WALLPAPERSWIDE.COM

Shared Variable Concurrency

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

WALLPAPERSWIDE.COM

Shared Variable Concurrency



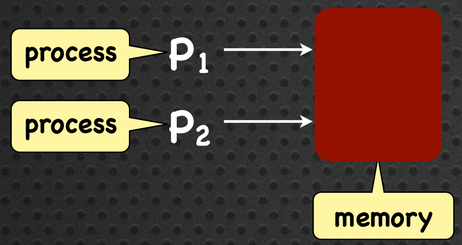
WALLPAPERSWIDE.COM

Shared Variable Concurrency



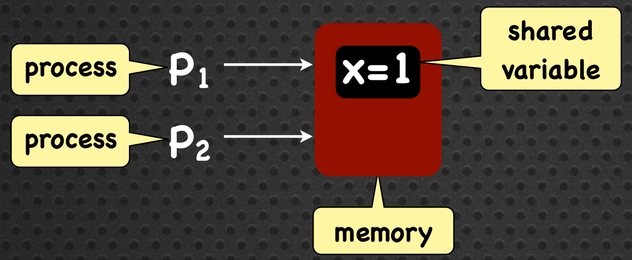
WALLPAPERSWIDE.COM

Shared Variable Concurrency



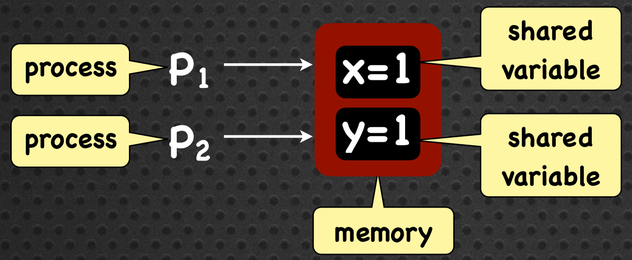
WALLPAPERSWIDE.COM

Shared Variable Concurrency



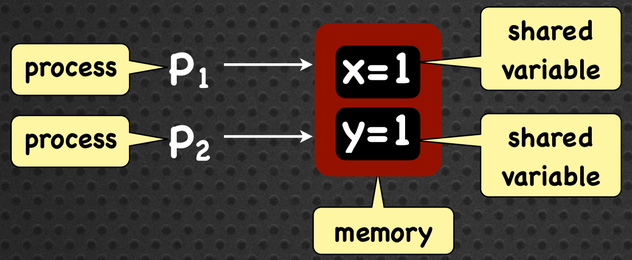
WALLPAPERSWIDE.COM

Shared Variable Concurrency



WALLPAPERSWIDE.COM

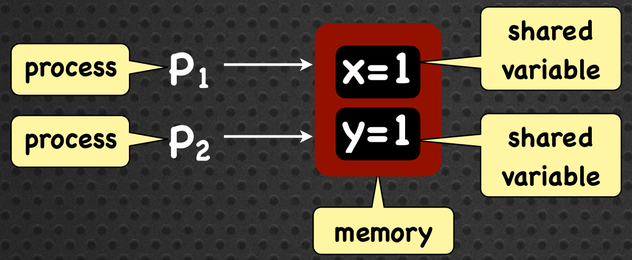
Shared Variable Concurrency



WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

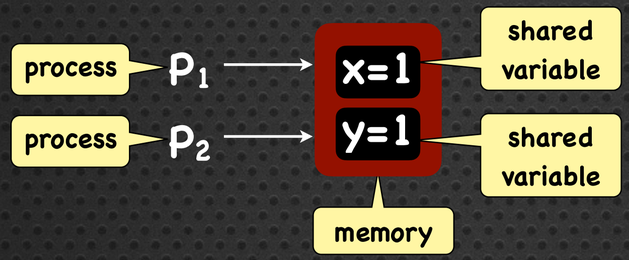


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations



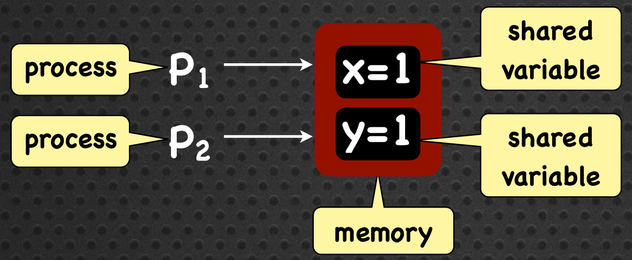
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



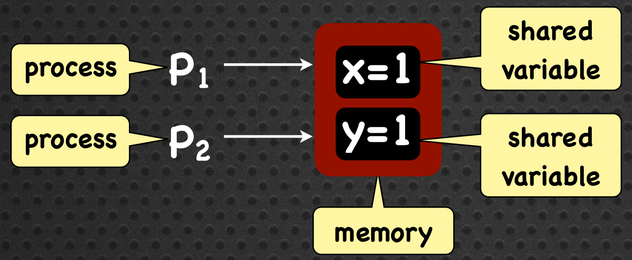
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



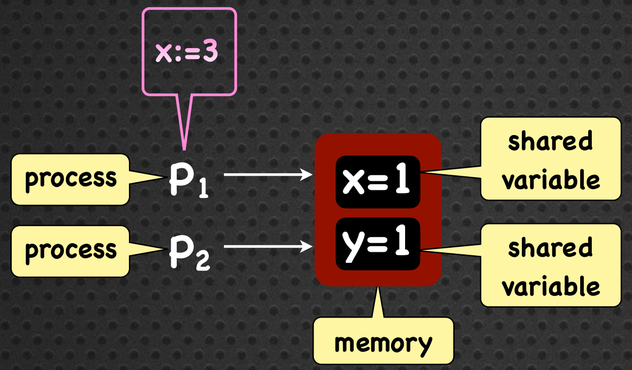
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



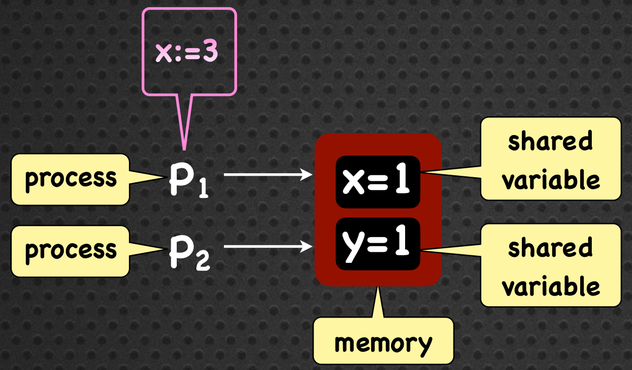
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



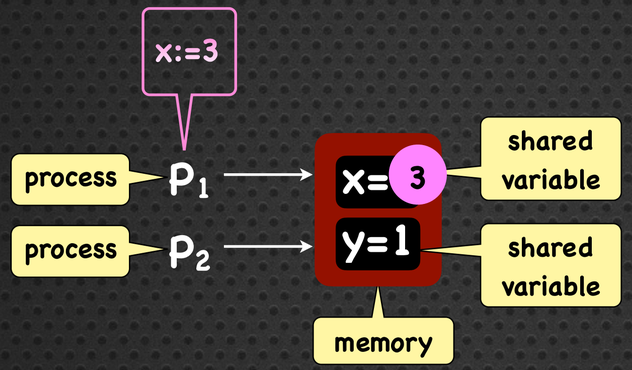
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



