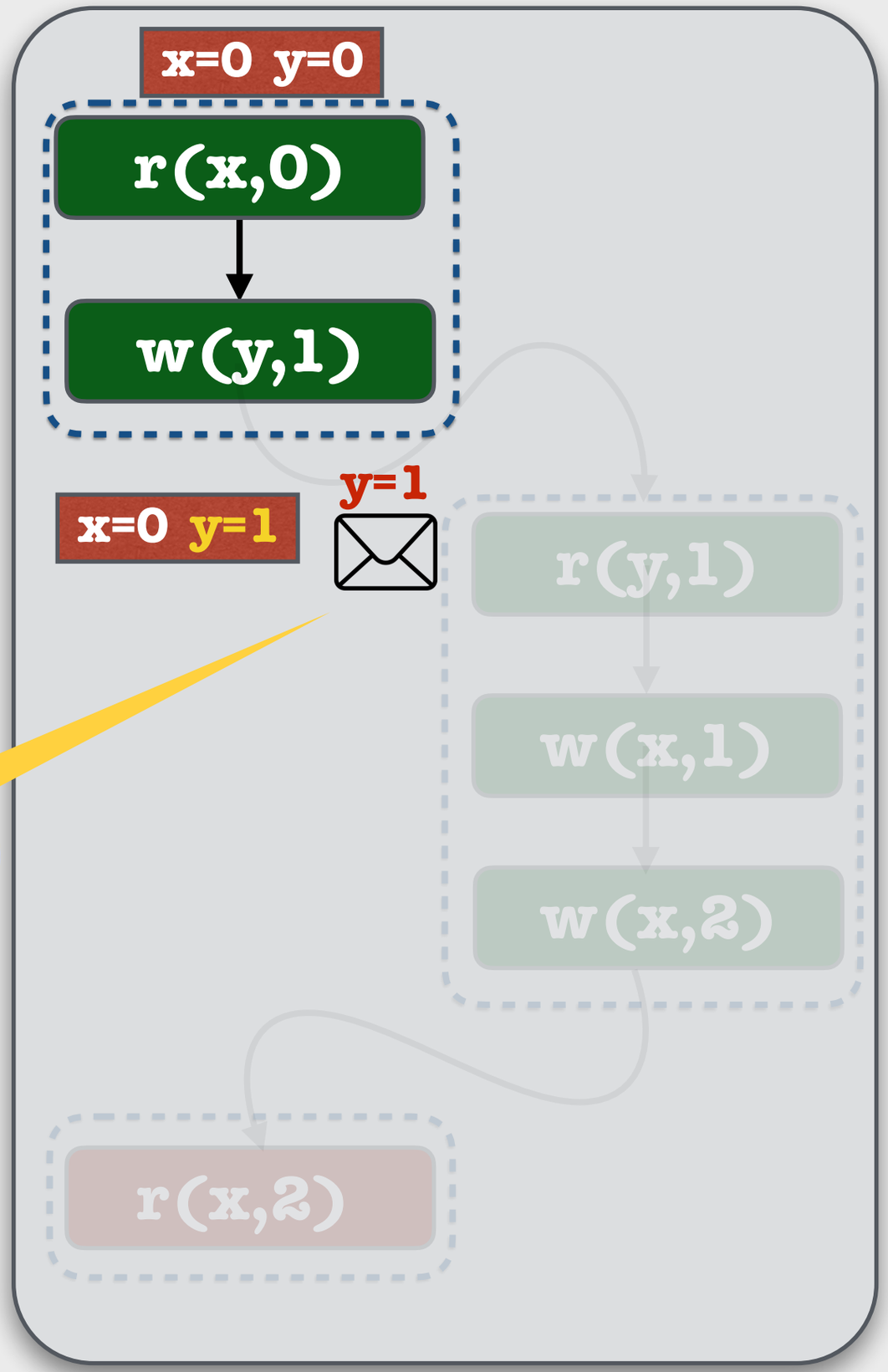


Guess essential messages, run processes generating them in order

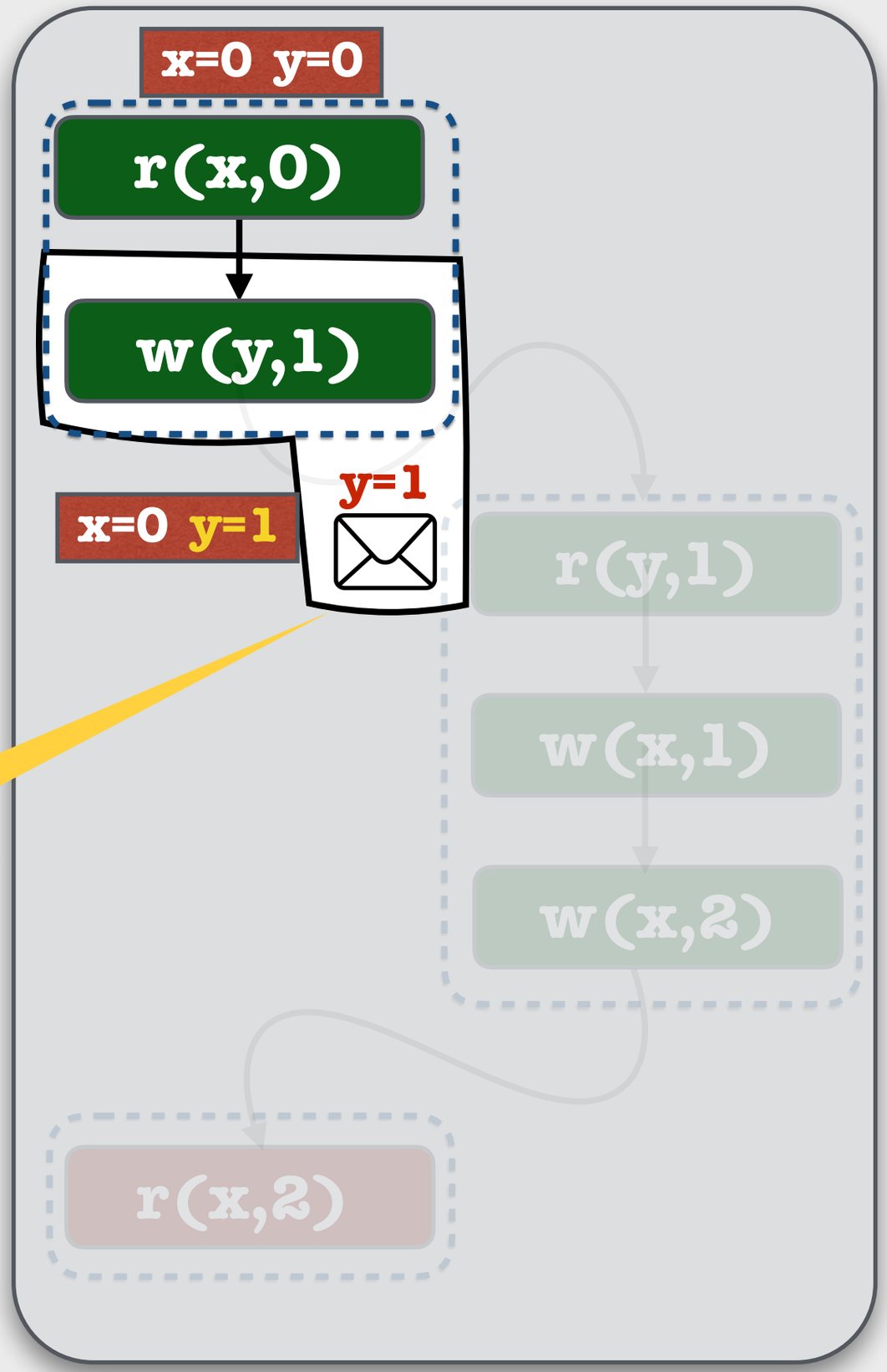


First essential message

Guess essential messages, run processes generating them in order



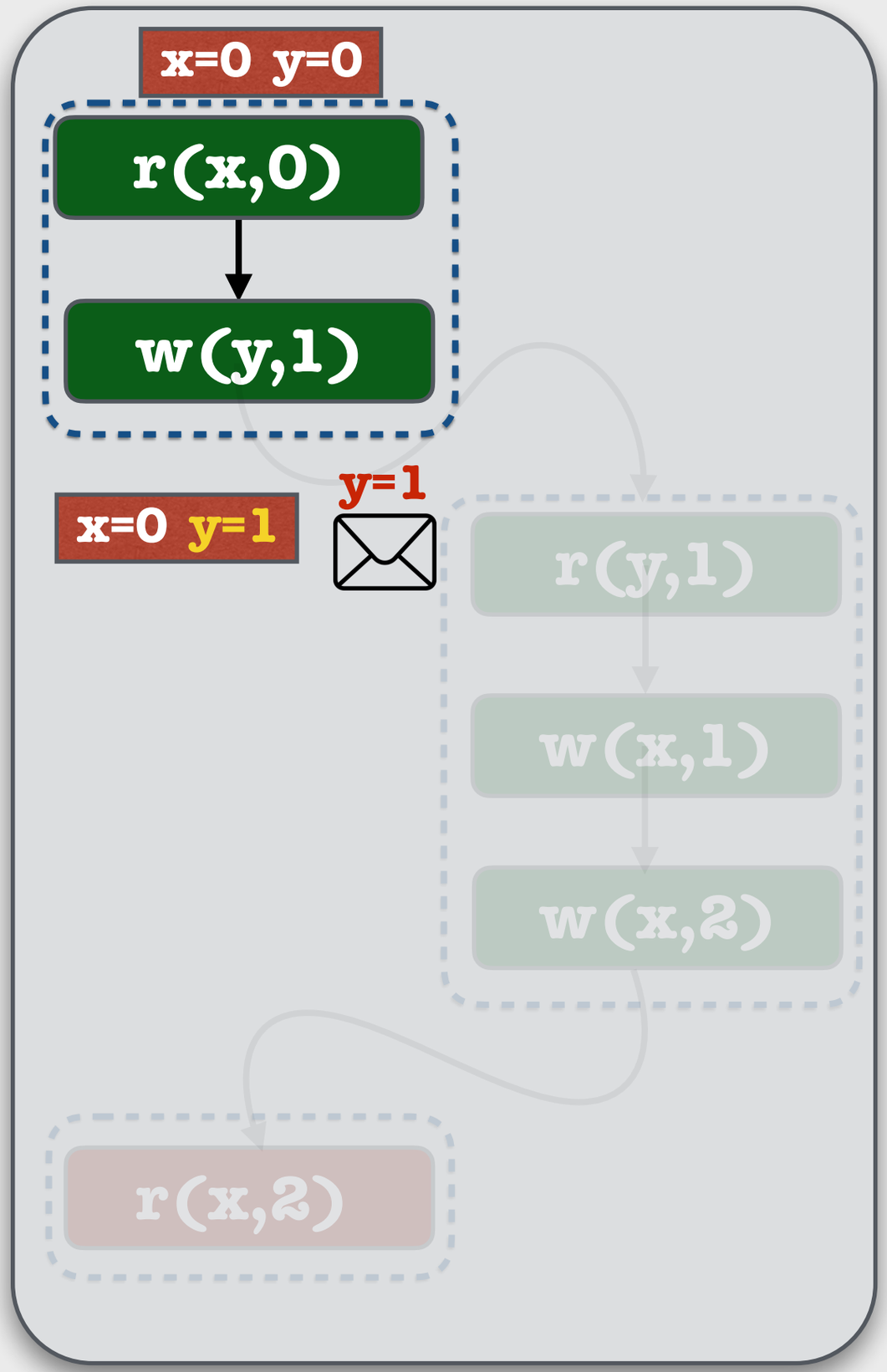
First essential message



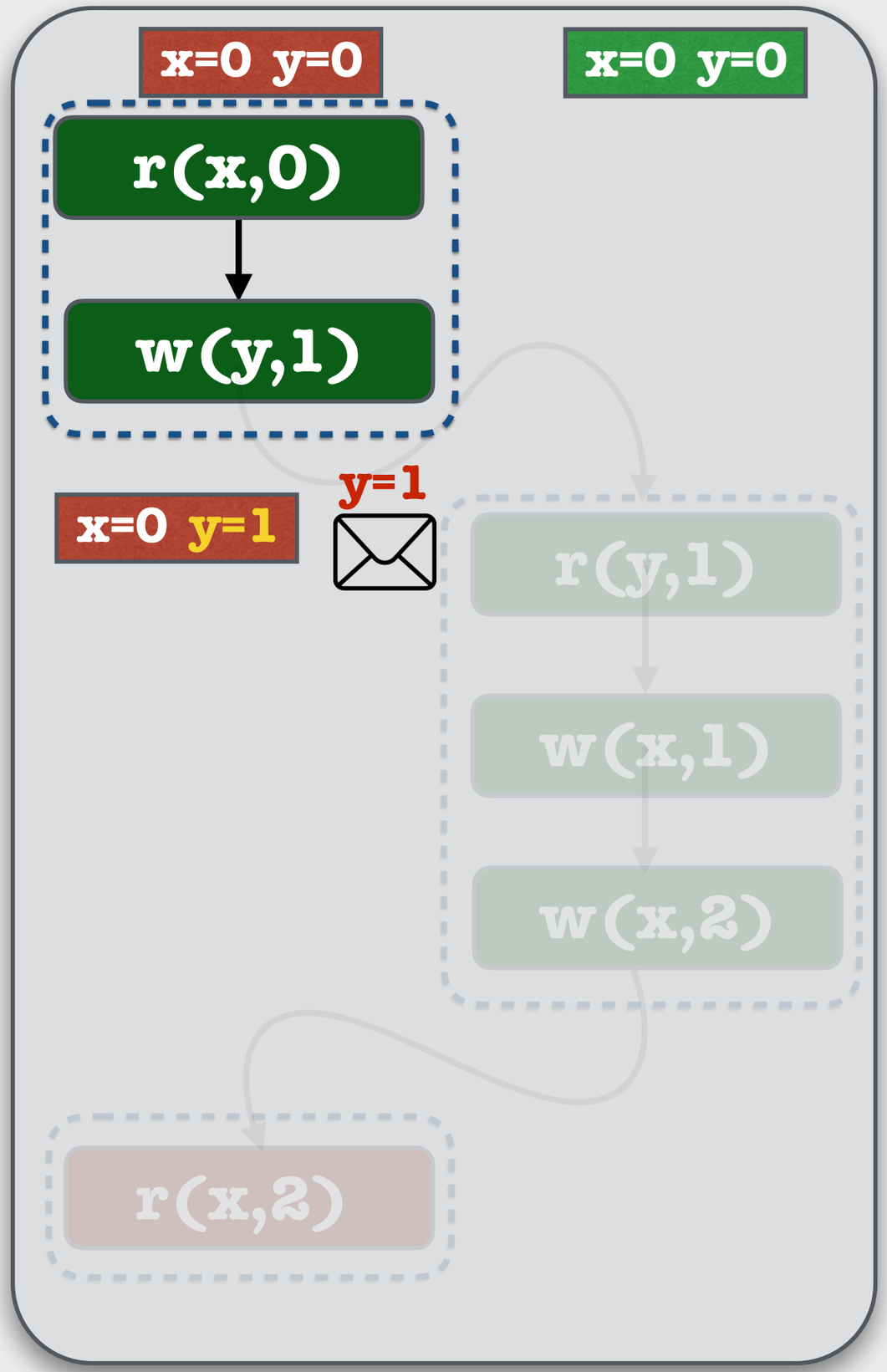
First essential message

Guess essential messages, run processes generating them in order

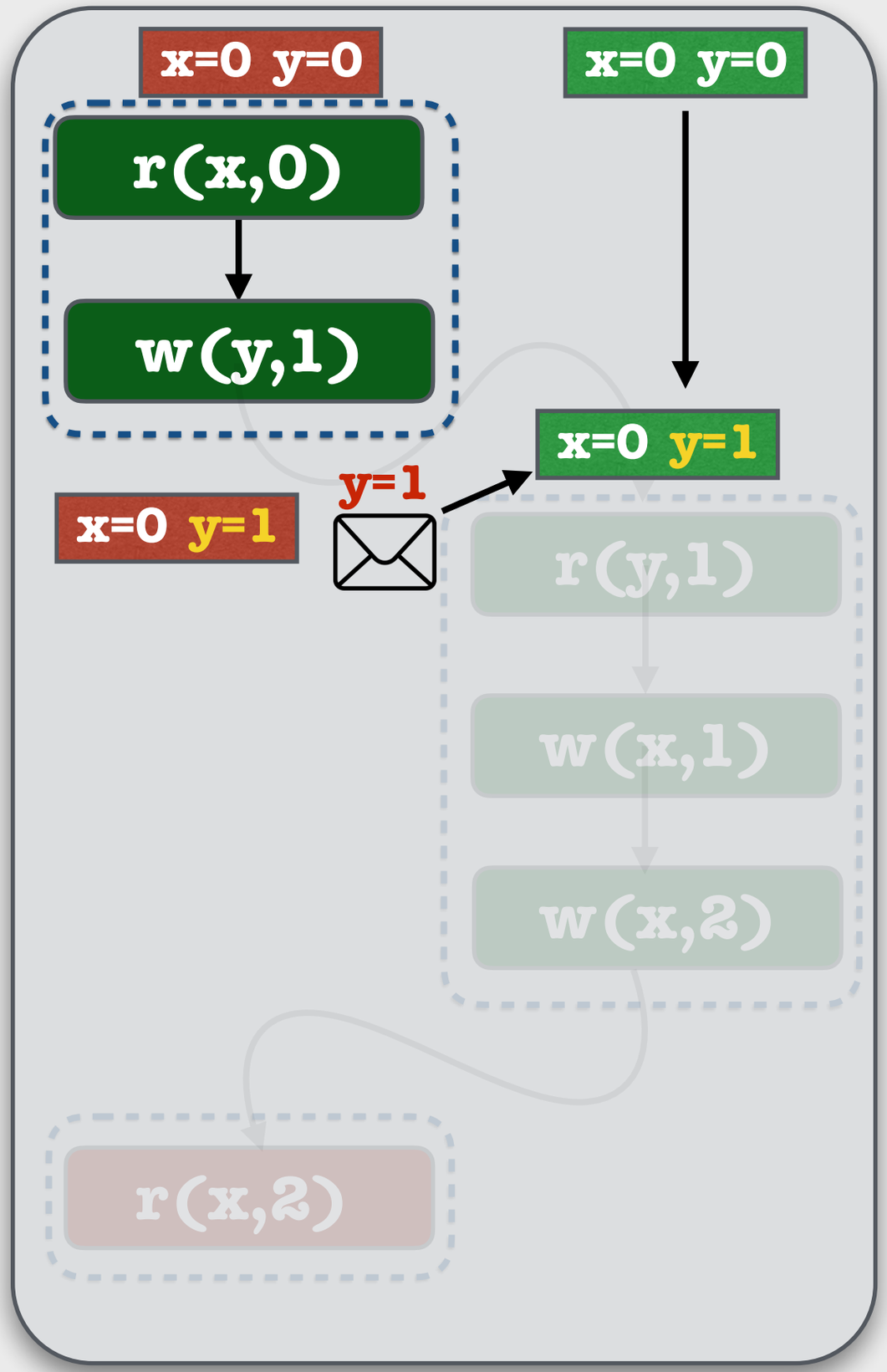
Guess essential messages, run processes generating them in order