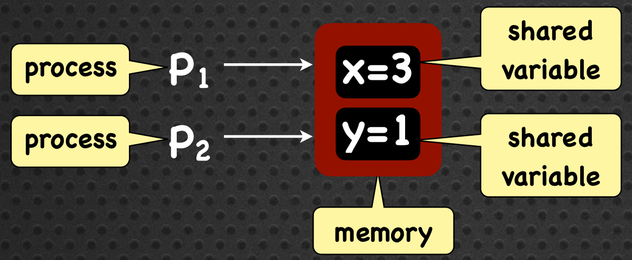
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



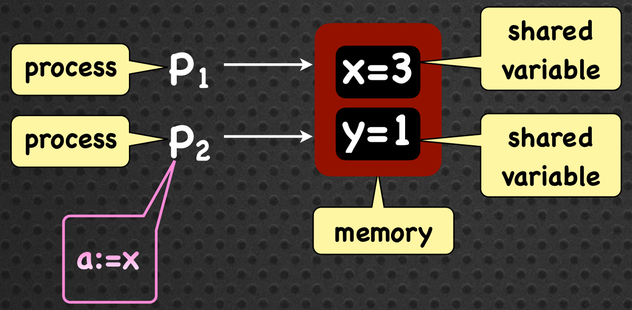
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



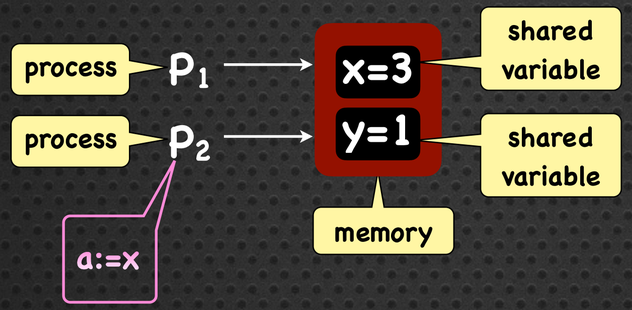
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



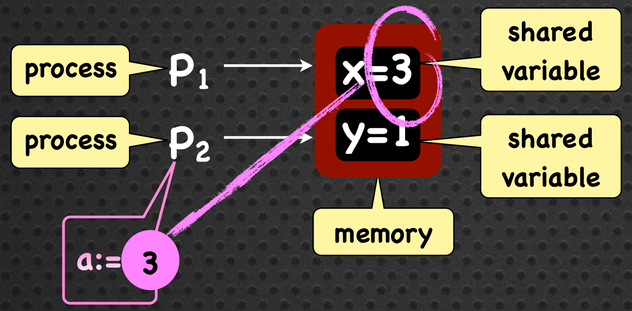
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



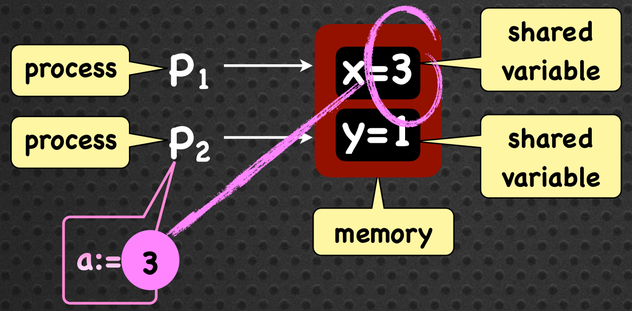
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



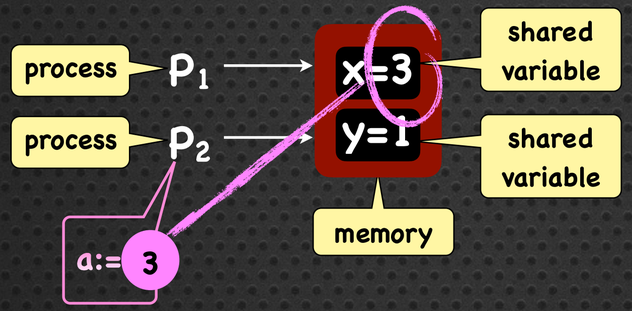
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory



Sequential Consistency (SC)

\+ simple & intuitive

\- expensive

WALLPAPERSWIDE.COM

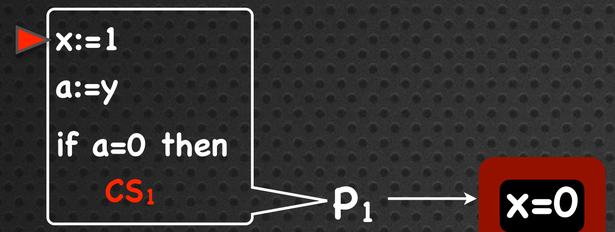
Shared Variable Concurrency

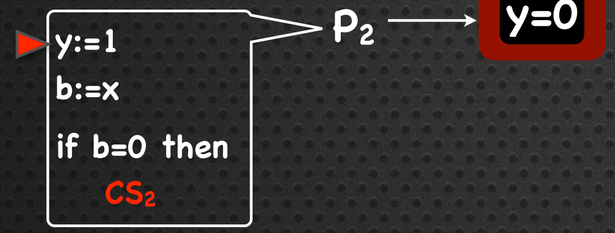
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P_1

P_2

x=0

y=0

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

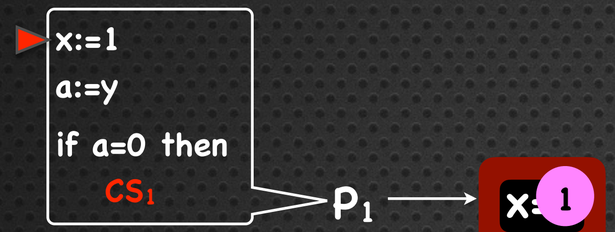
Shared Variable Concurrency

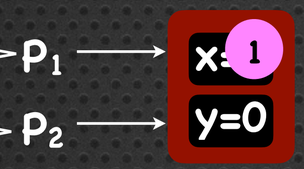
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

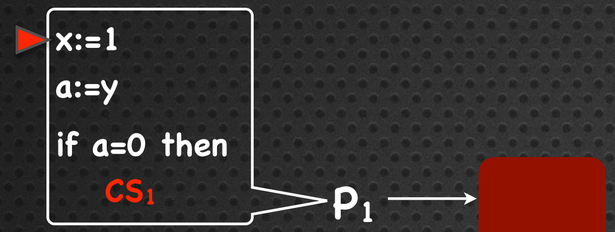
Shared Variable Concurrency

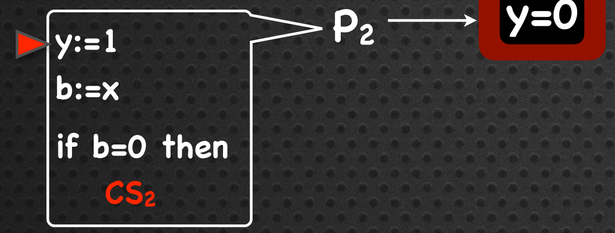
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1

P2

y=0

not (CS1 and CS2)

assertion

WALLPAPERSWIDE.COM

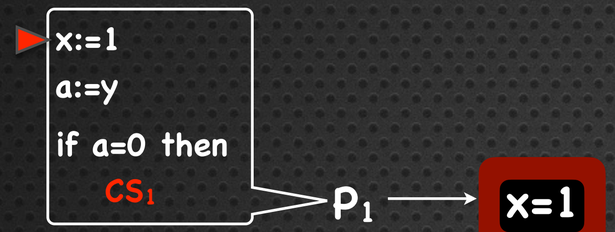
Shared Variable Concurrency

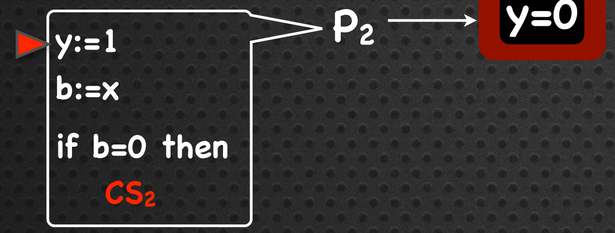
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=0

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

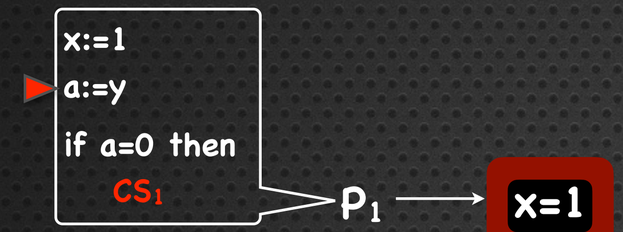
Shared Variable Concurrency

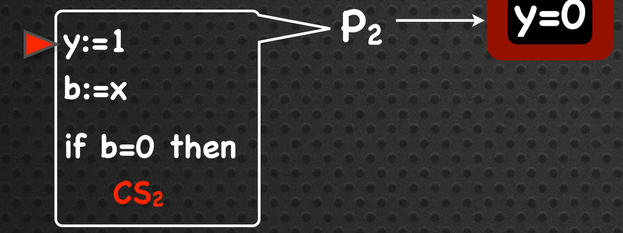
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P_1

P_2

x=1
y=0

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

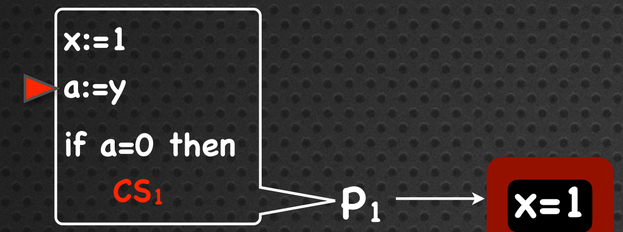
Shared Variable Concurrency

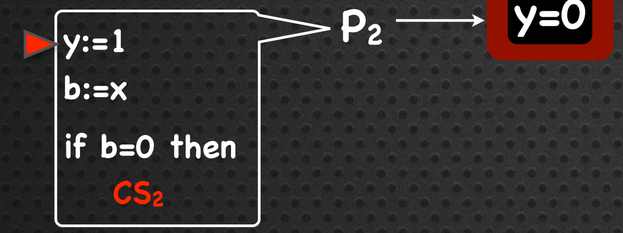
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P_1

P_2

x=1
y=0

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

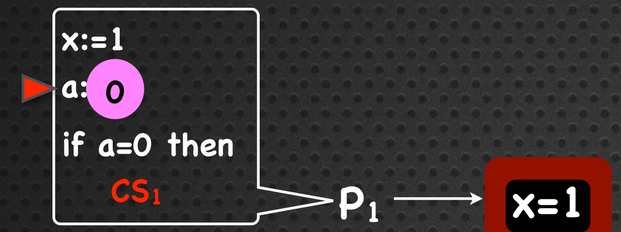
Shared Variable Concurrency

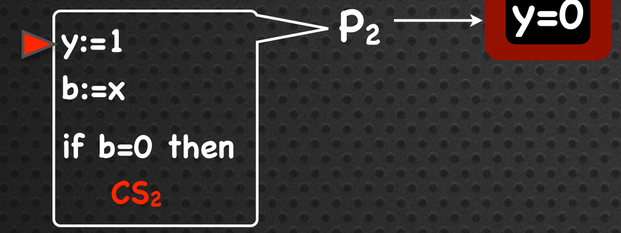
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P_1

P_2

x=1
y=0

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

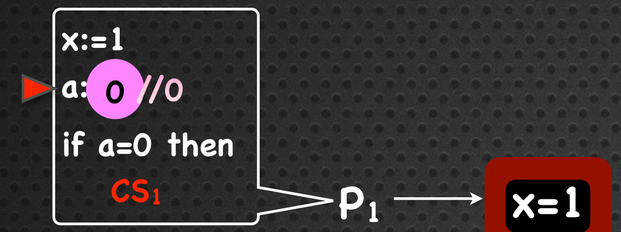
Shared Variable Concurrency

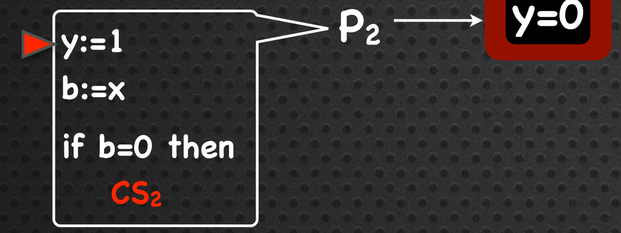
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P_1

P_2

x=1
y=0

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

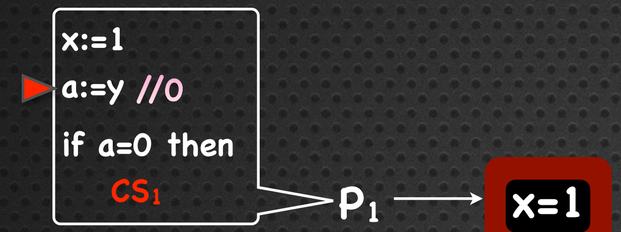
Shared Variable Concurrency

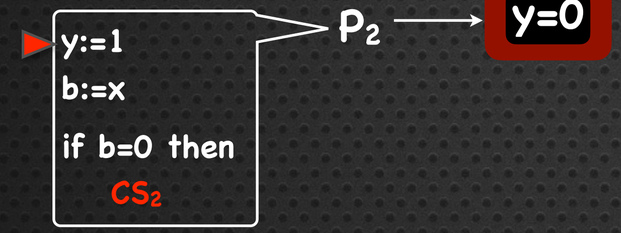
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=0

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

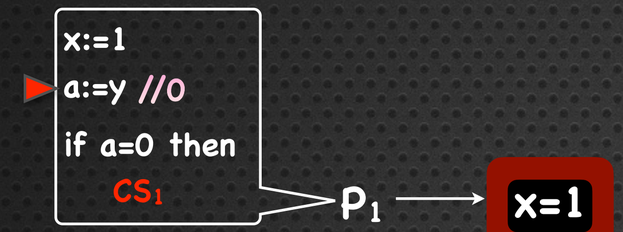
Shared Variable Concurrency

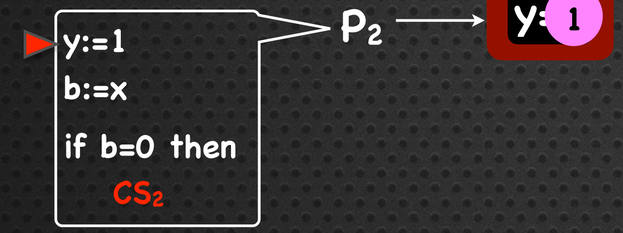
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