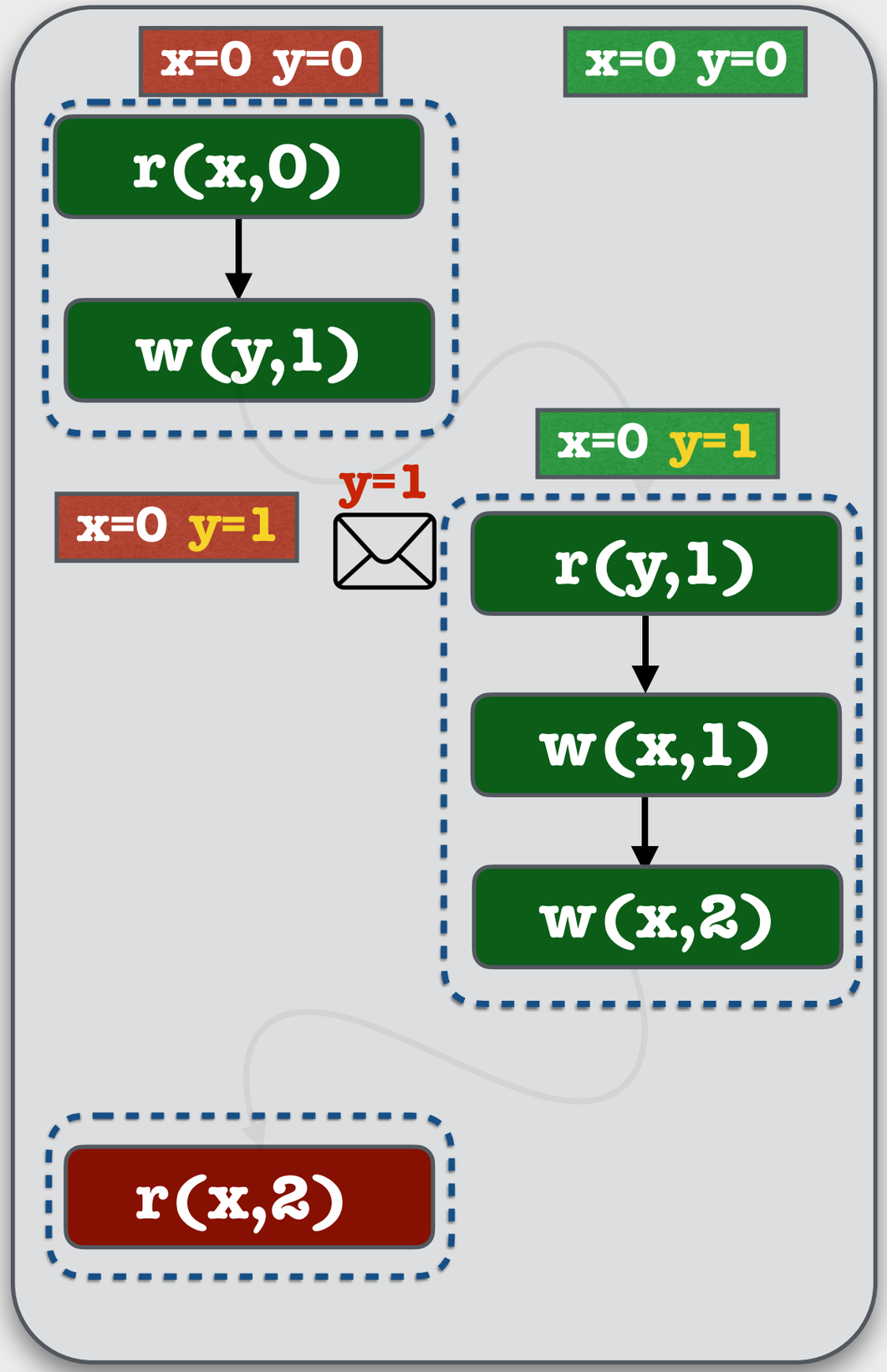
Guess essential messages, run processes generating them in order



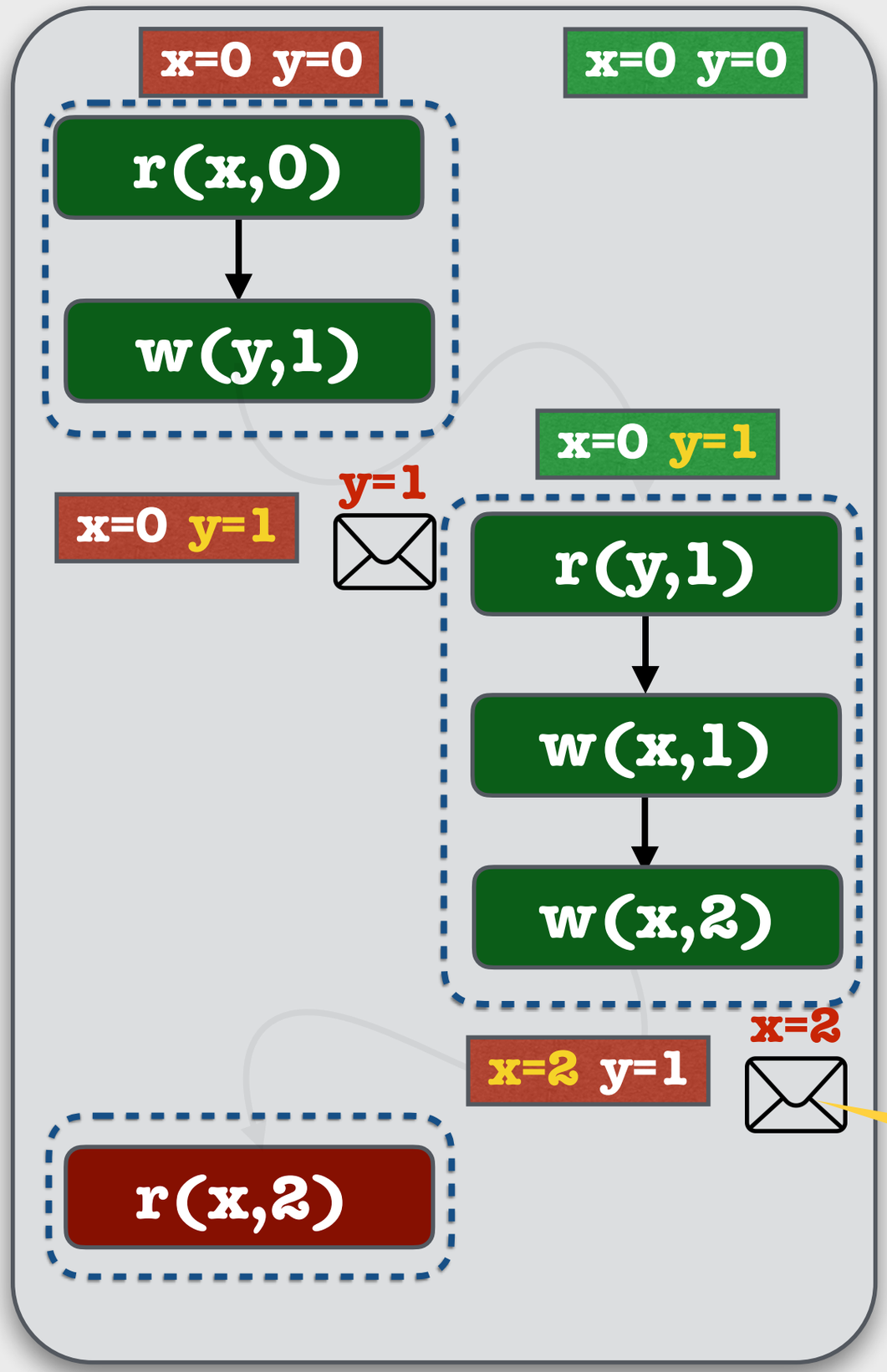
Guess essential messages, run processes generating them in order



Guess essential messages, run processes generating them in order

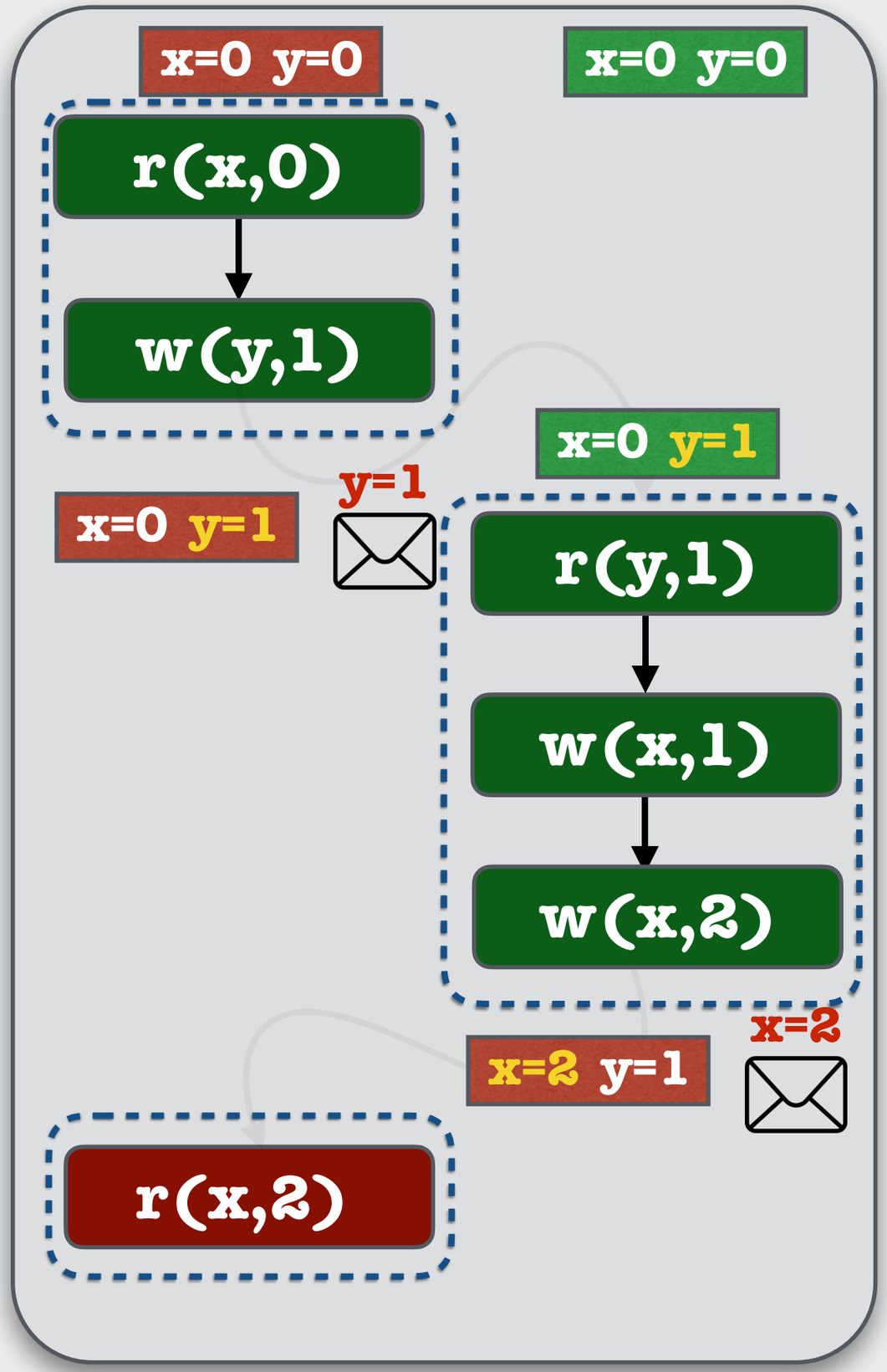


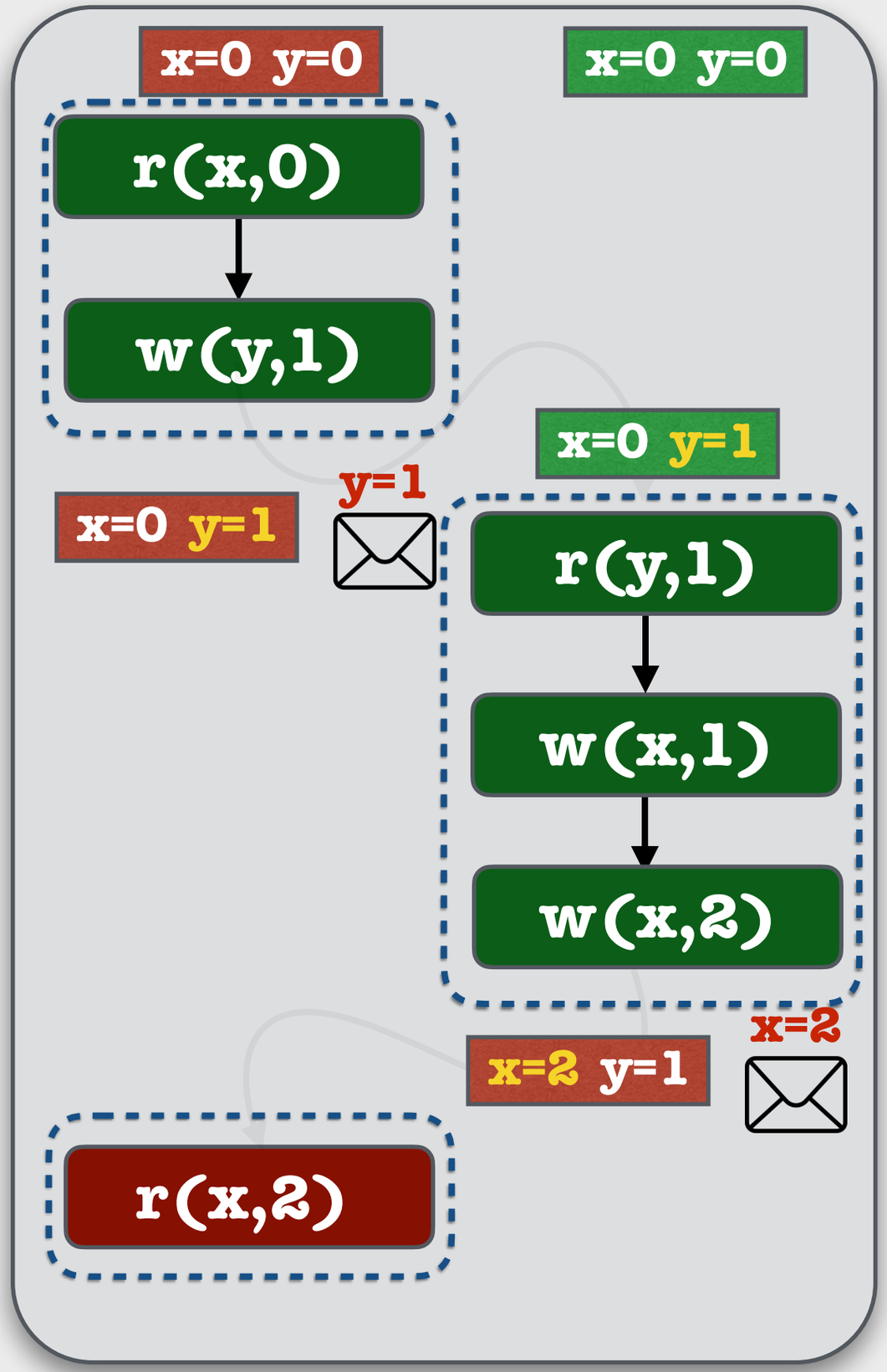
Guess essential messages, run processes generating them in order



second essential message

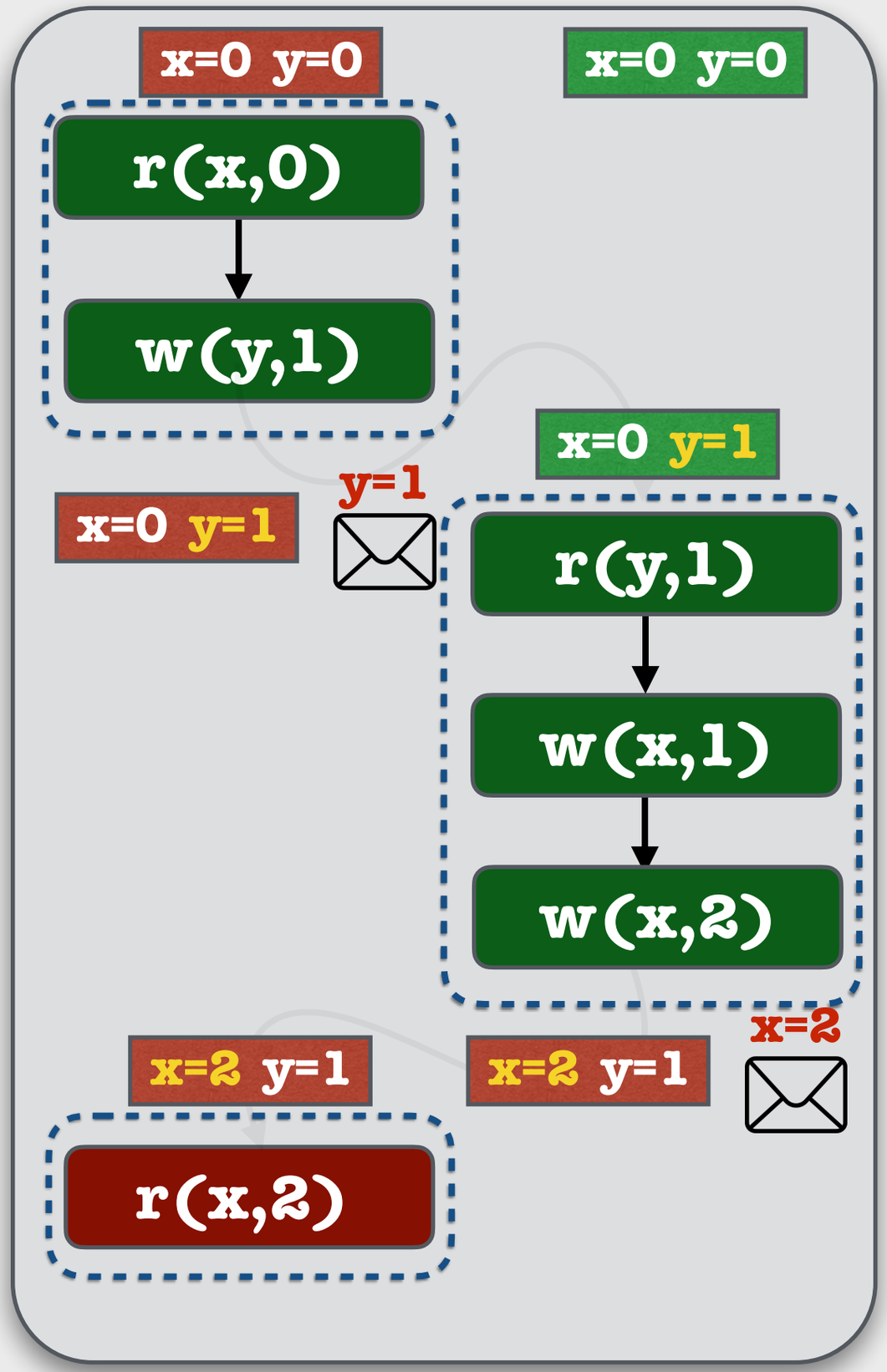
Guess essential messages, run processes generating them in order





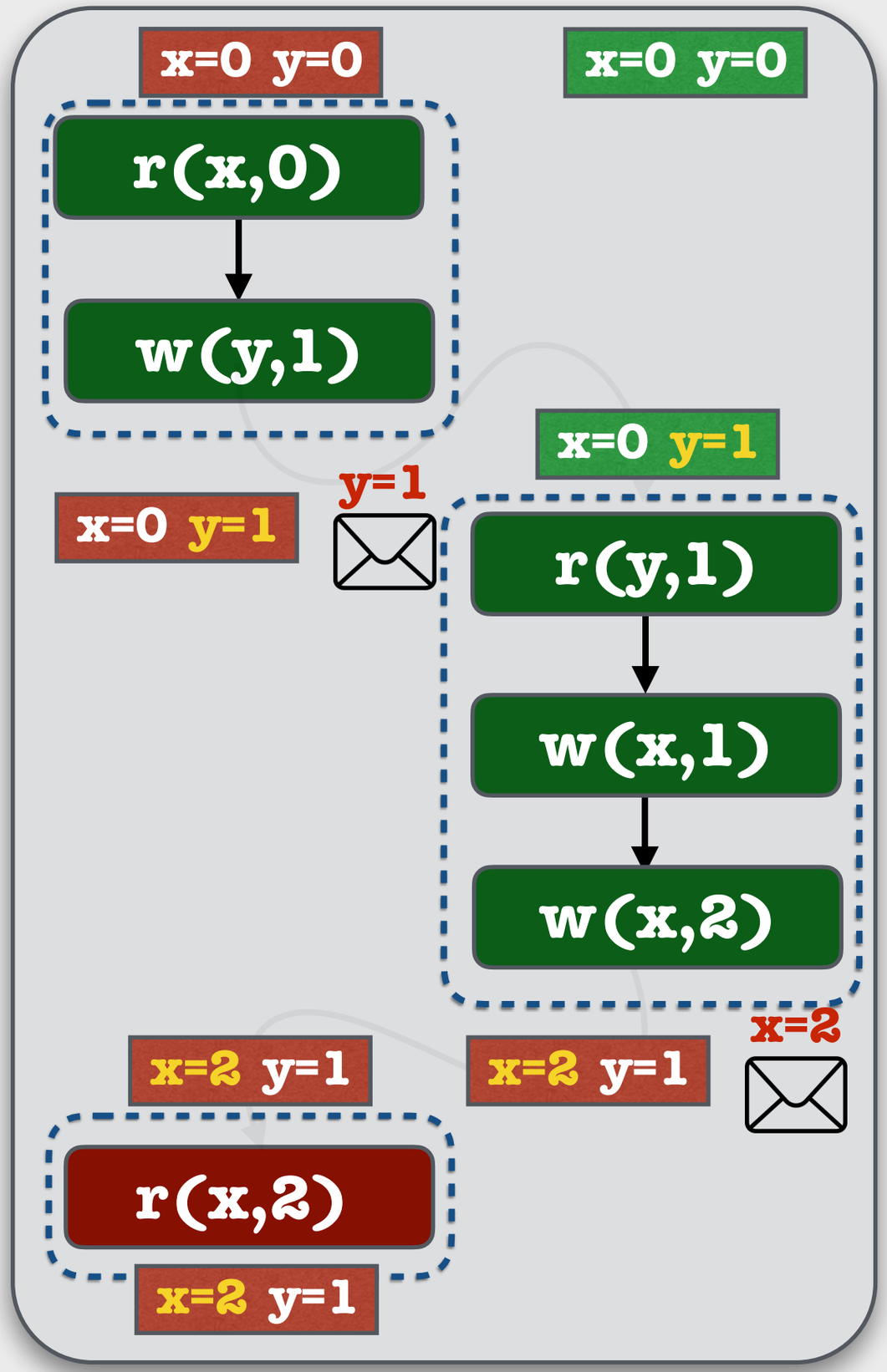
Guess essential messages, run processes generating them in order

Once all essential messages are generated, run all processes to completion



Guess essential messages, run processes generating them in order

Once all essential messages are generated, run all processes to completion



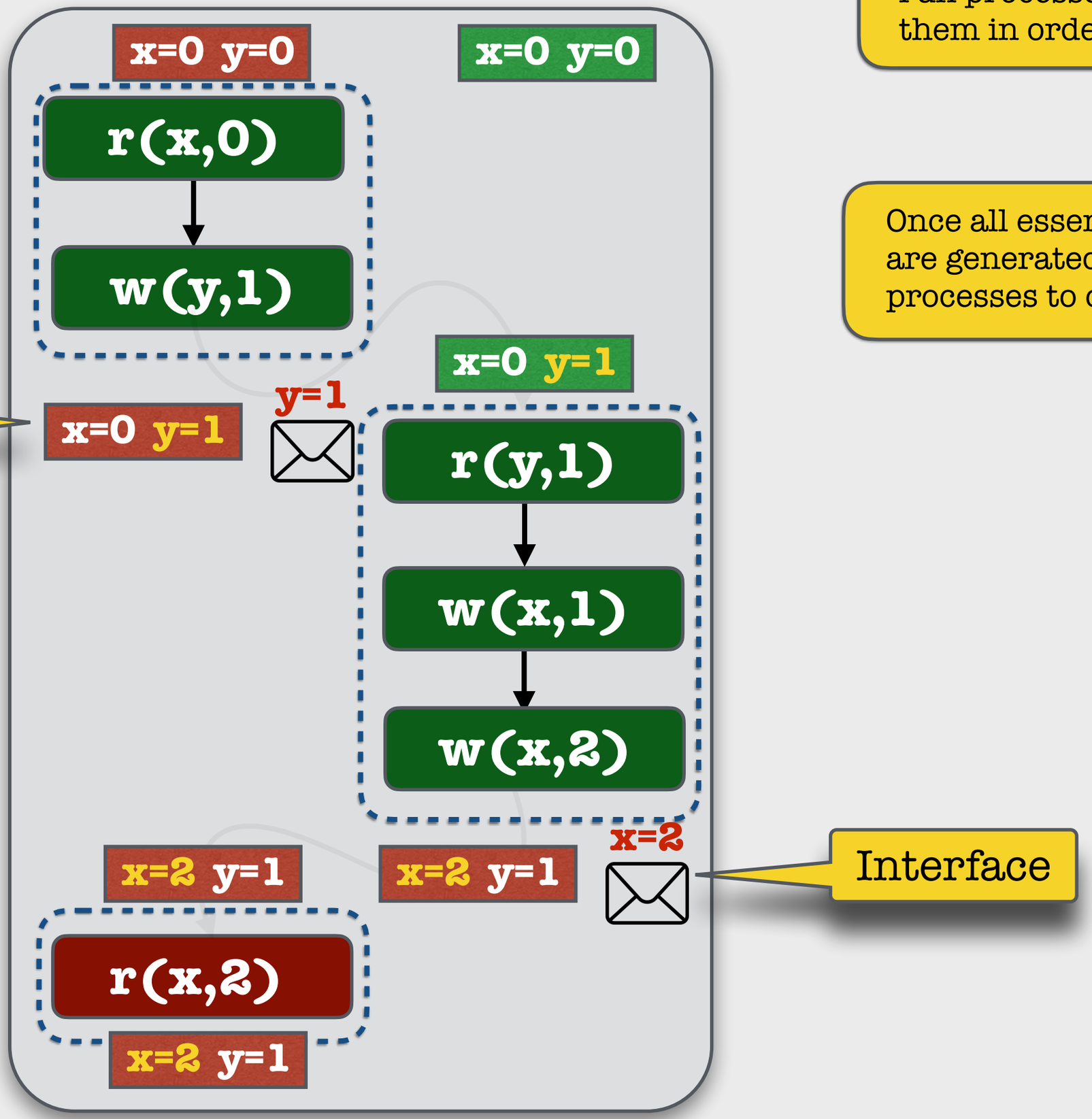
Guess essential messages, run processes generating them in order

Once all essential messages are generated, run all processes to completion

Guess essential messages, run processes generating them in order

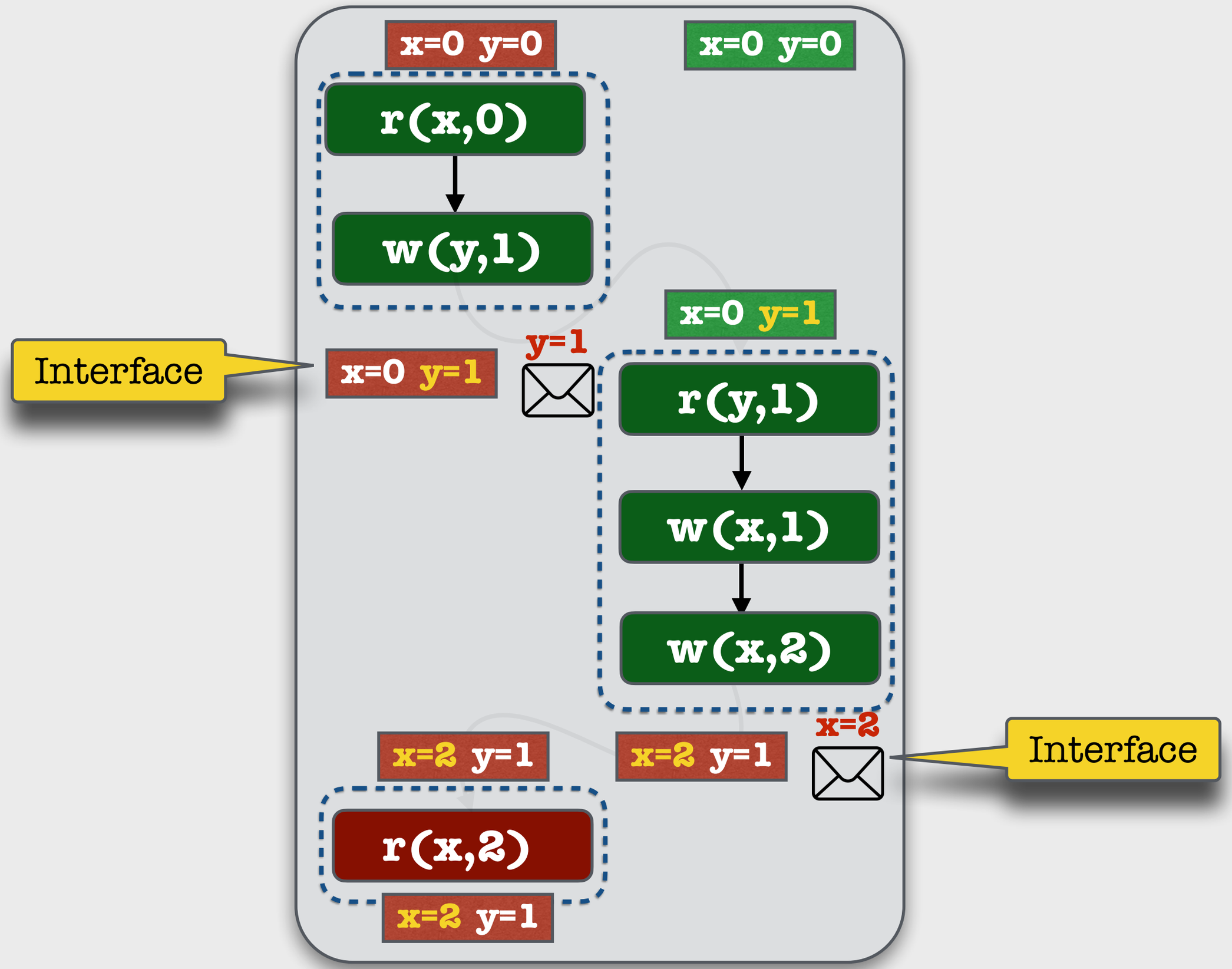
Once all essential messages are generated, run all processes to completion

Interface



Locality

Validation



Locality

Validation

x=0 y=0

r(x,0)

w(y,1)

x=0 y=1

y=1

Interface

r(y,1)

w(x,1)

w(x,2)

x=2

x=2 y=1

Interface

x=2 y=1

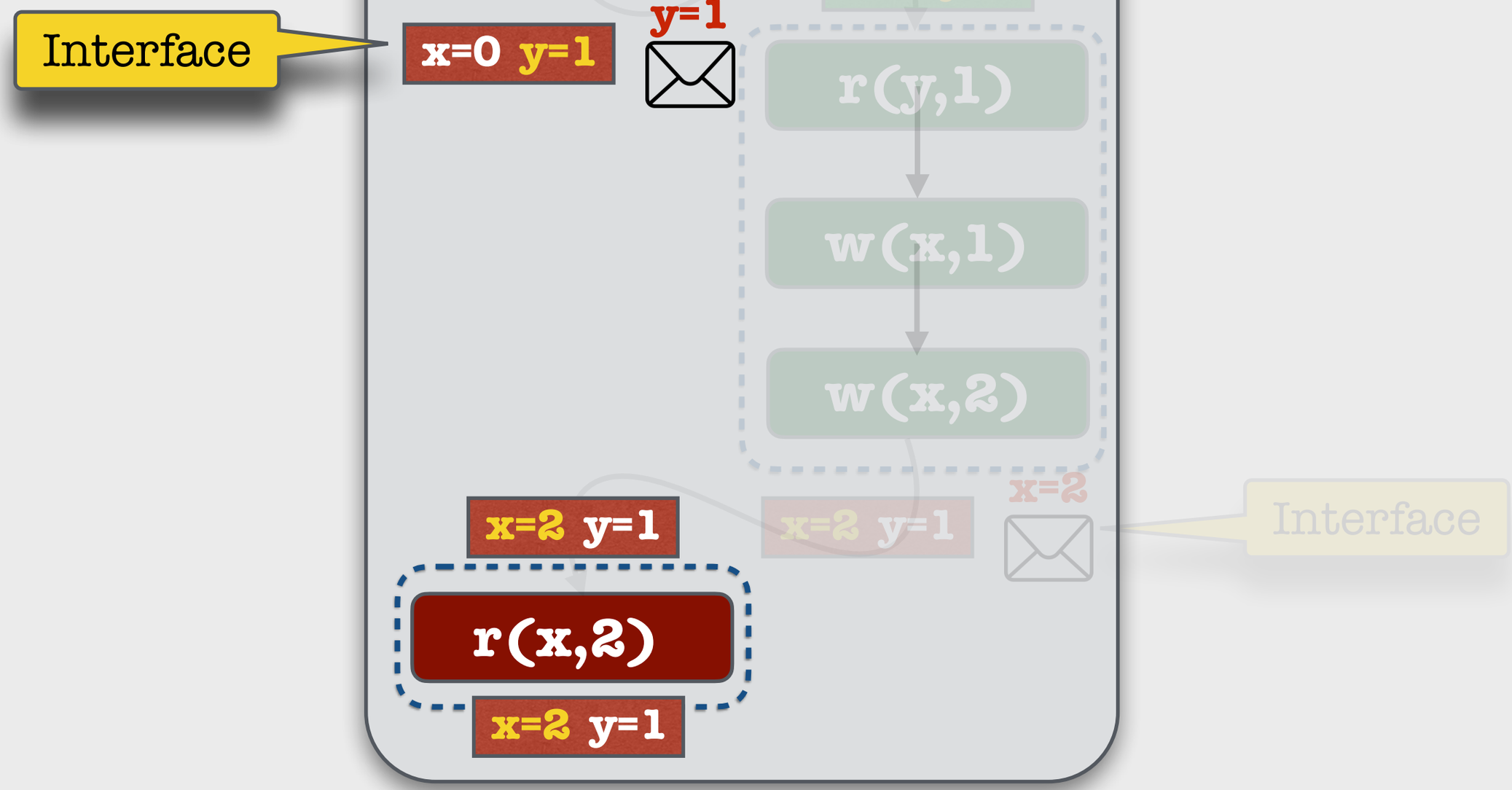
r(x,2)