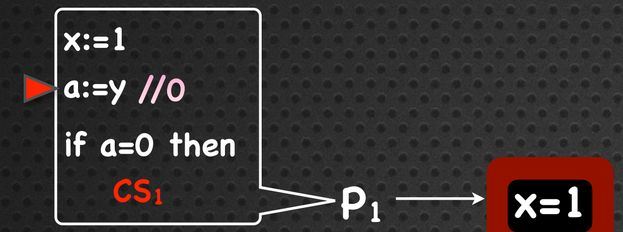
Shared Variable Concurrency

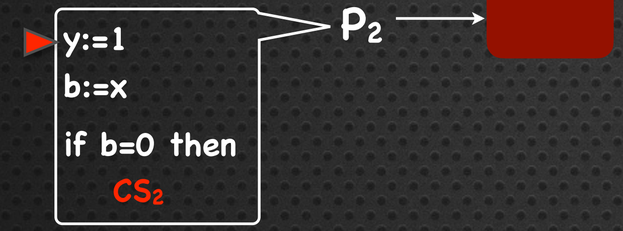
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 →

not (CS1 and CS2)

assertion

WALLPAPERSWIDE.COM

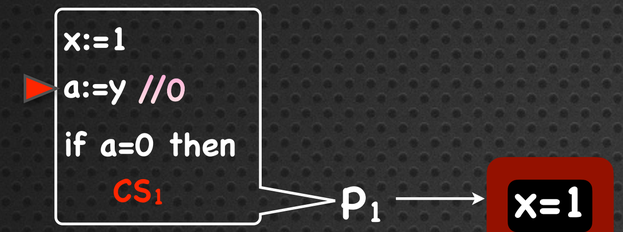
Shared Variable Concurrency

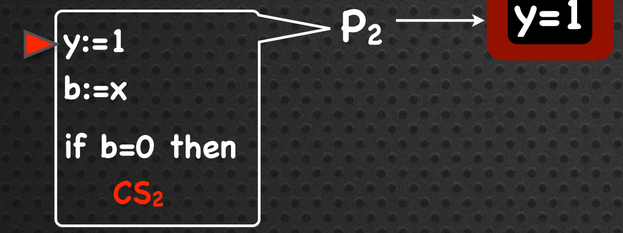
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

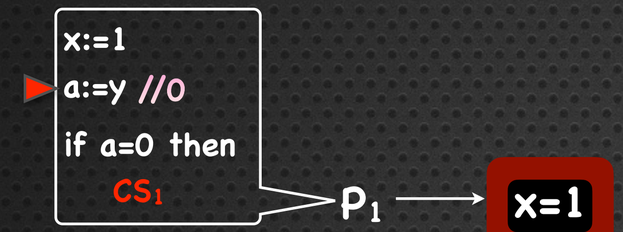
Shared Variable Concurrency

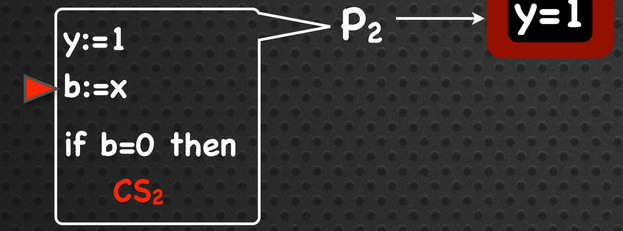
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

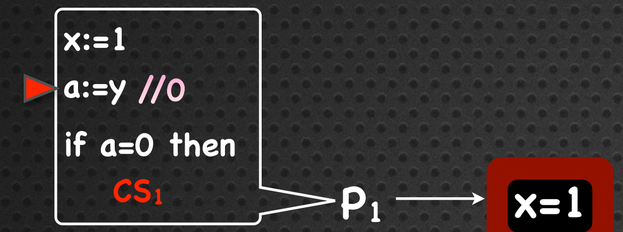
Shared Variable Concurrency

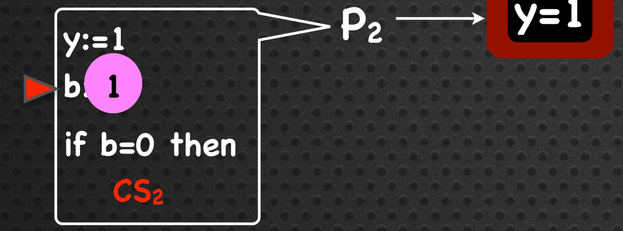
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

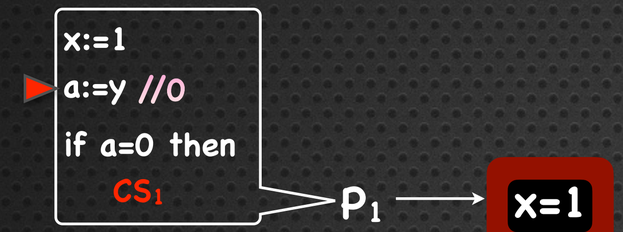
Shared Variable Concurrency

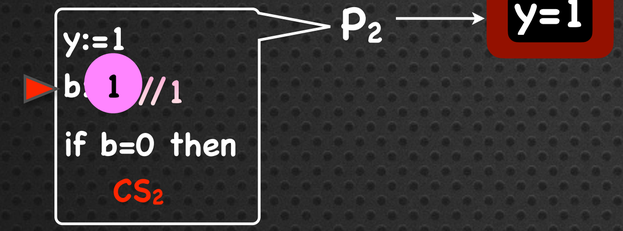
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P_1

P_2

x=1

y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

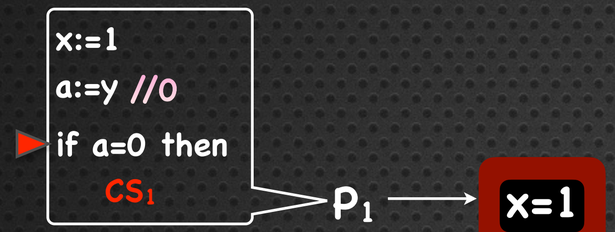
Shared Variable Concurrency

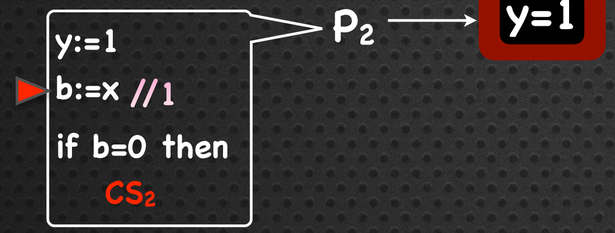
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P_1

P_2

x=1

y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

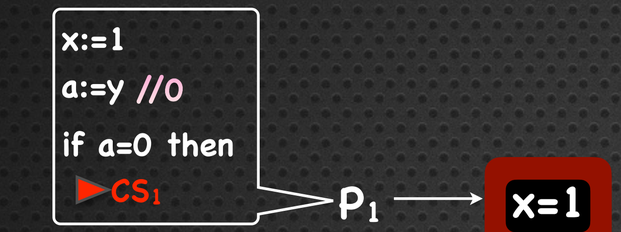
Shared Variable Concurrency

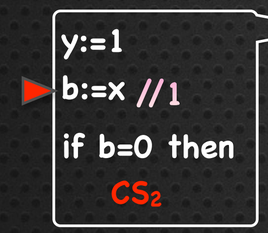
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

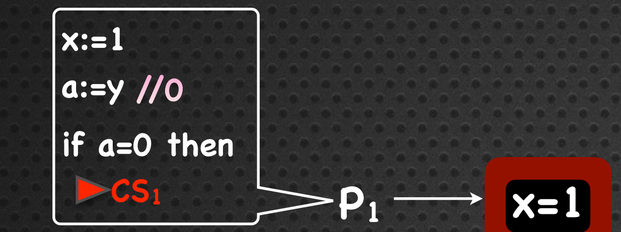
Shared Variable Concurrency

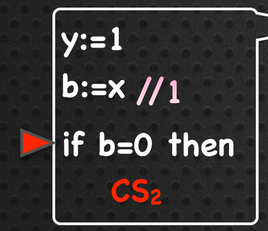
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