x=2 y=1

Locality

Validation

Simulate **P1** under **SC**



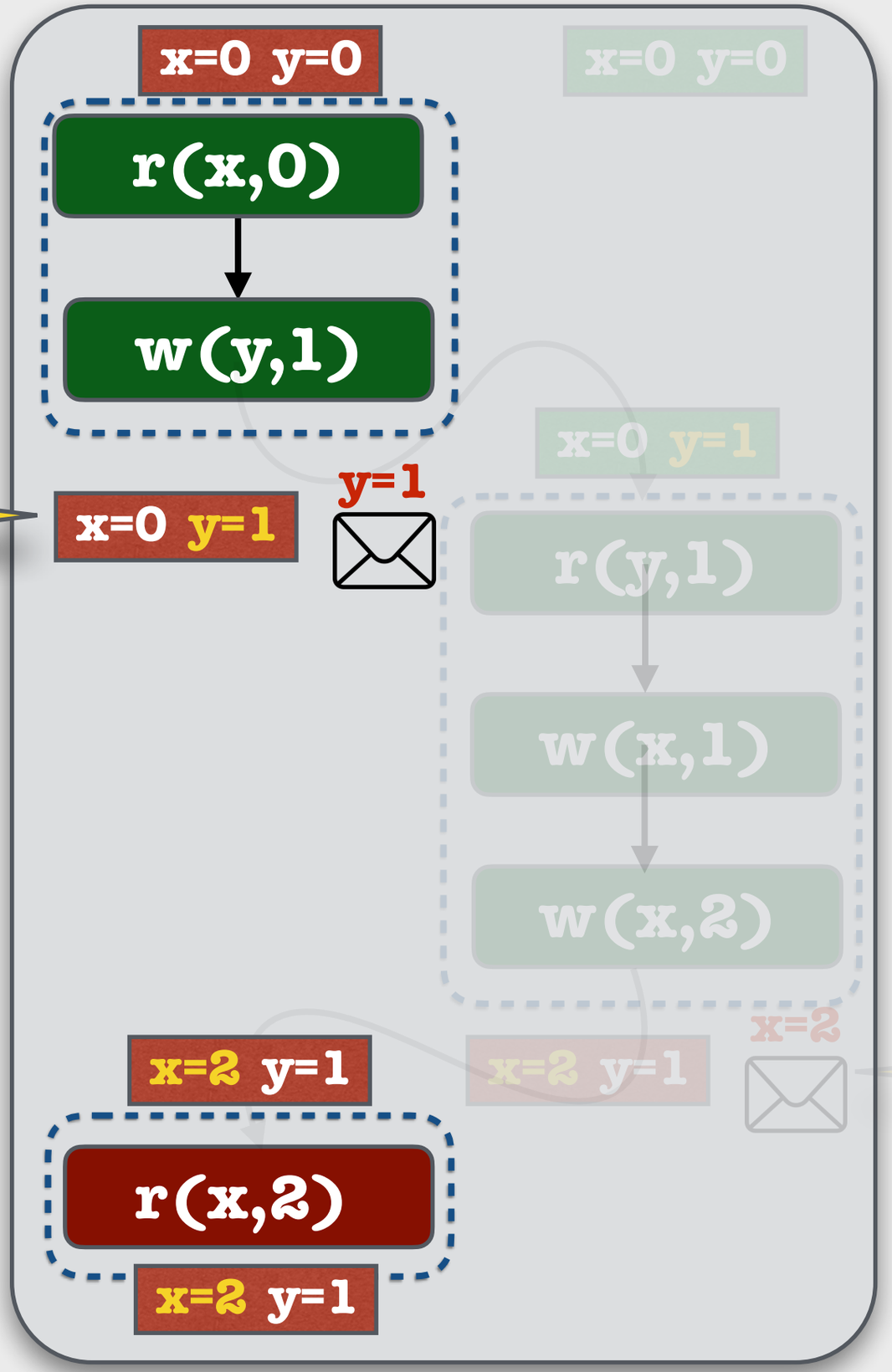
Locality

Validation

Simulate **P1** under **SC**

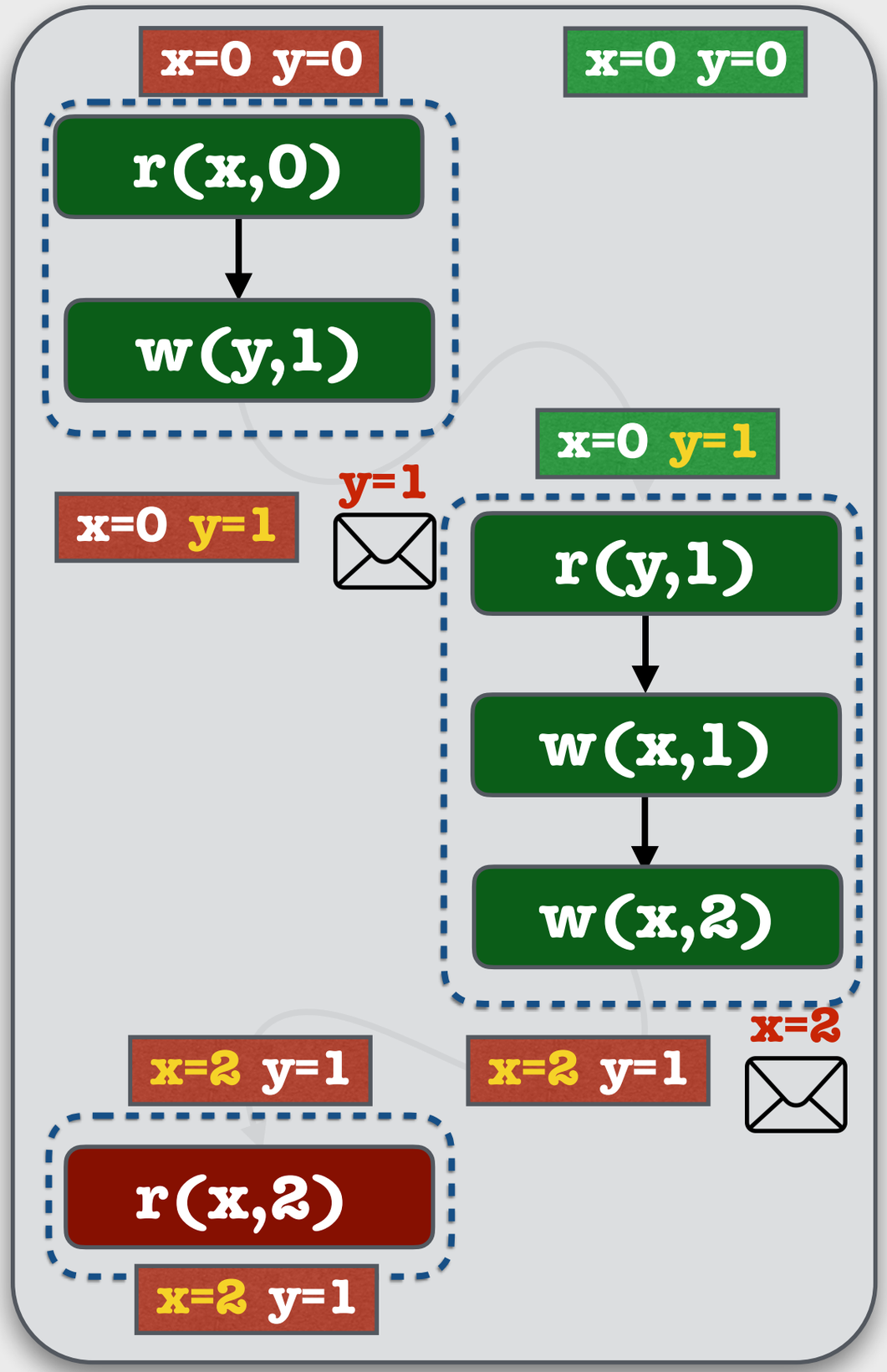
Local and global data structures
to simulate views and the memory pool

Interface



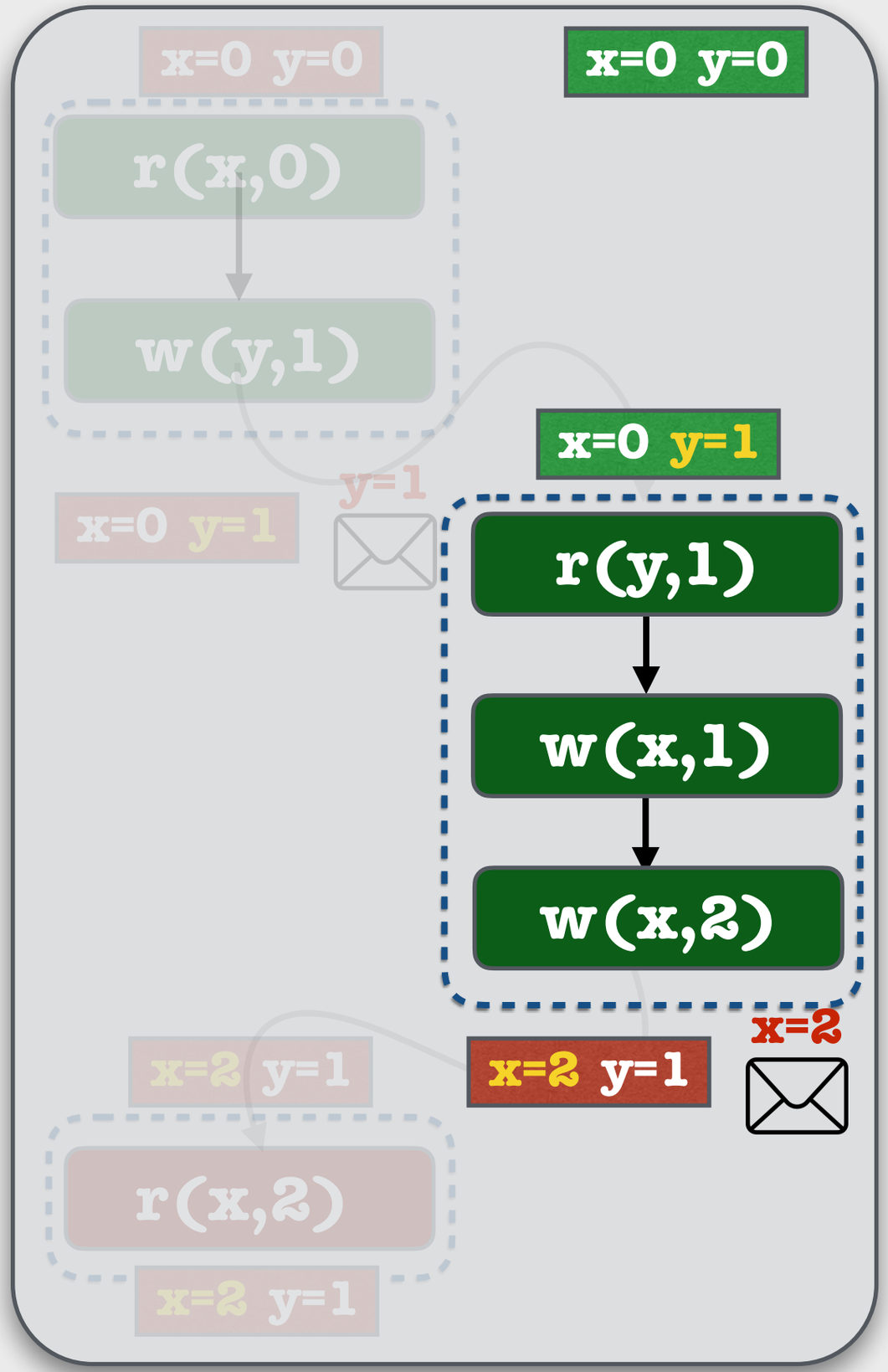
Locality

Validation



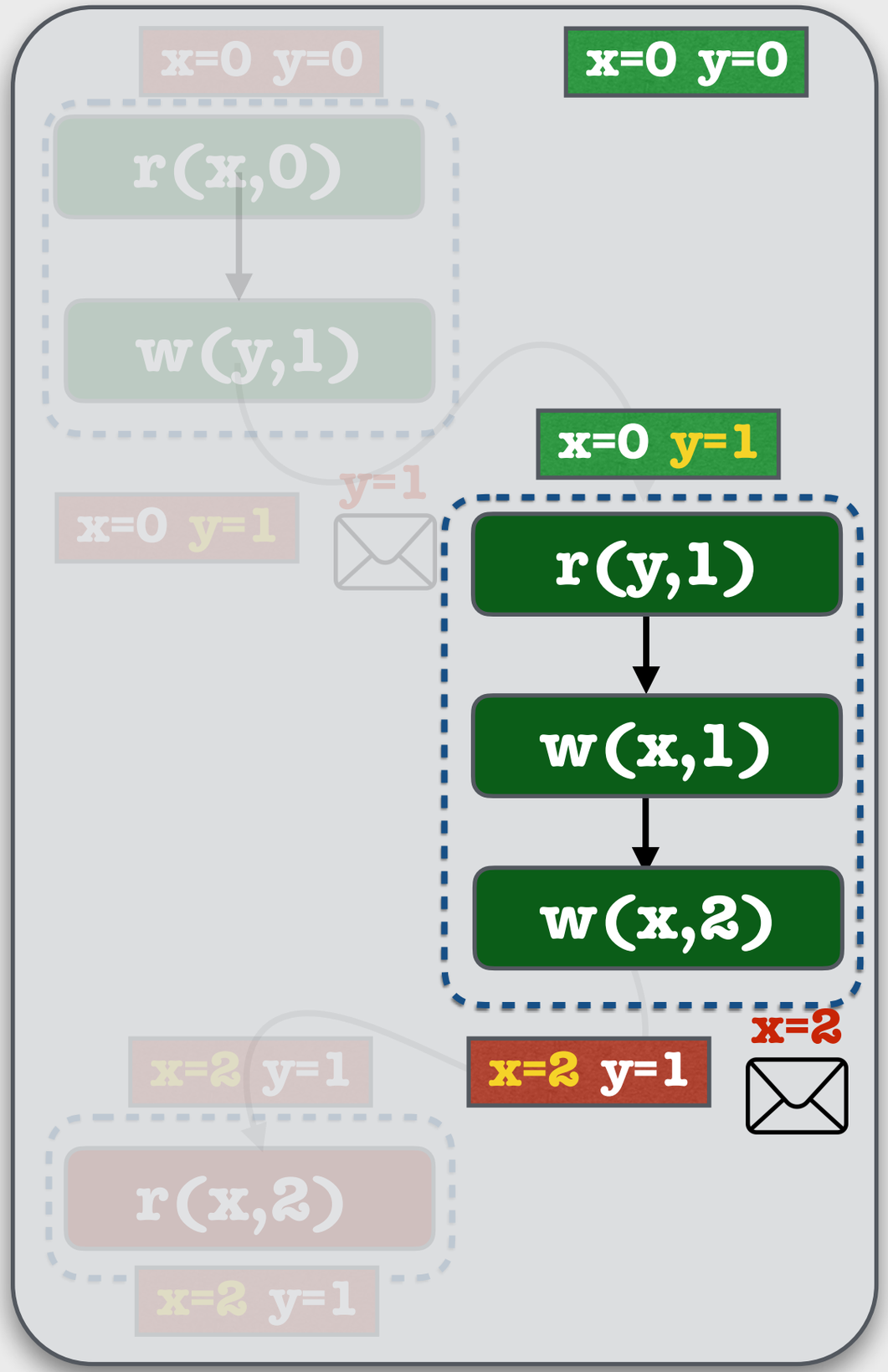
Locality

Validation



Locality

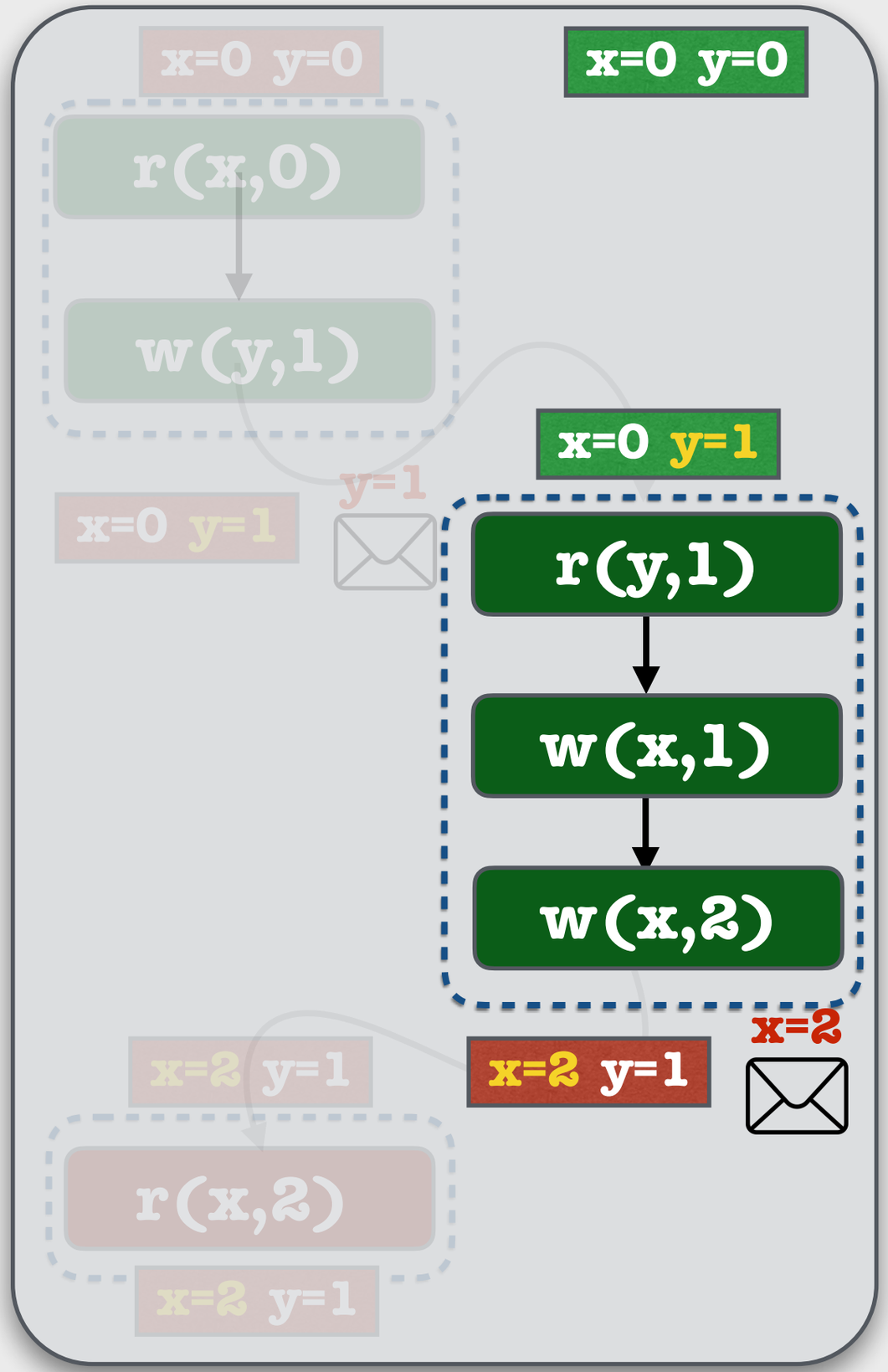
Validation



Simulate **P2** under **SC**

Locality

Validation

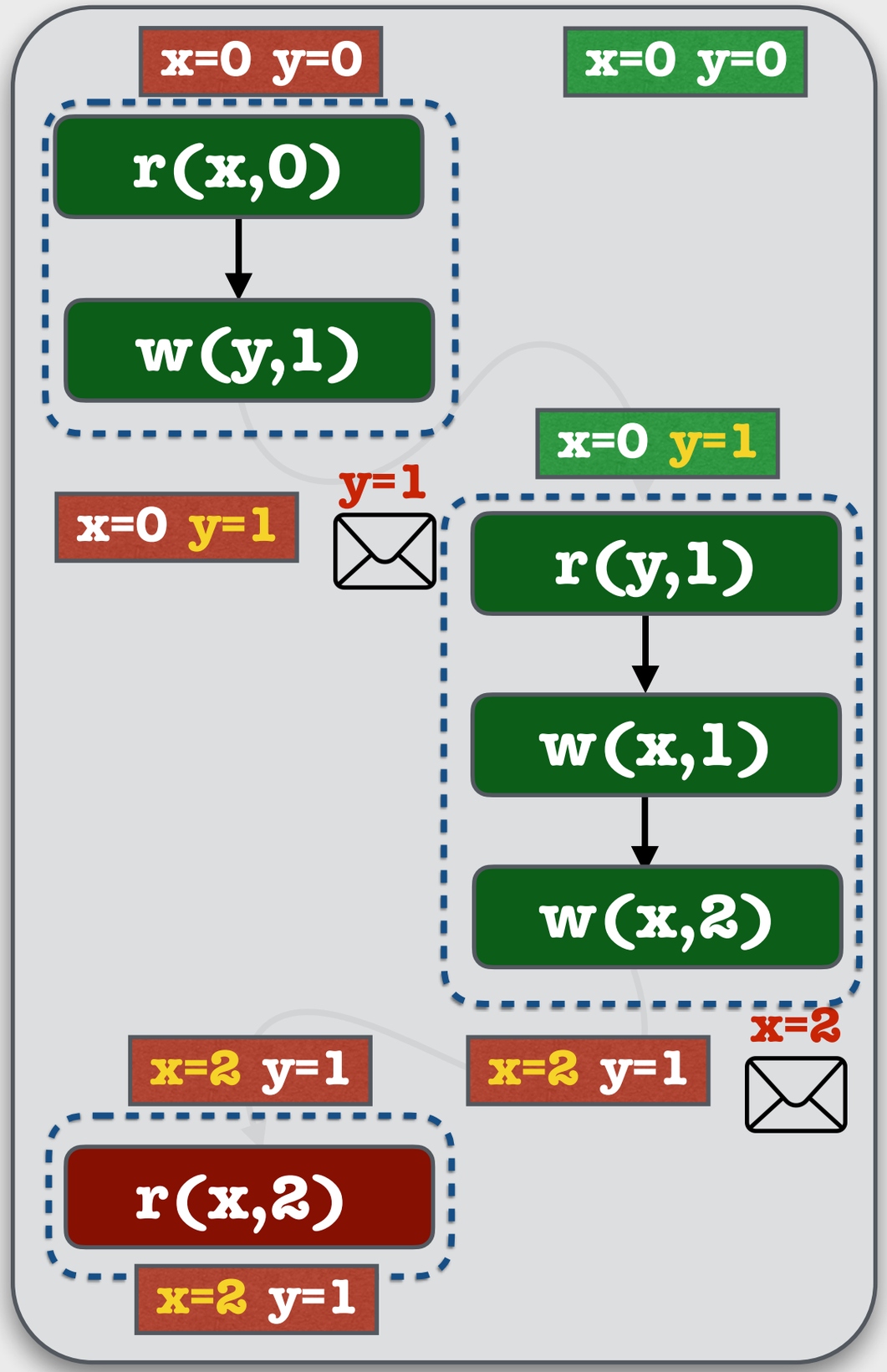


Simulate **P2** under **SC**

Local and global data structures
to simulate views and the memory pool

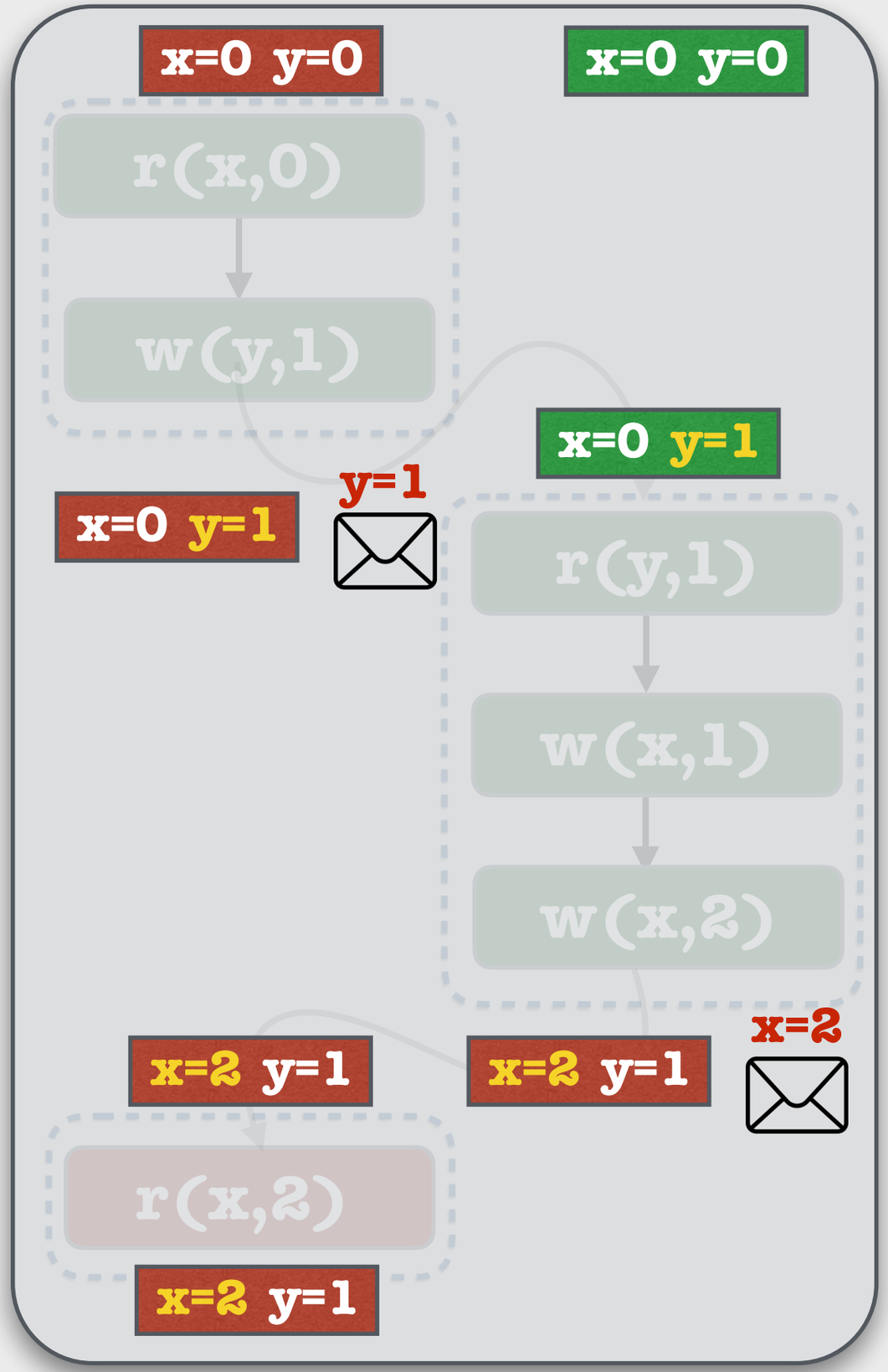
Locality

Validation



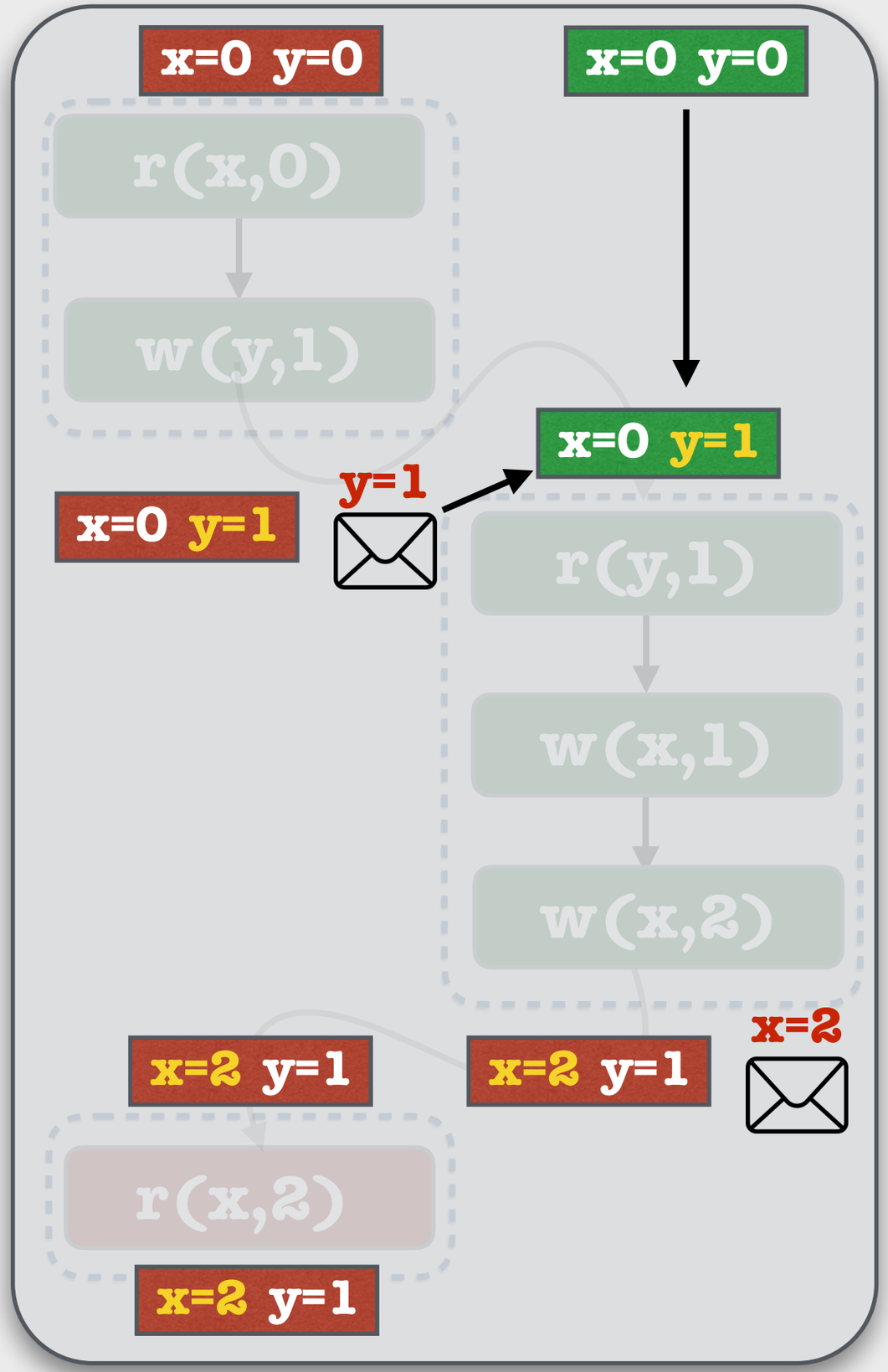
Locality

Validation



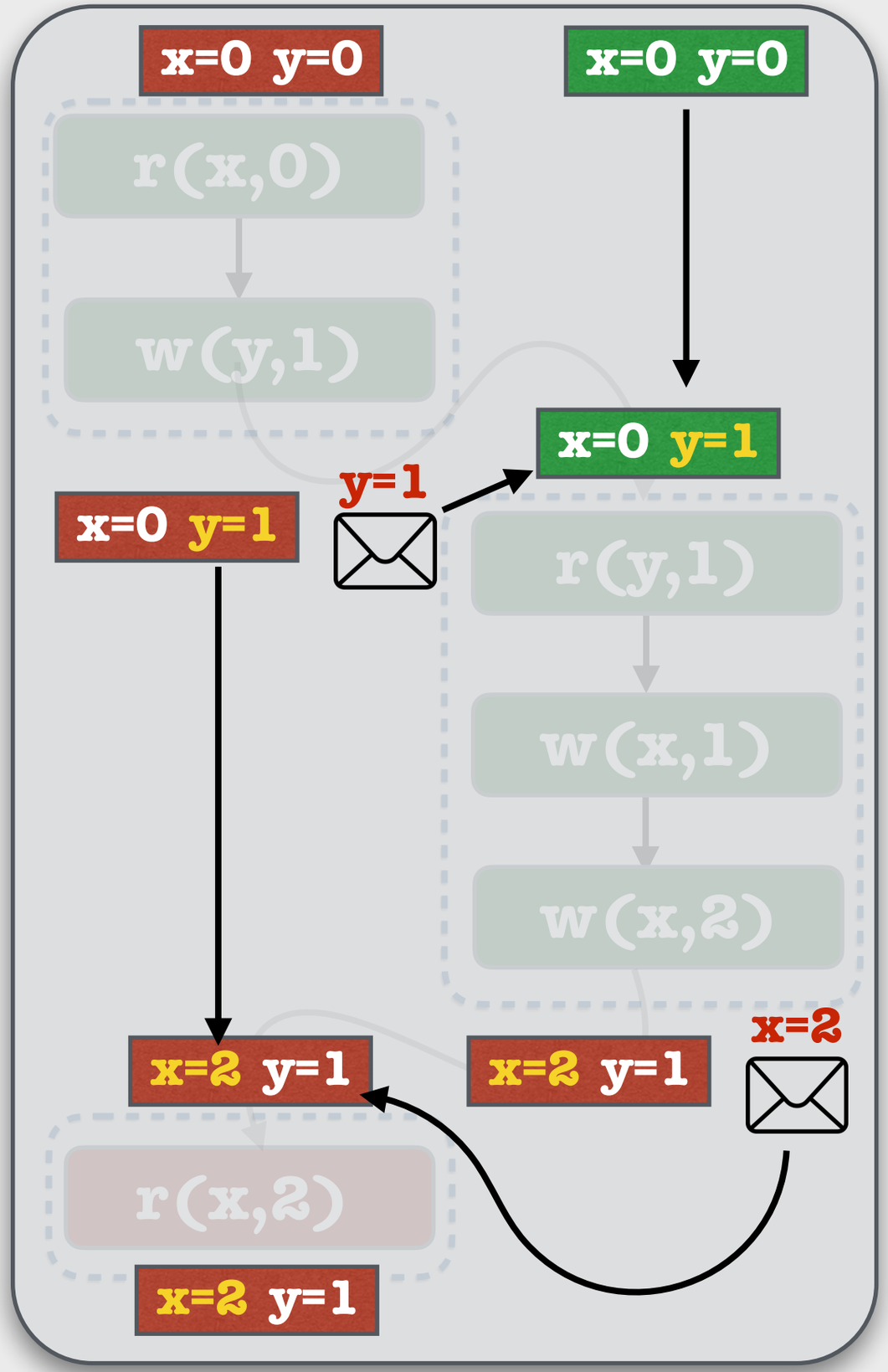
Locality

Validation



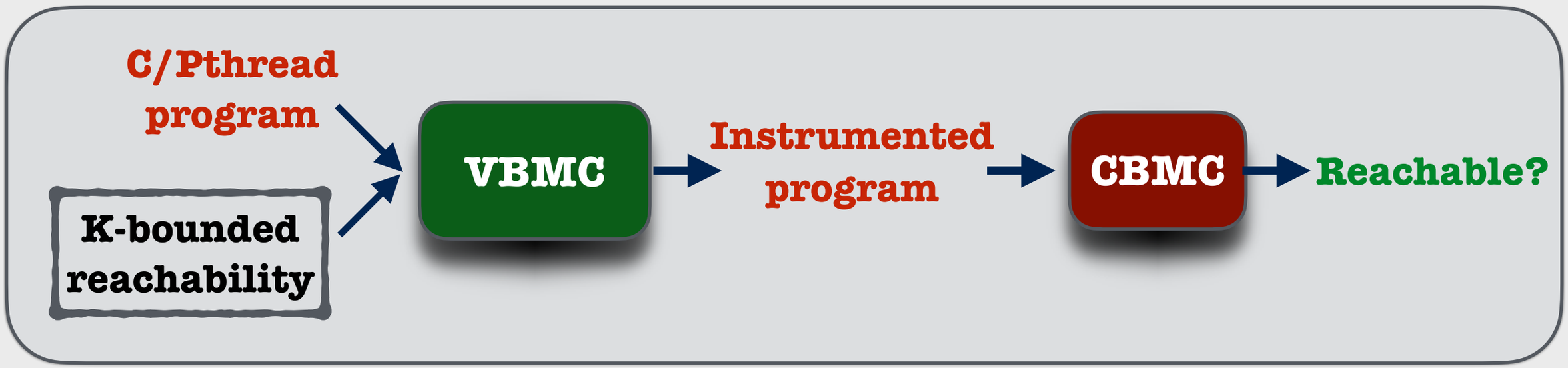
Locality

Validation



View Bounded Model Checker (VBMC)

- Using CBMC as backend model checker



View Bounded Model Checker (VBMC)

- ✦ Tested with 4004 litmus tests [Sarkar et al. 2011]:
  - Same results as Herd [Alglave et al. 2014]
- ✦ Tested on concurrent benchmarks:
  - Few number of contexts sufficient for bug detection under RA
  - Catches isolated bugs faster than state of the art SMC tools Tracer, RCMC and CDSChecker