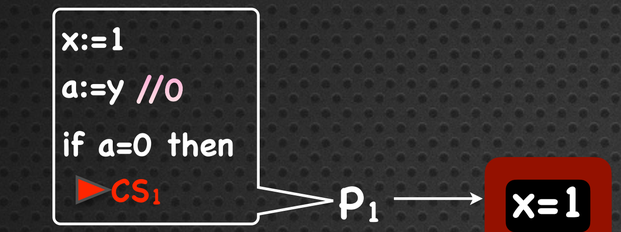
Shared Variable Concurrency

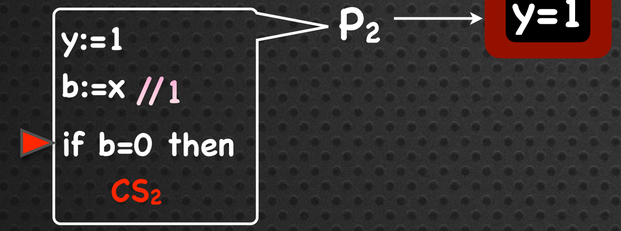
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)

assertion

WALLPAPERSWIDE.COM

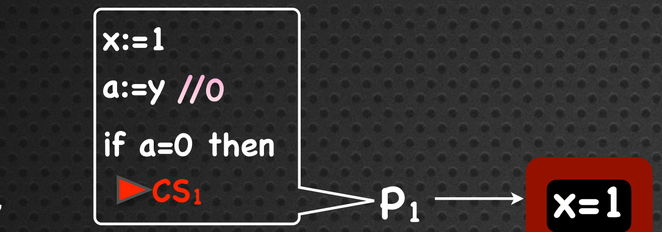
Shared Variable Concurrency

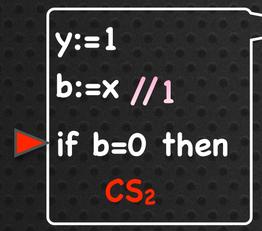
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)



assertion

WALLPAPERSWIDE.COM

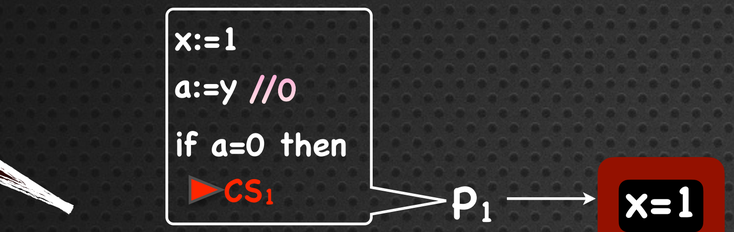
Shared Variable Concurrency

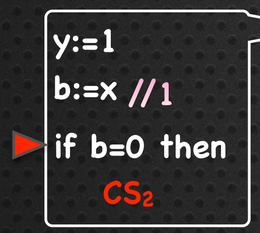
Sequential Consistency (SC)

atomic operations

- atomic writes
- read from memory

Dekker Protocol





P1 → x=1

P2 → y=1

not (CS_1 and CS_2)

assertion

under SC

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

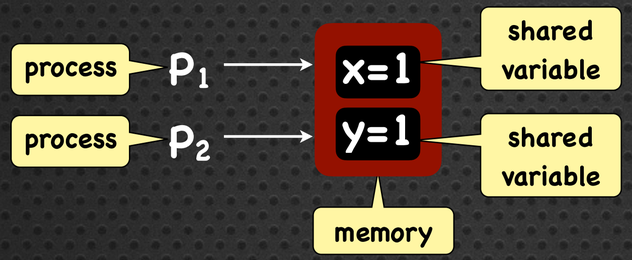
- non-atomic writes
- read locally or from memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

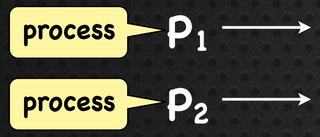


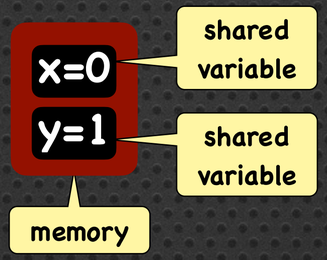
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory



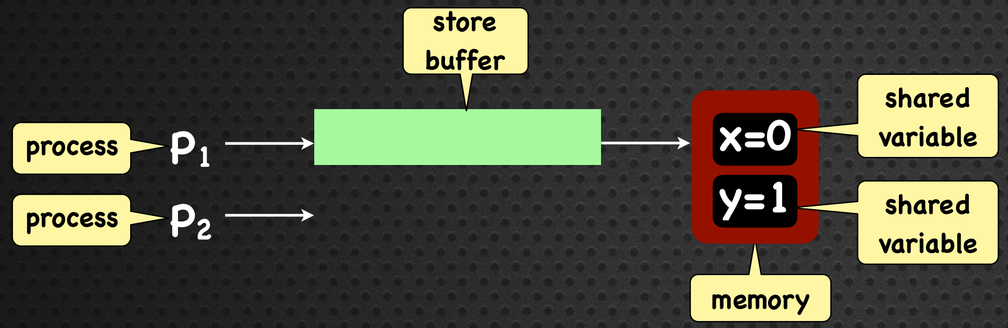


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

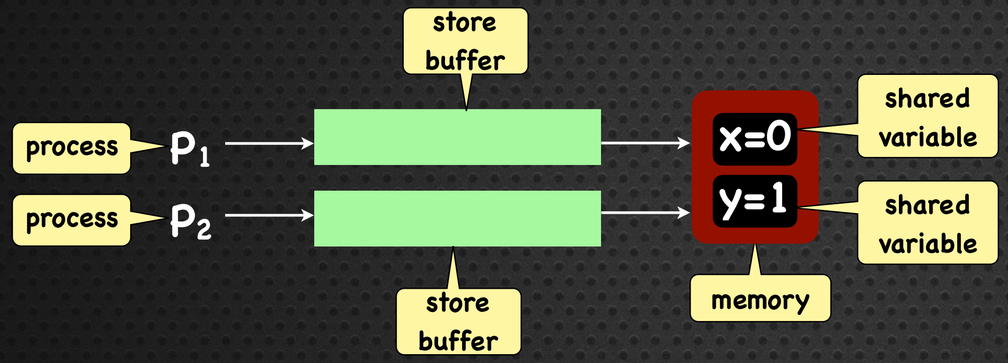


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

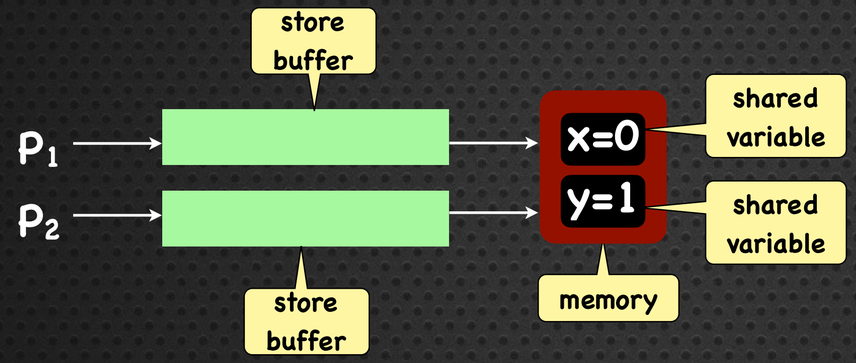


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

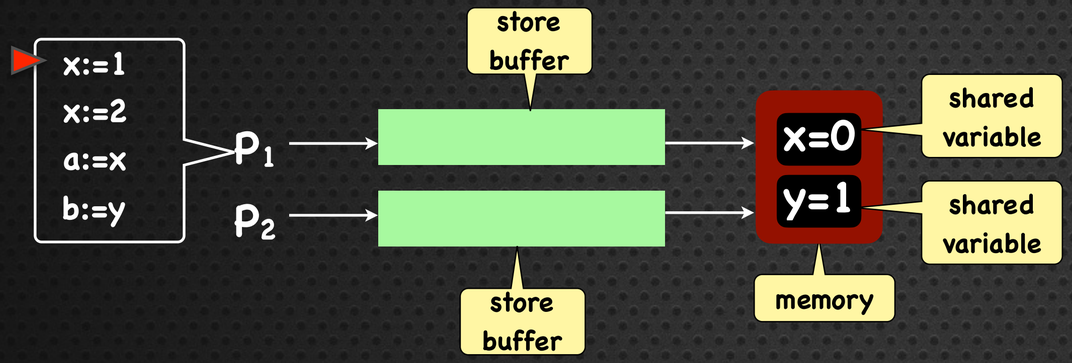


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

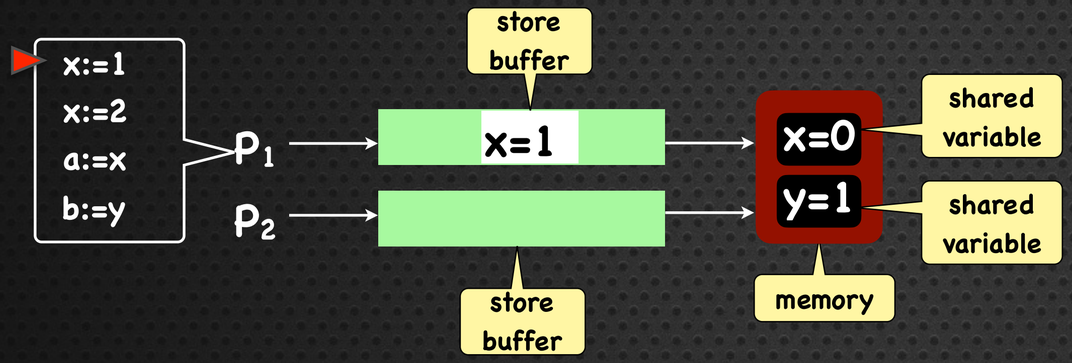


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

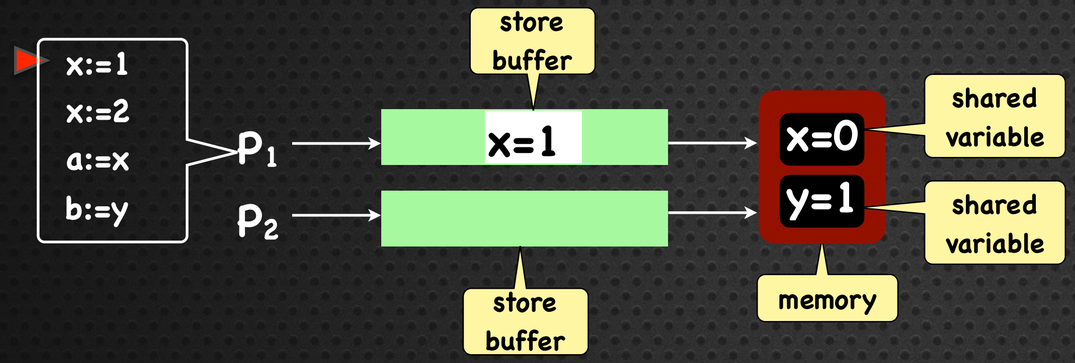


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