Parameterized Reachability in RA

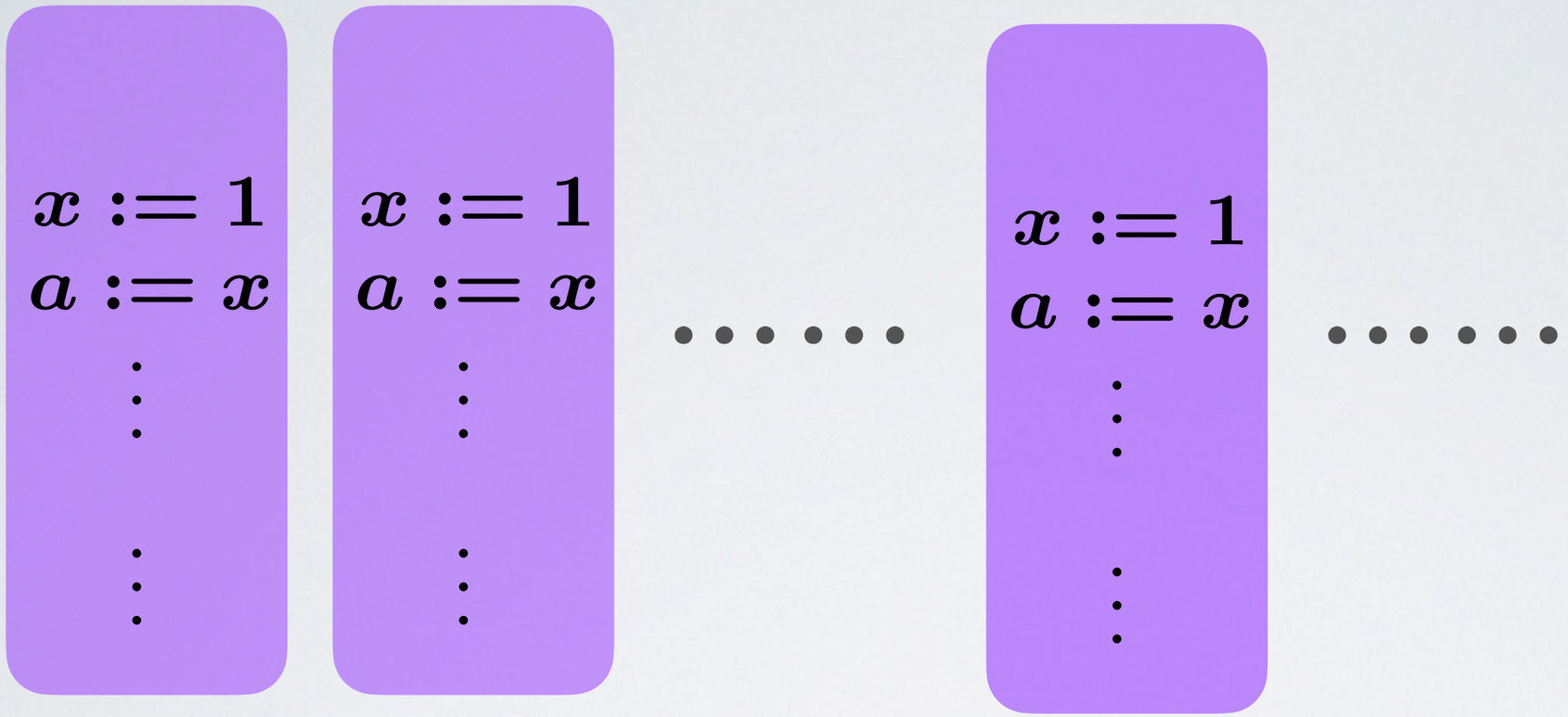
Parameterized Reachability in RA



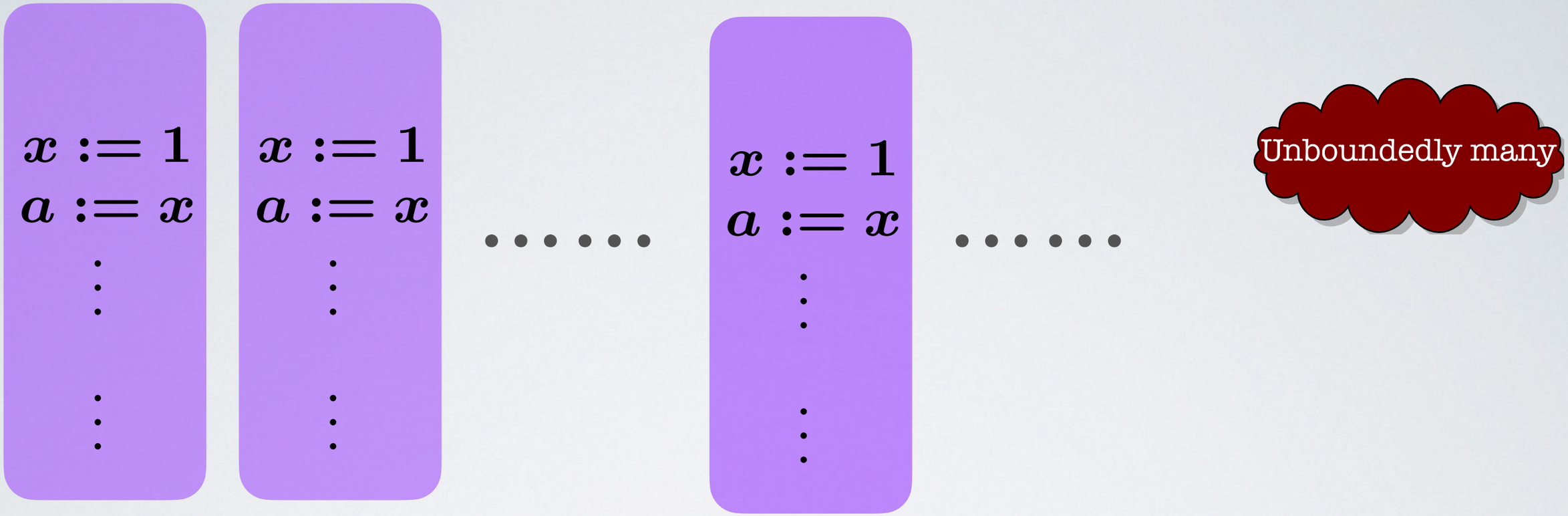

PODC 2022

Parameterized Reachability

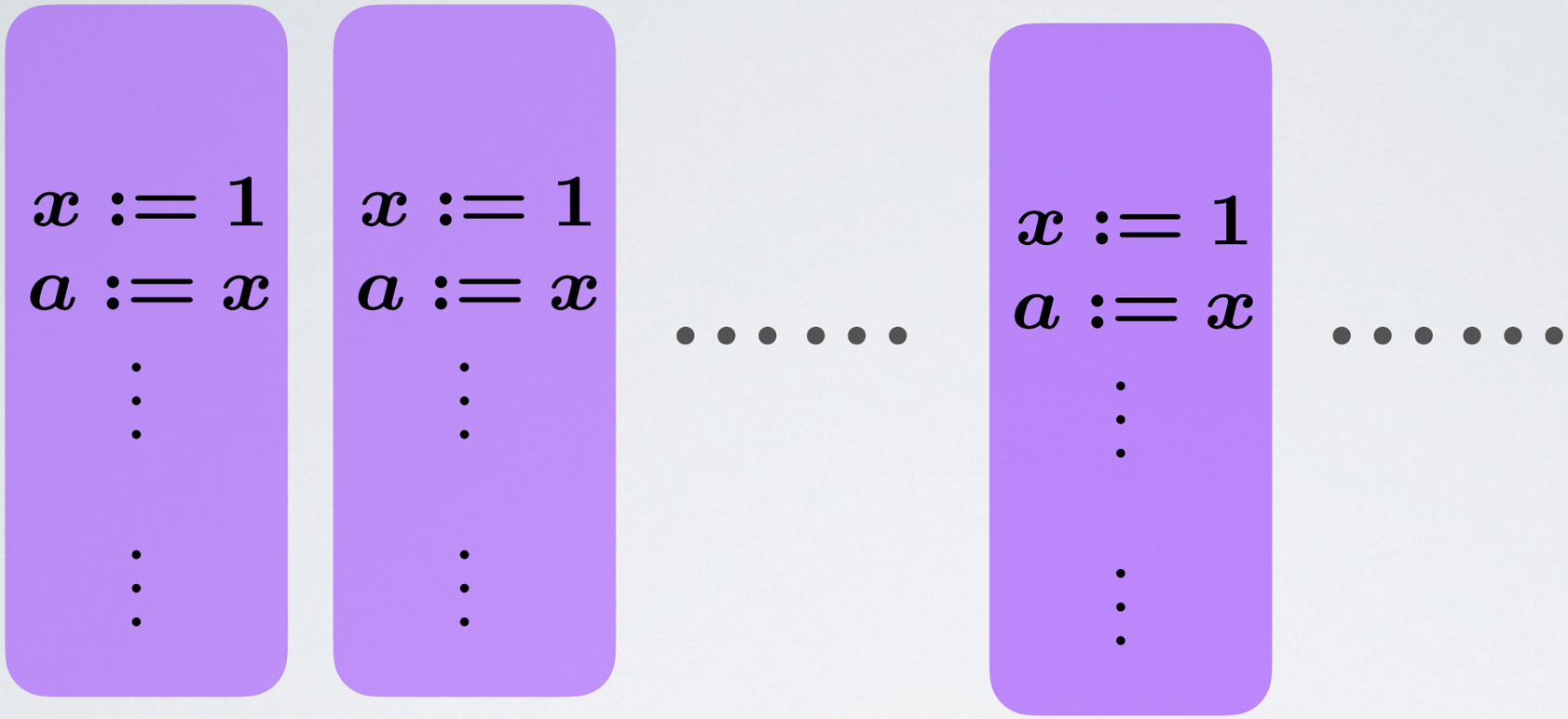
Parameterized Reachability



Parameterized Reachability



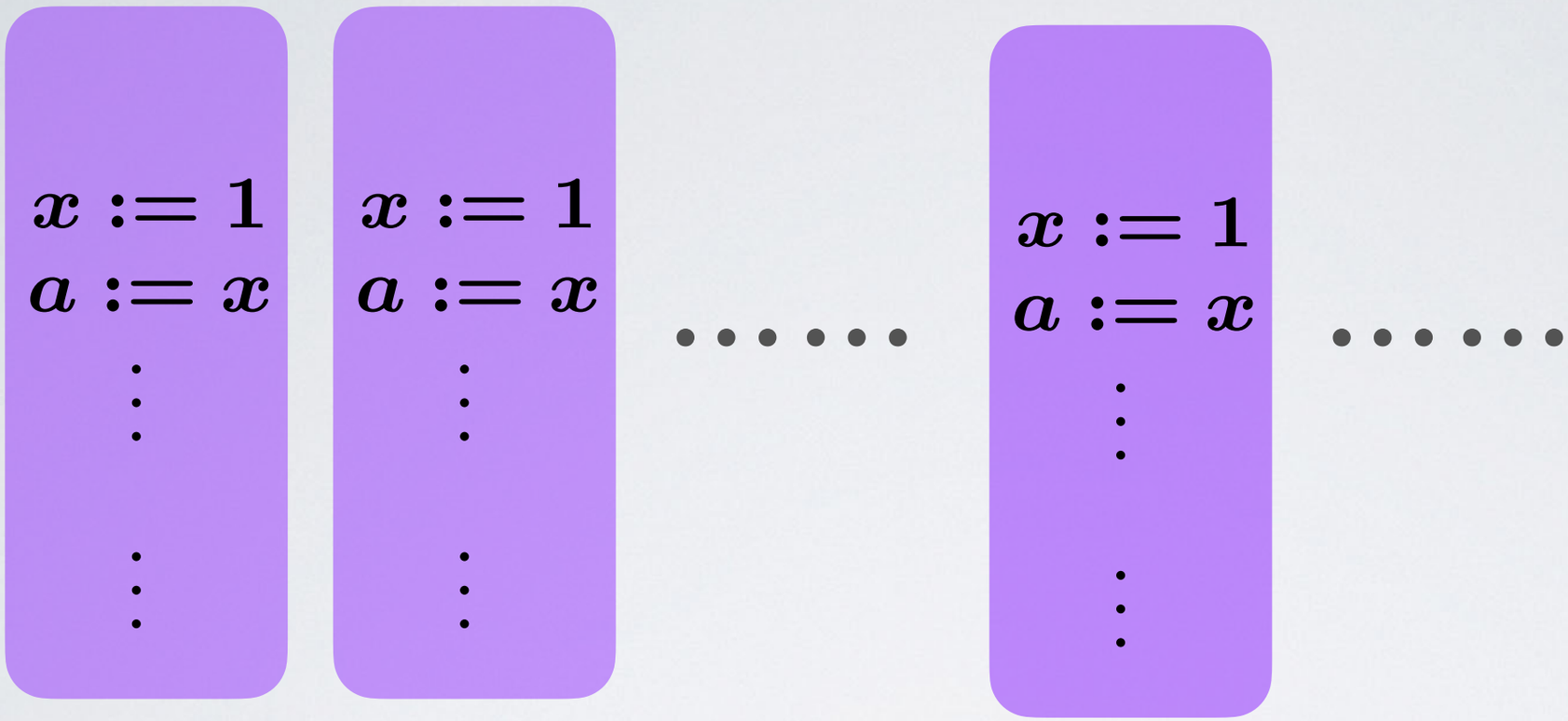
Parameterized Reachability



Unboundedly many

Identical threads

Parameterized Reachability

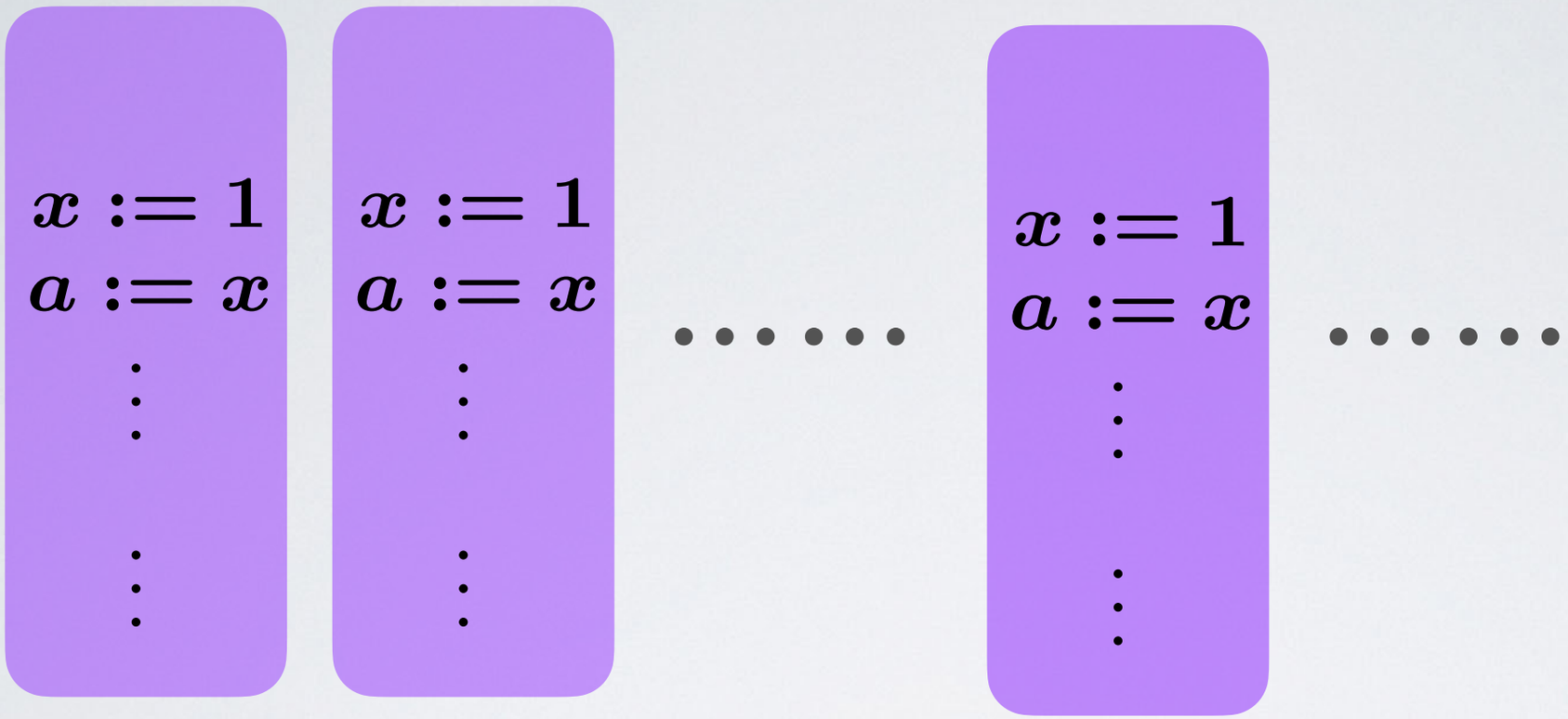


Unboundedly many

Identical threads

No CAS

Parameterized Reachability



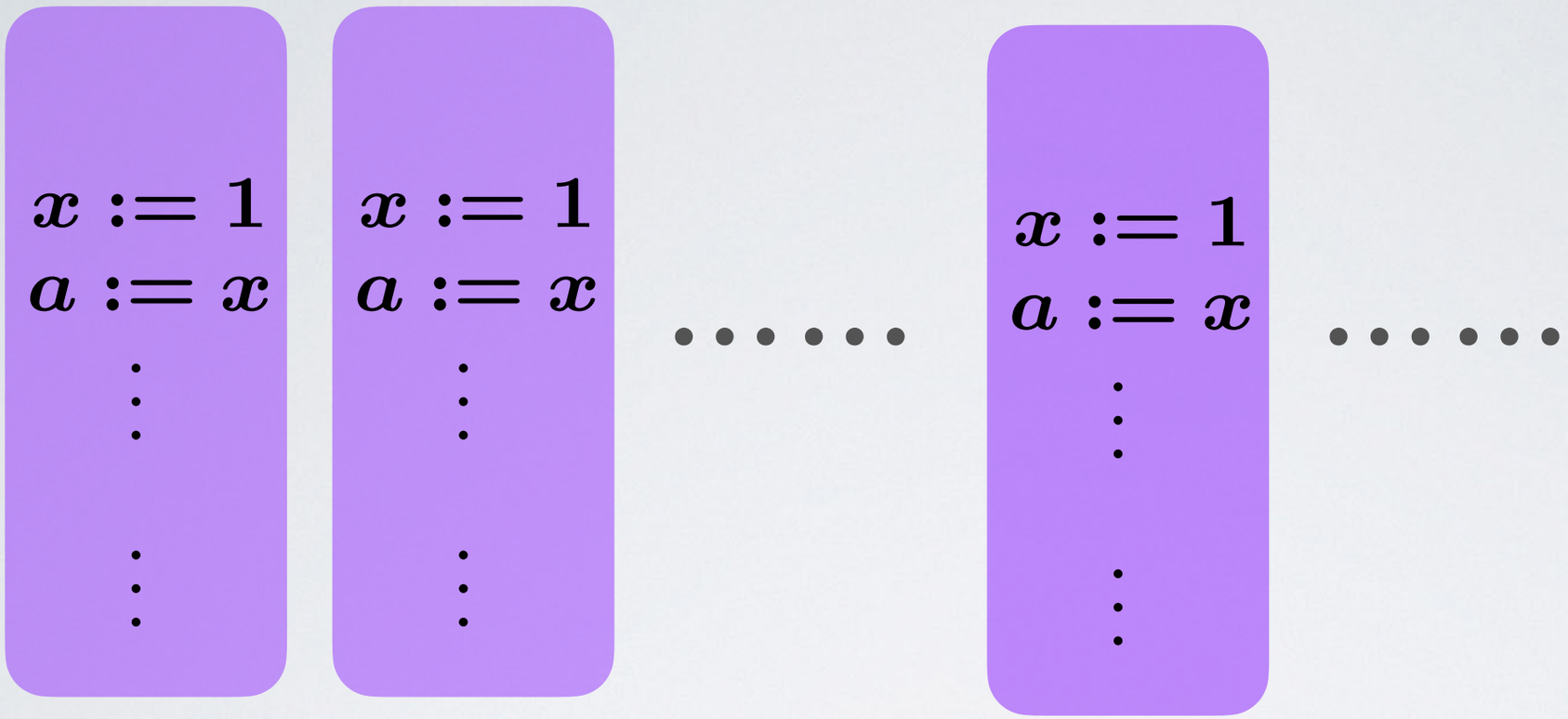
Unboundedly many

Identical threads

No CAS

Allowing CAS operations render state reachability undecidable for parameterized **RA**, even with acyclic, identical threads

Parameterized Reachability



Unboundedly many

Identical threads

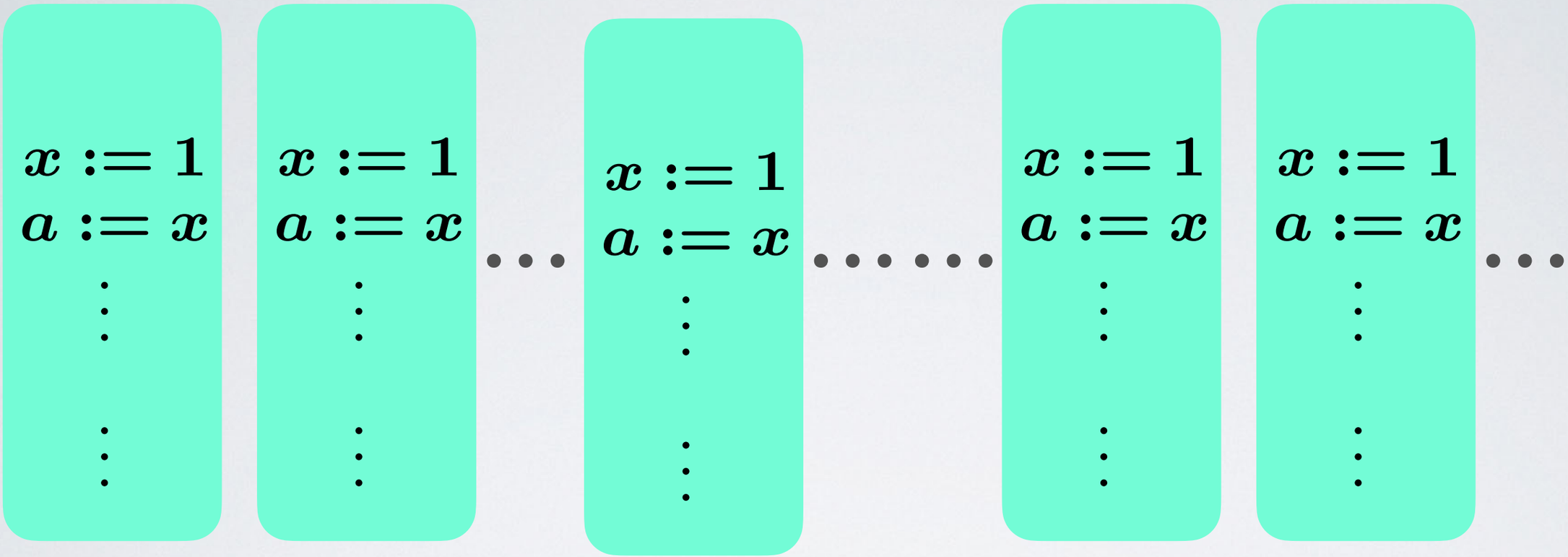
No CAS

Allowing CAS operations render state reachability undecidable for parameterized **RA**, even with acyclic, identical threads

Simulate the non parameterized setting

Simplified Semantics

Simplified Semantics



Simplified Semantics

generates (x,l,t)

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

unbounded, identical, no CAS

Simplified Semantics

generates (x,l,t)

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

unbounded, identical, no CAS

generates
(x,l,t+i)

Simplified Semantics

generates (x,l,t)

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

unbounded, identical, no CAS

generates (x,l,t+i)

generates (x,l,t+i+j)

Simplified Semantics

generates (x,l,t)

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

unbounded, identical, no CAS

generates (x,l,t+i)

generates (x,l,t+i+j)

$\ldots\ldots$

Simplified Semantics

generates (x,l,t)

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

unbounded, identical, no CAS

generates (x,l,t+i)

generates (x,l,t+i+j)

$\ldots\ldots$

generates (x,l,t+i+j+k)

Simplified Semantics

generates (x,l,t)

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

unbounded, identical, no CAS

generates (x,l,t+i)

generates (x,l,t+i+j)

$\ldots\ldots$

generates (x,l,t+i+j+k)

$\ldots\ldots$

Simplified Semantics

generates (x,l,t)

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

unbounded, identical, no CAS

generates (x,l,t+i)

generates (x,l,t+i+j)

$\ldots\ldots$

generates (x,l,t+i+j+k)

$\ldots\ldots$

If (x,l) can be read, it can be read arbitrary many times

Simplified Semantics

generates (x,l,t)

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$\vdots$

$\vdots$

$\ldots$

Infinite Supply Property

unbounded, identical, no CAS

generates (x,l,t+i)

generates (x,l,t+i+j)

$\ldots\ldots$

generates (x,l,t+i+j+k)

$\ldots\ldots$

If (x,l) can be read, it can be read arbitrary many times

Simplified Semantics

generates (x,l,t)

$x := 1$

$a := x$

Infinite Supply Property

No recency check needed

unbounded, identical, no CAS

generates (x,l,t+i)

generates (x,l,t+i+j)

generates (x,l,t+i+j+k)

If (x,l) can be read, it can be read arbitrary many times

Simplified Semantics

generates (x,l,t)

$x := 1$

$a := x$

Infinite Supply Property

No recency check needed

Abstraction of timestamps

unbounded, identical, no CAS

generates (x,l,t+i)

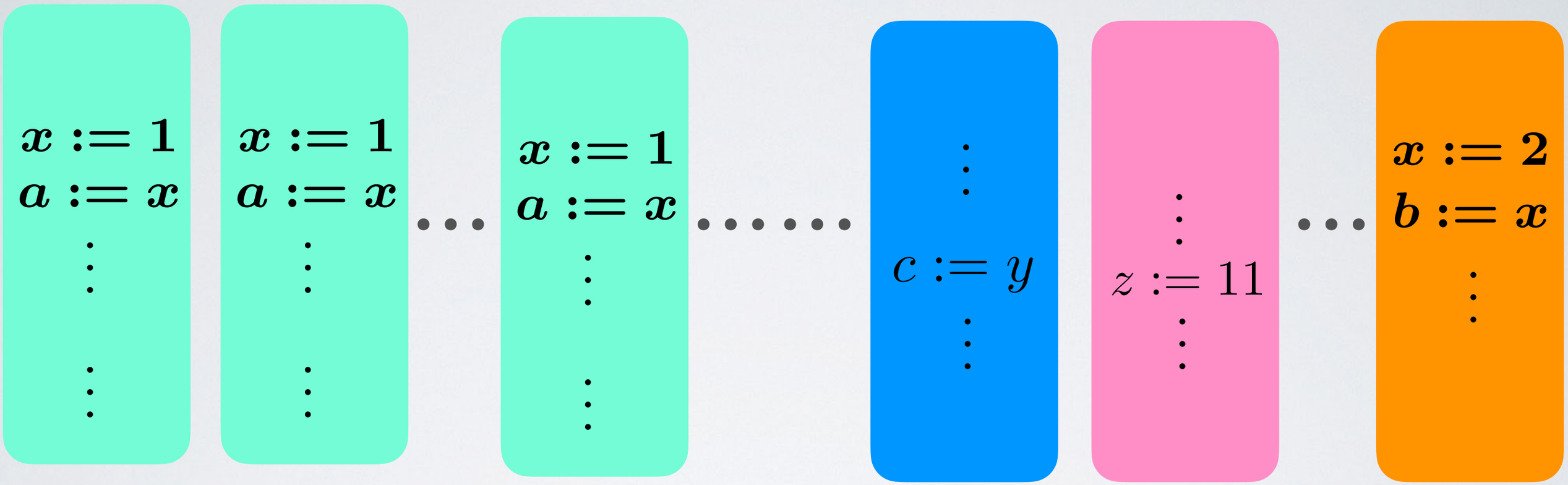
generates (x,l,t+i+j)

generates (x,l,t+i+j+k)

If (x,l) can be read, it can be read arbitrary many times

Simplified Semantics

Simplified Semantics



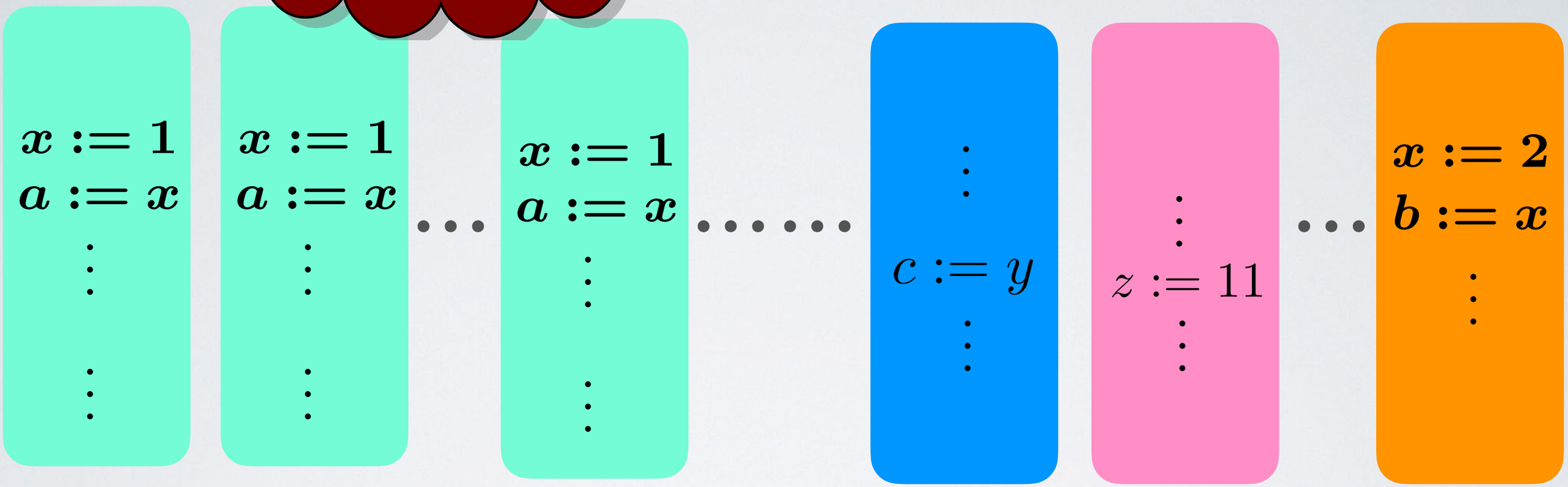
unbounded, identical no CAS

acyclic, distinguished

has CAS

Simplified Semantics

Maintain relative ordering with distinguished threads



unbounded, identical no CAS

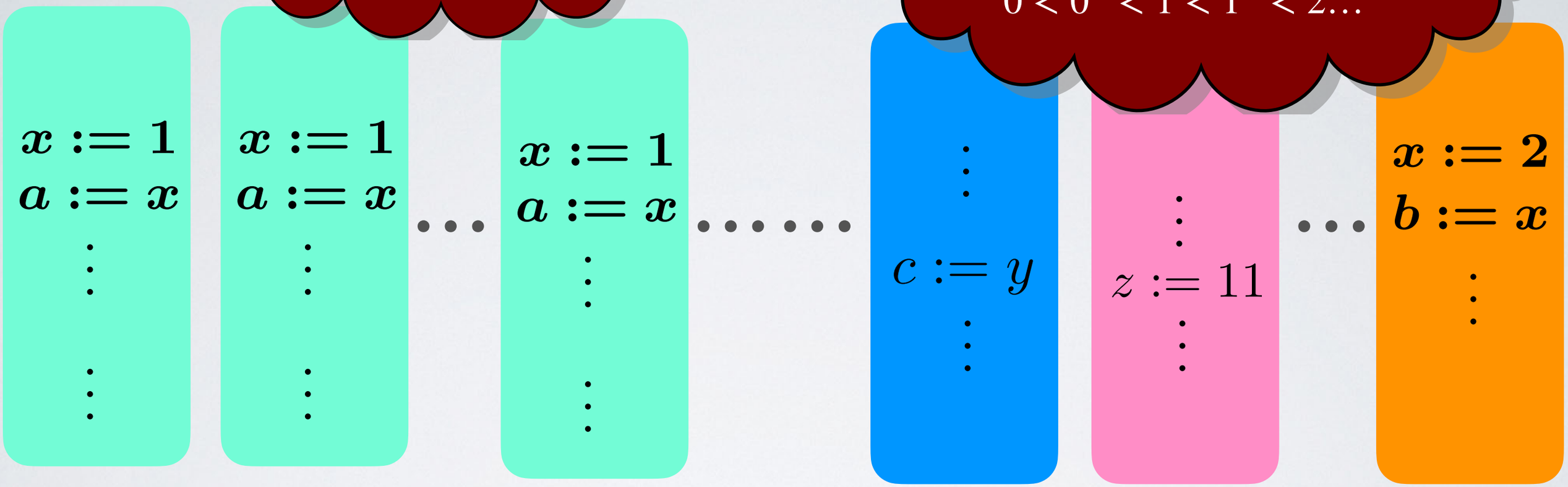
acyclic, distinguished

has CAS

Simplified Semantics

Maintain relative ordering with distinguished threads

env timestamps $\{0^+,1^+,2^+,\ldots\}$
dis timestamps $\{0,1,2,\ldots\}$
$0 < 0^+ < 1 < 1^+ < 2\ldots$



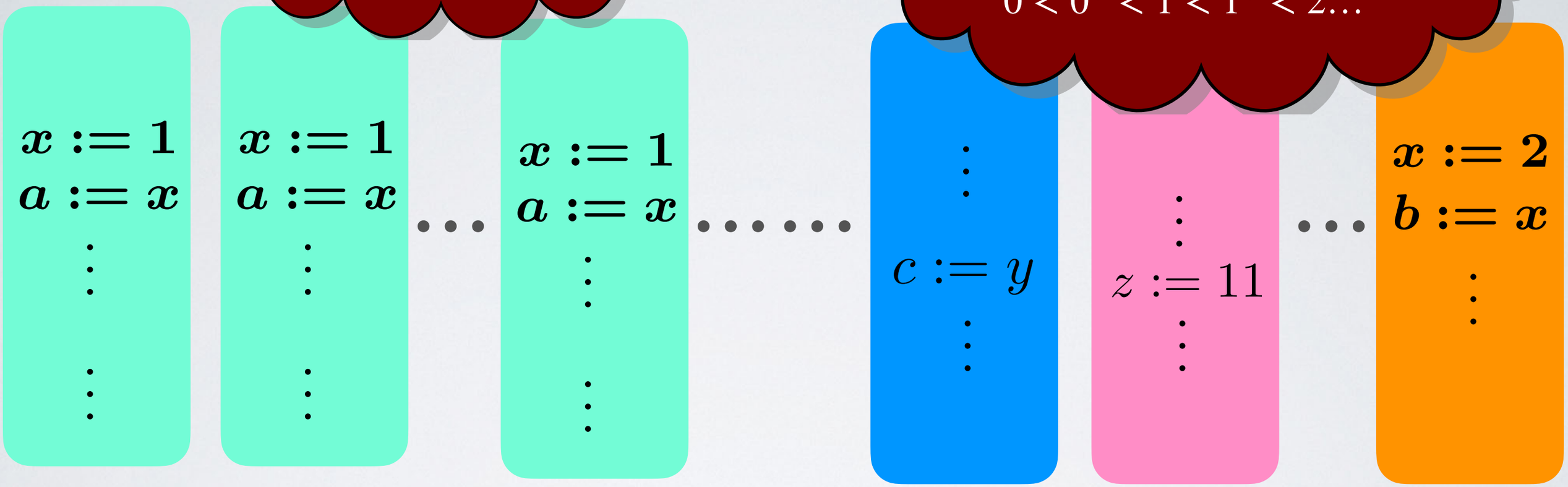
unbounded, identical no CAS

acyclic, distinguished
has CAS

Simplified Semantics

Maintain relative ordering with distinguished threads

env timestamps $\{0^+, 1^+, 2^+, \ldots\}$
dis timestamps $\{0, 1, 2, \ldots\}$
$0 < 0^+ < 1 < 1^+ < 2 \ldots$



unbounded, identical no CAS

acyclic, distinguished

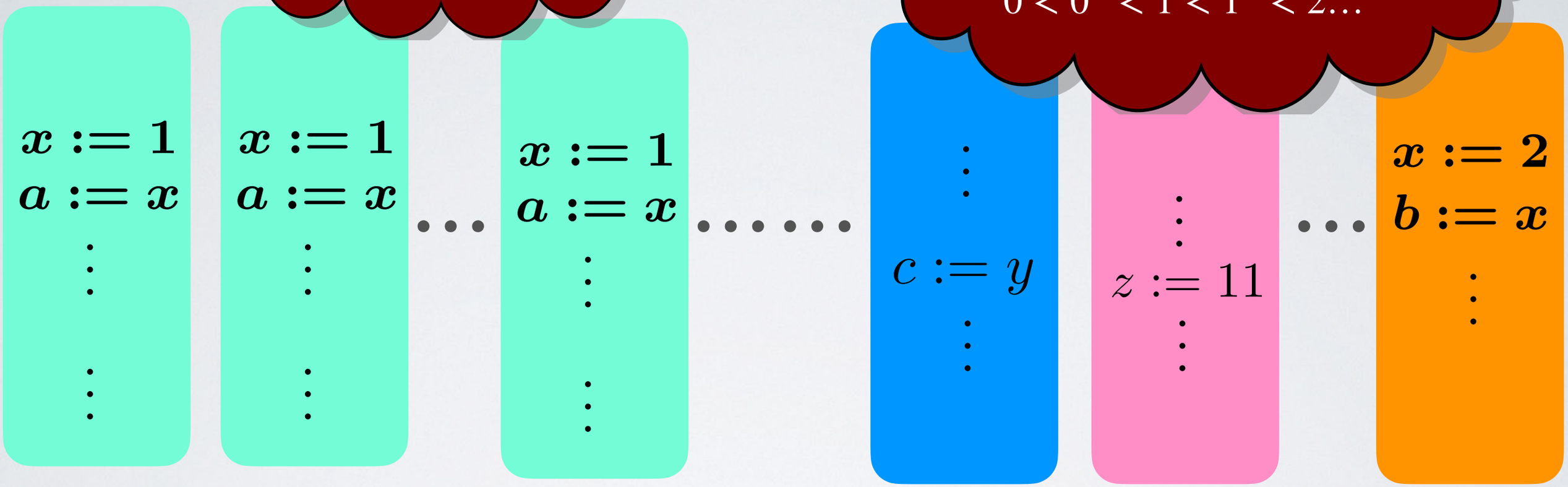
has CAS

Concrete time stamps 0 2 3 5 7 8 10 11 13

Simplified Semantics

Maintain relative ordering with distinguished threads

env timestamps $\{0^+, 1^+, 2^+, \ldots\}$
dis timestamps $\{0, 1, 2, \ldots\}$
$0 < 0^+ < 1 < 1^+ < 2 \ldots$



unbounded, identical no CAS

acyclic, distinguished
has CAS

Concrete time stamps	0	2	3	5	7	8	10	11	13
Abstract time stamps	0	1	1^+	2	2^+	2^+	3	3^+	3^+

Simplified Semantics

Maintain relative ordering with distinguished threads

env timestamps $\{0^+,1^+,2^+,\ldots\}$
dis timestamps $\{0,1,2,\ldots\}$
$0 < 0^+ < 1 < 1^+ < 2\ldots$

$x := 1$
$a := x$
$\vdots$

$x := 1$
$a := x$
$\vdots$

$\ldots$

$x := 1$
$a := x$
$\vdots$

$\ldots\ldots$

$\vdots$
$c := y$

$\vdots$
$z := 11$
$\vdots$

$\ldots$

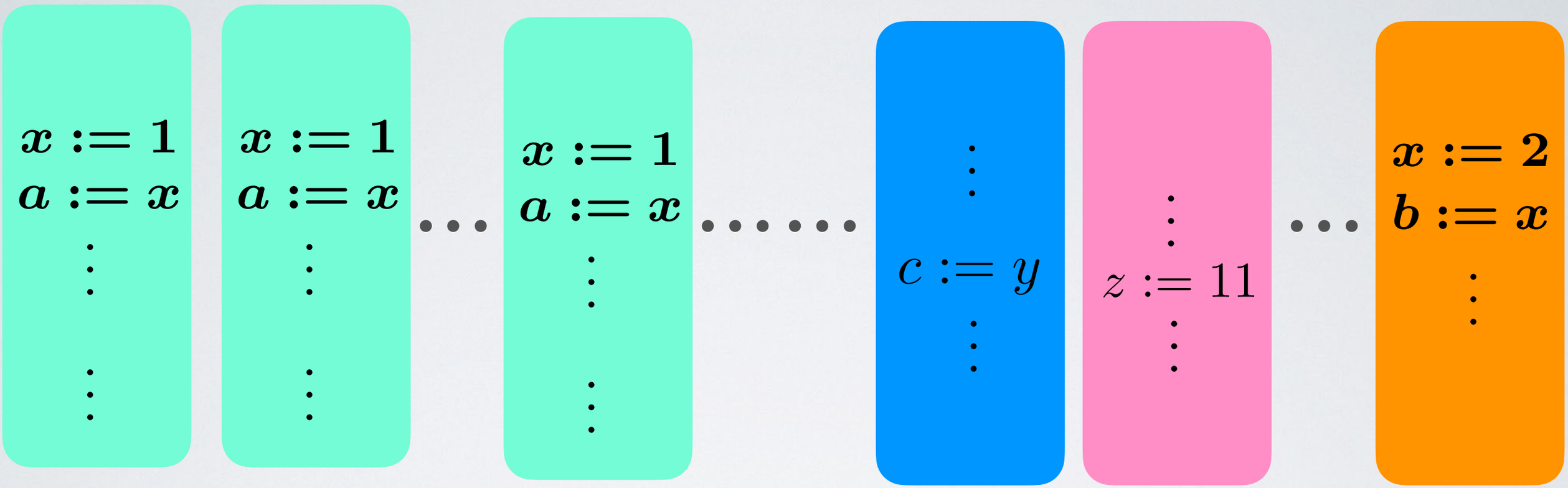
$x := 2$
$b := x$
$\vdots$

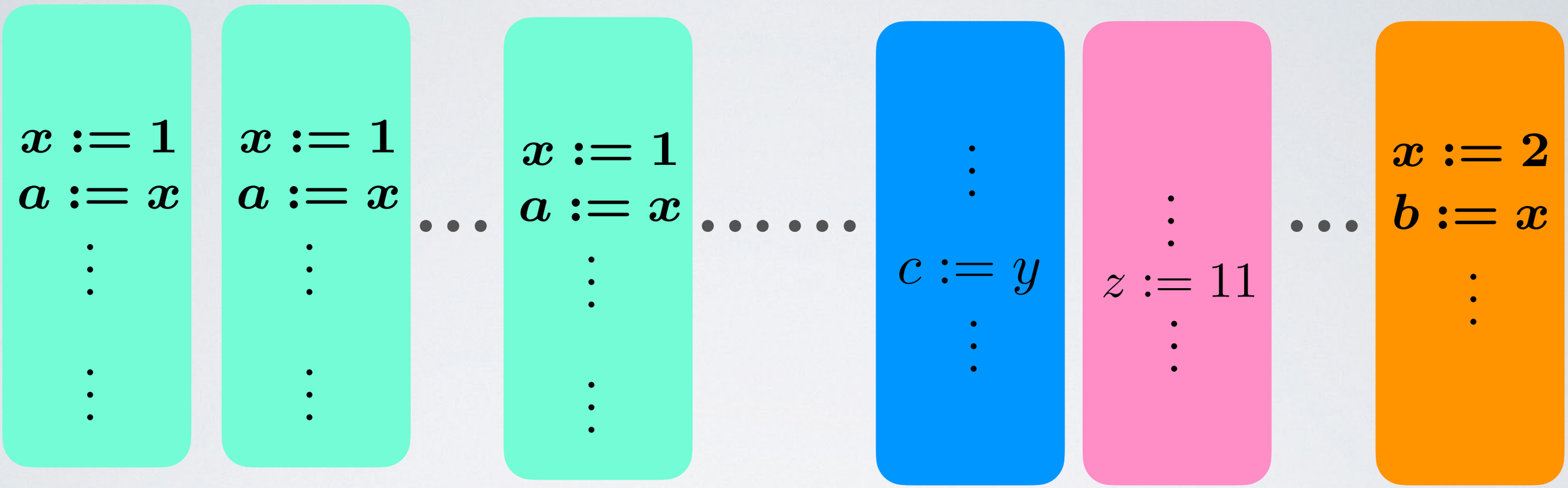
#timestamps=O(#(dis timestamps))

unbounded, identic… …cyclic, distinguished

has CAS

Concrete time stamps	0	2	3	5	7	8	10	11	13
Abstract time stamps	0	1	1^+	2	2^+	2^+	3	3^+	3^+

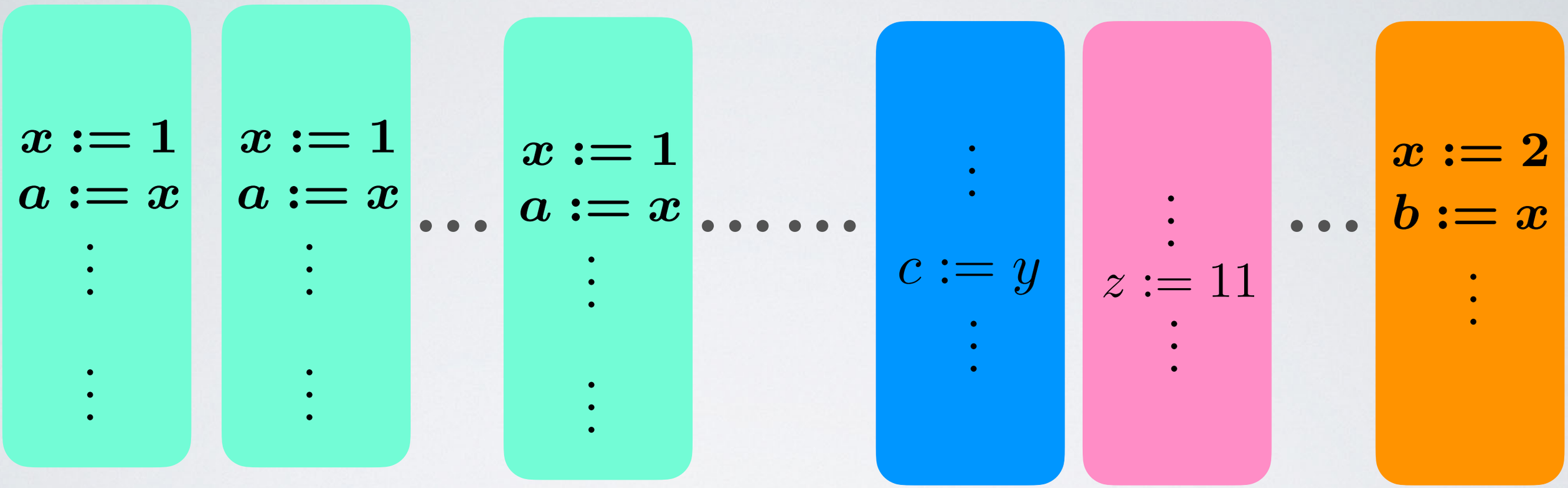

$x := 1$
$a := x$
$x := 1$
$a := x$
$x := 1$
$a := x$
$c := y$
$z := 11$
$x := 2$
$b := x$



unbounded, identical
no CAS

acyclic, distinguished
has CAS

QBF Sat



unbounded, identical

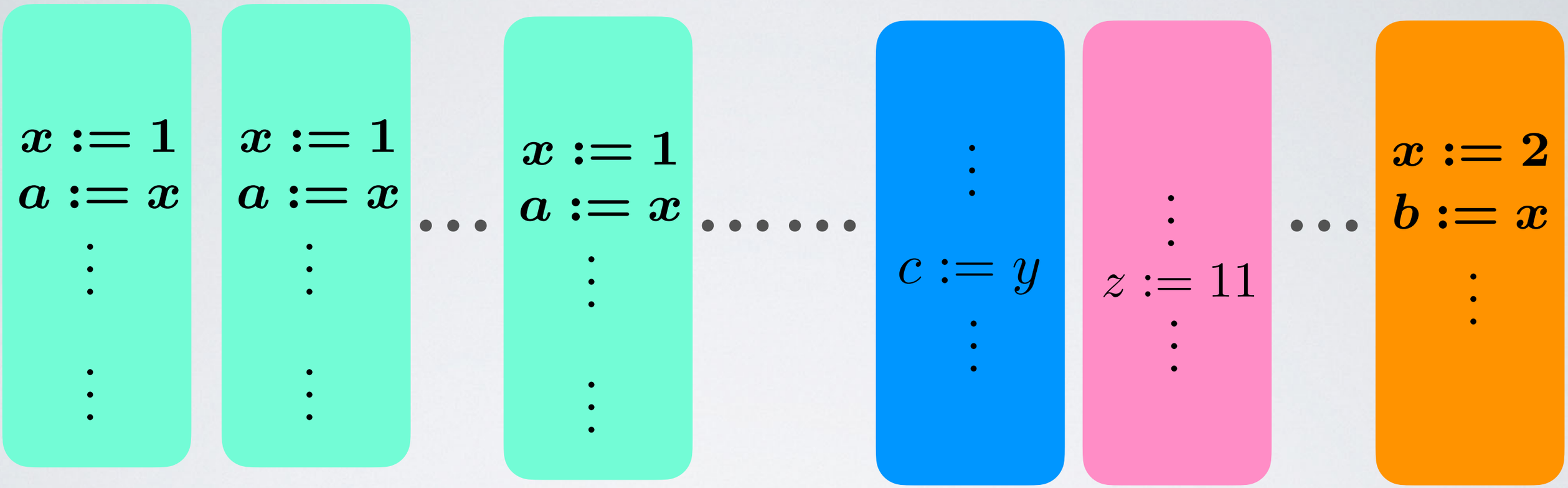
no CAS

acyclic, distinguished

has CAS

QBF Sat

$\leq$

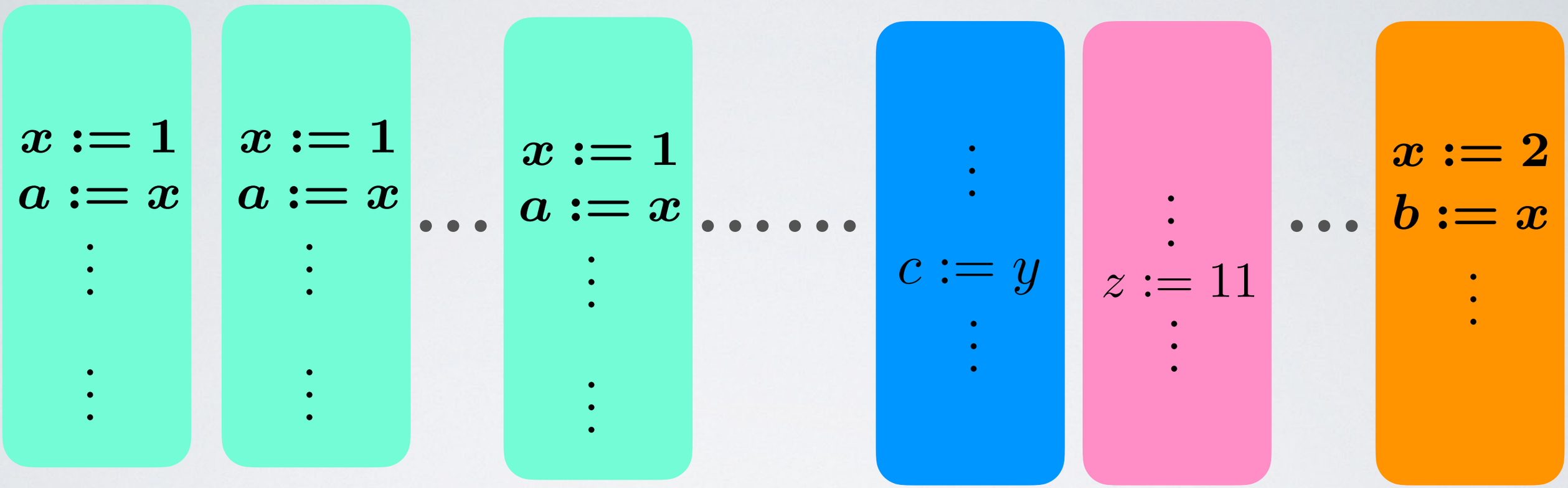


unbounded, identical
no CAS

acyclic, distinguished
has CAS

QBF Sat

$\leq$



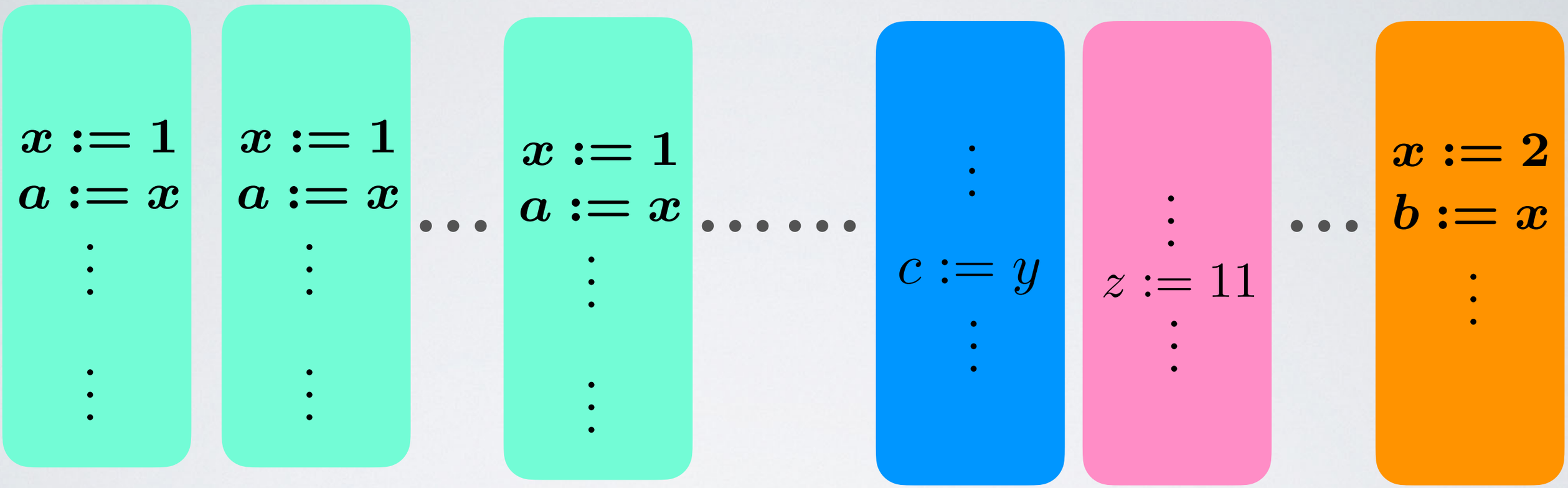
unbounded, identical
no CAS

acyclic, distinguished
has CAS

$\leq$

QBF Sat

$\leq$



unbounded, identical
no CAS

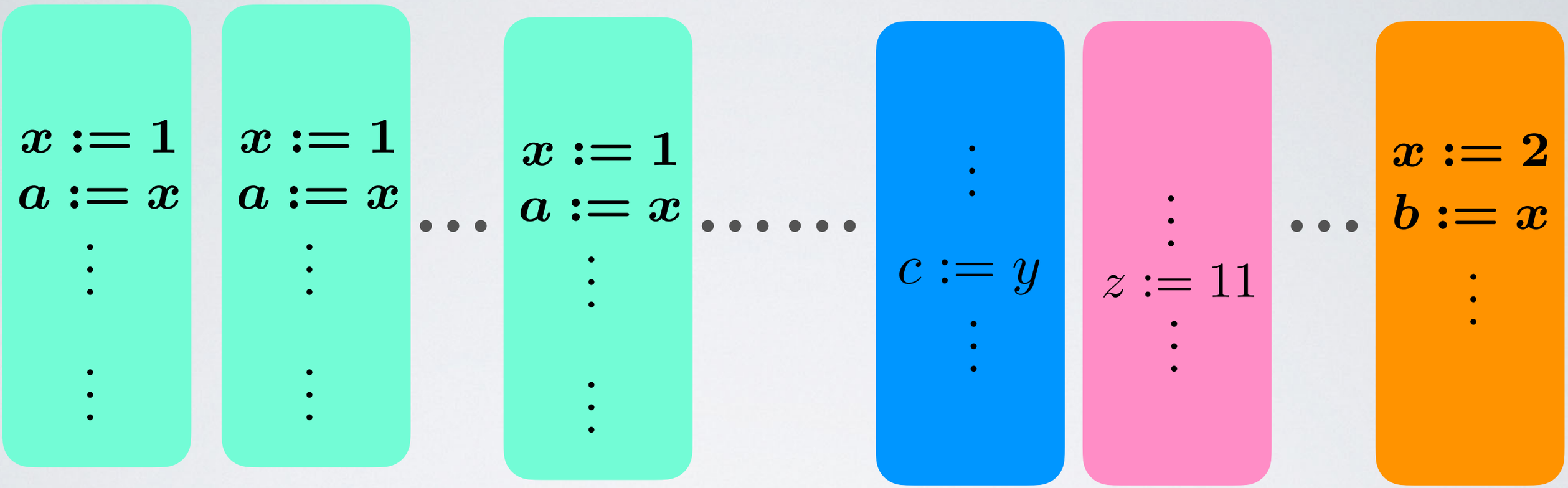
acyclic, distinguished
has CAS

$\leq$

Query evaluation in linear Datalog

QBF Sat

$\leq$



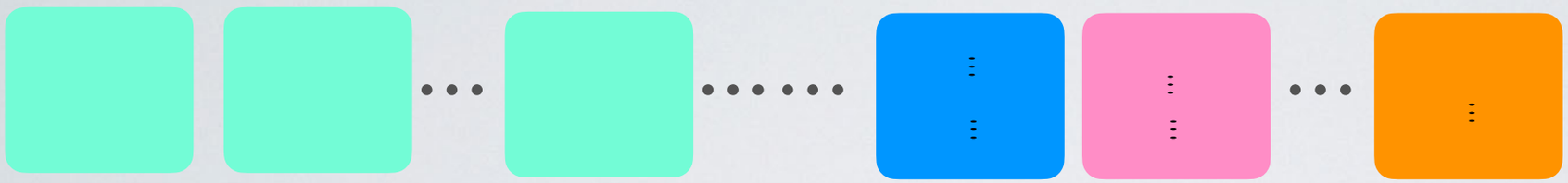
unbounded, identical
no CAS

acyclic, distinguished
has CAS

$\leq$

Query evaluation in linear Datalog

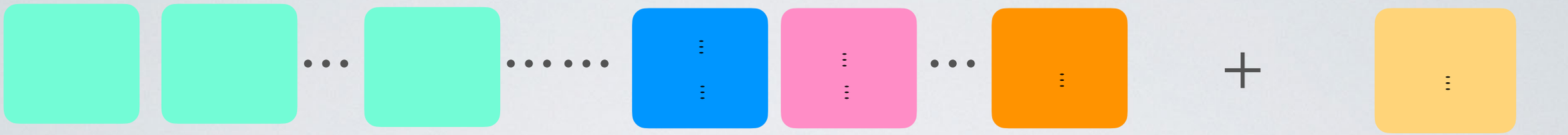
PSPACE completeness



unbounded, identical
no CAS

acyclic, distinguished

has CAS

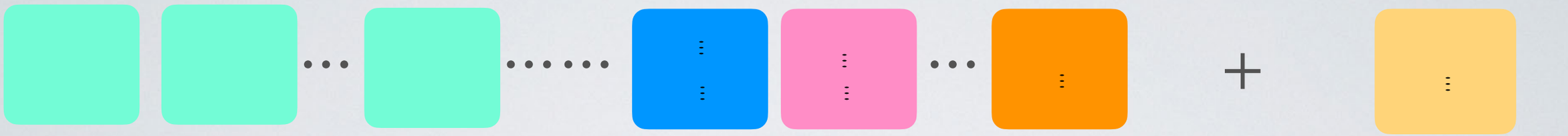


unbounded, identical
no CAS

acyclic, distinguished

has CAS

unrestricted



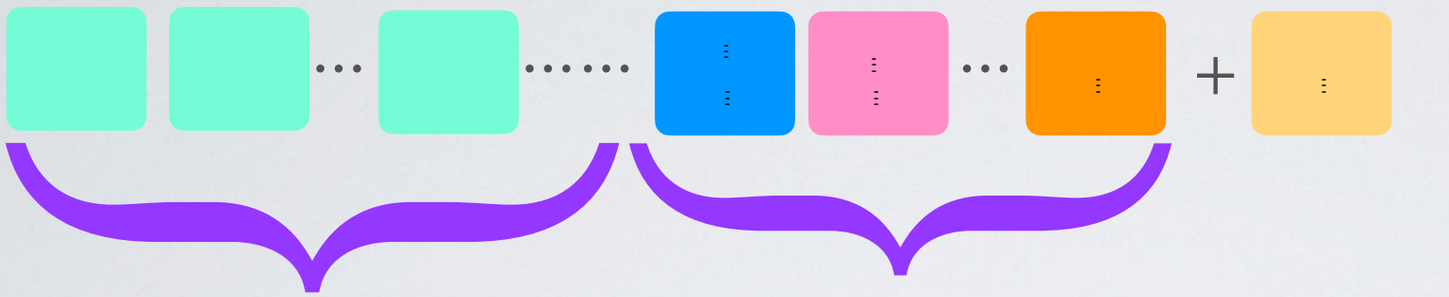
unbounded, identical
no CAS

acyclic, distinguished

has CAS

unrestricted

NEXPTIME completeness



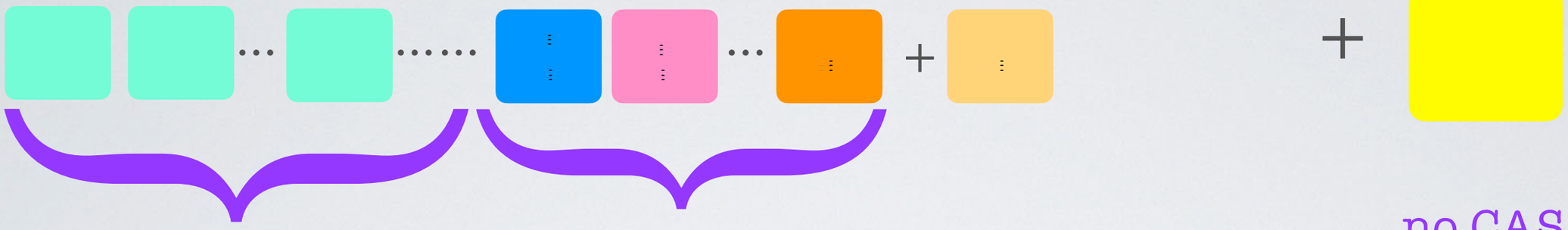
unbounded, identical

no CAS

acyclic, distinguished

has CAS

unrestricted



Non primitive recursive

Decidability Open

Thankyou