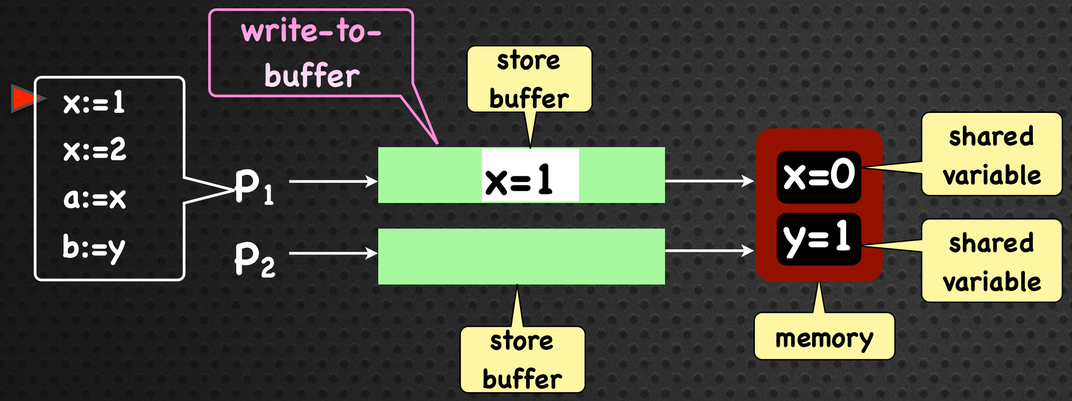


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

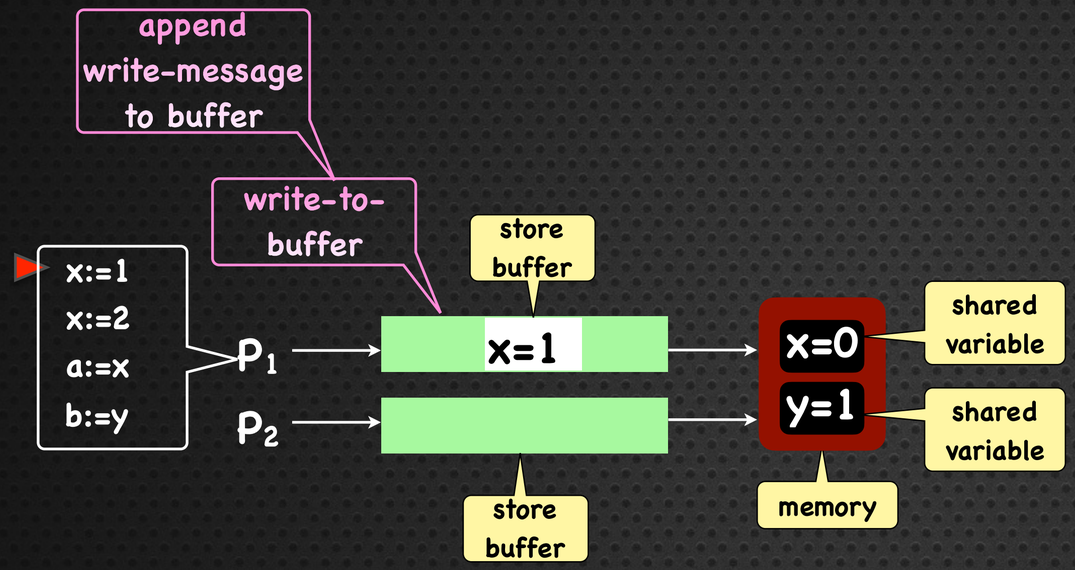


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

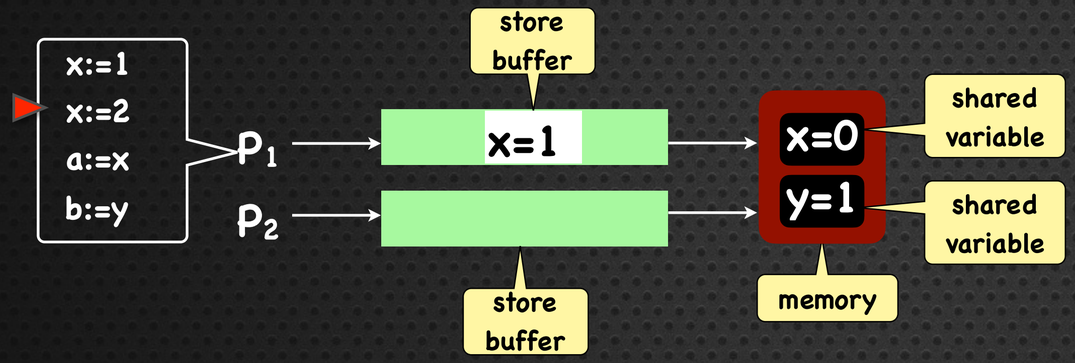


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

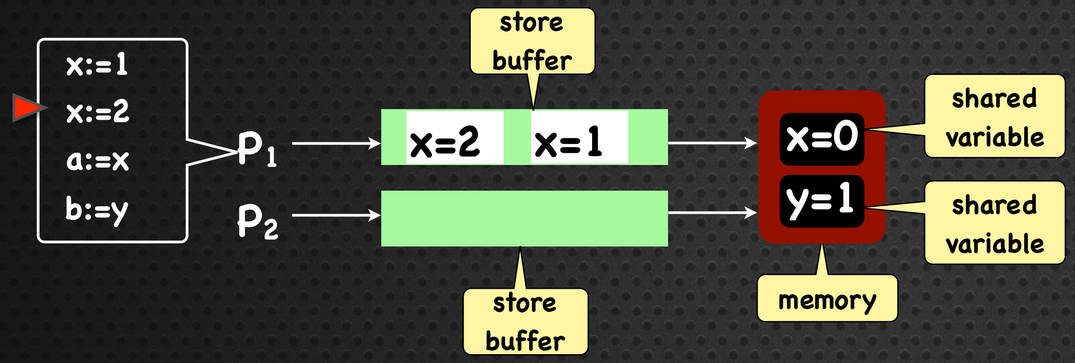


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

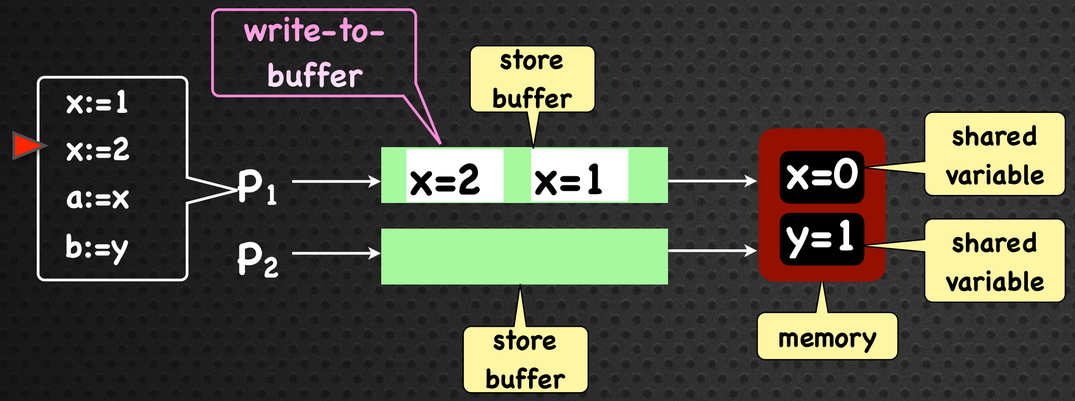


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

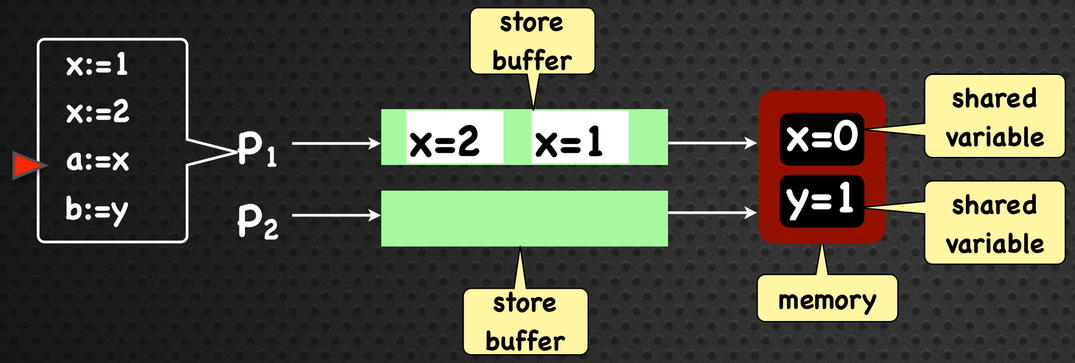
- non-atomic writes
- read locally or from memory



Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

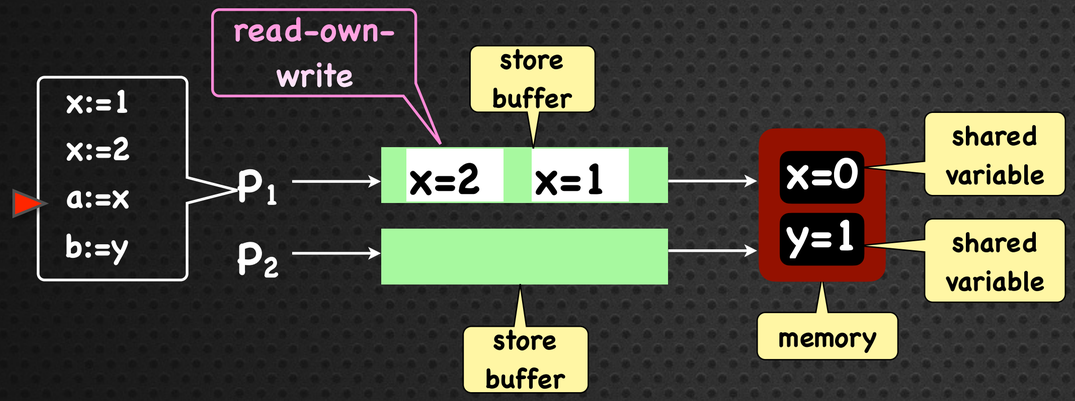


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

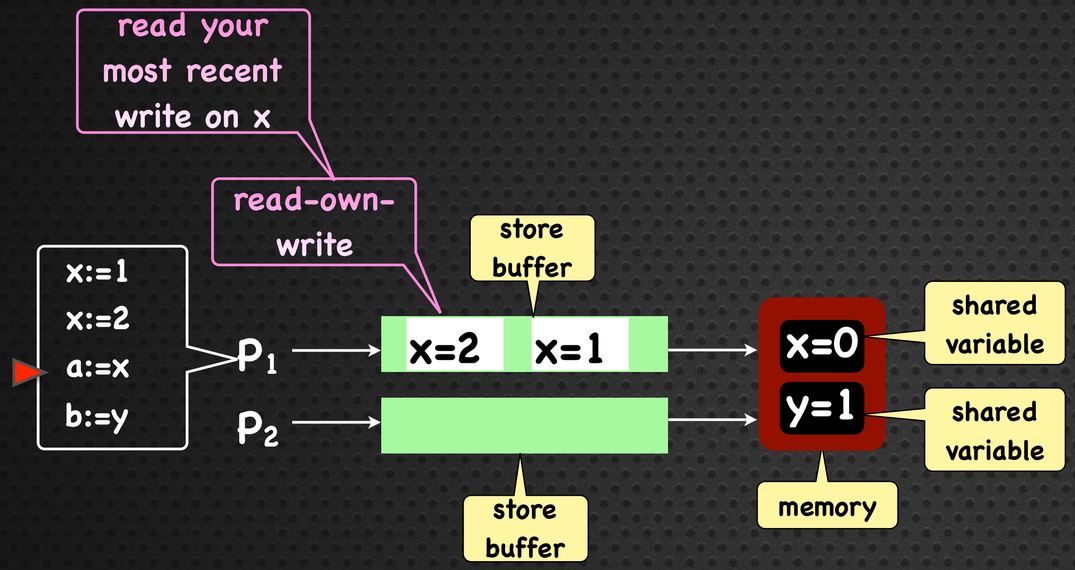


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

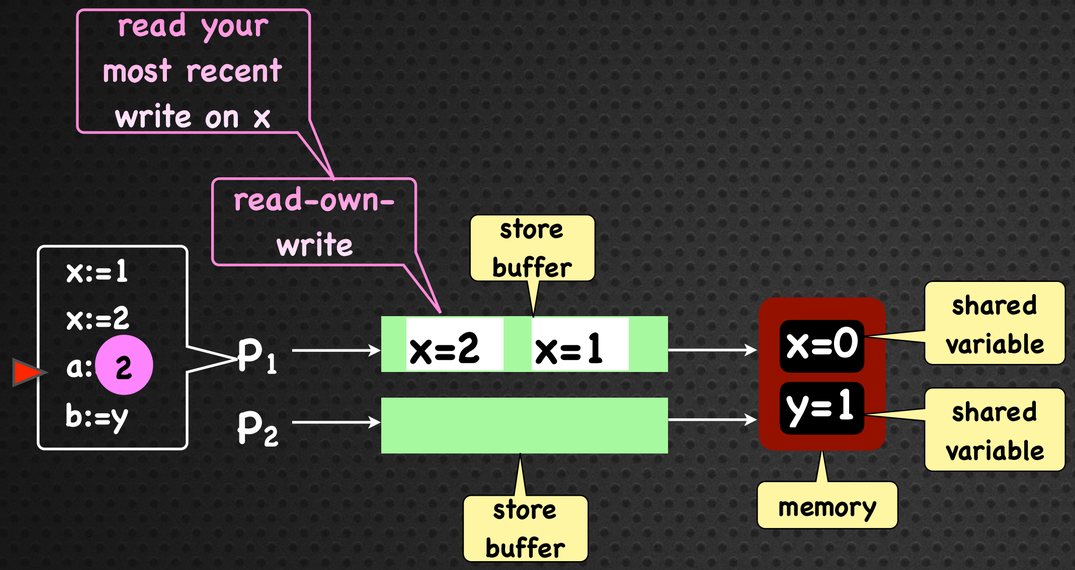


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

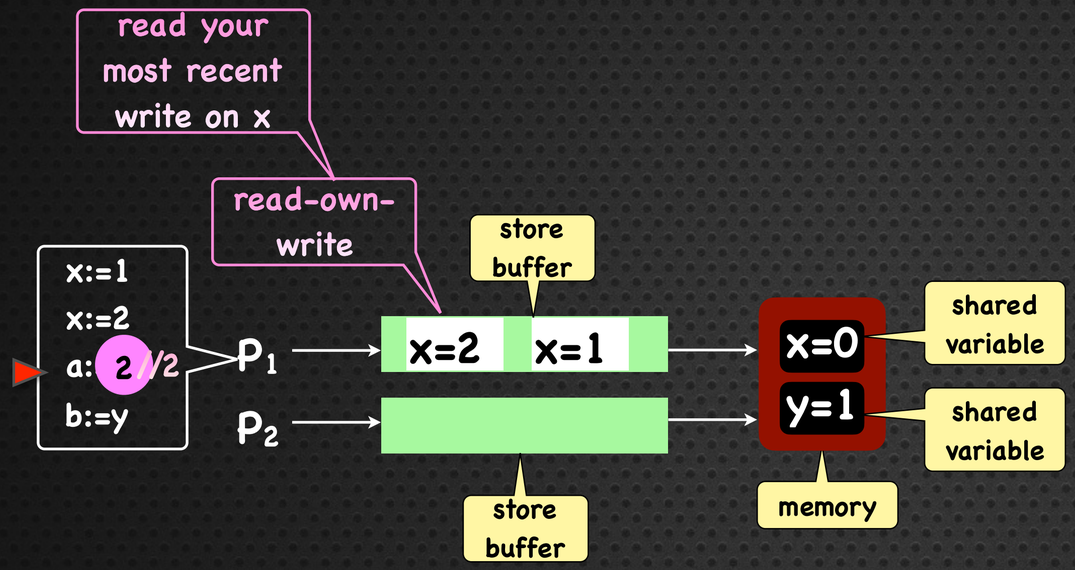
- non-atomic writes
- read locally or from memory



Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

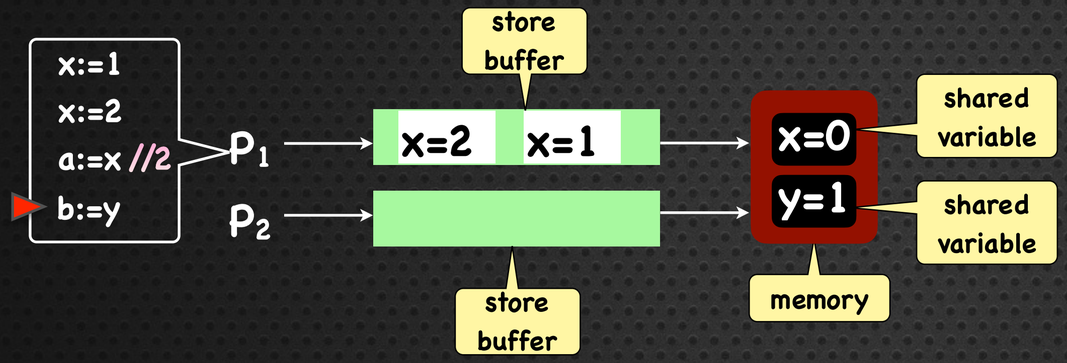


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

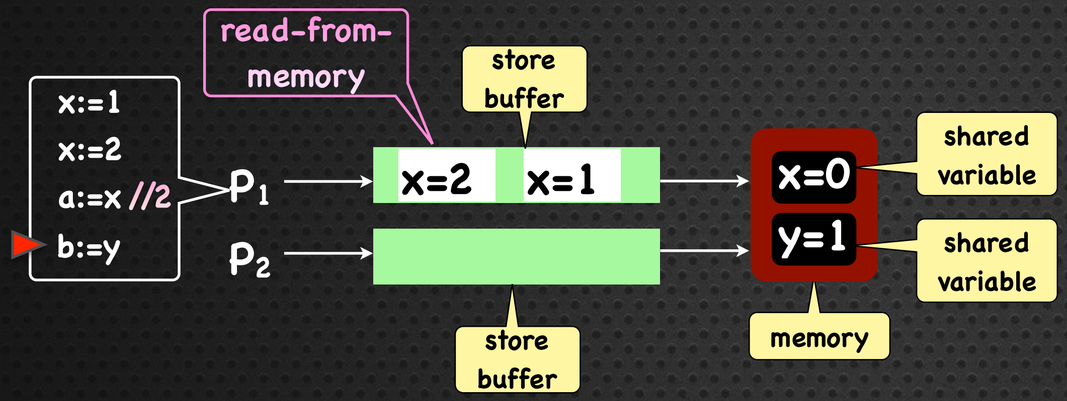


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

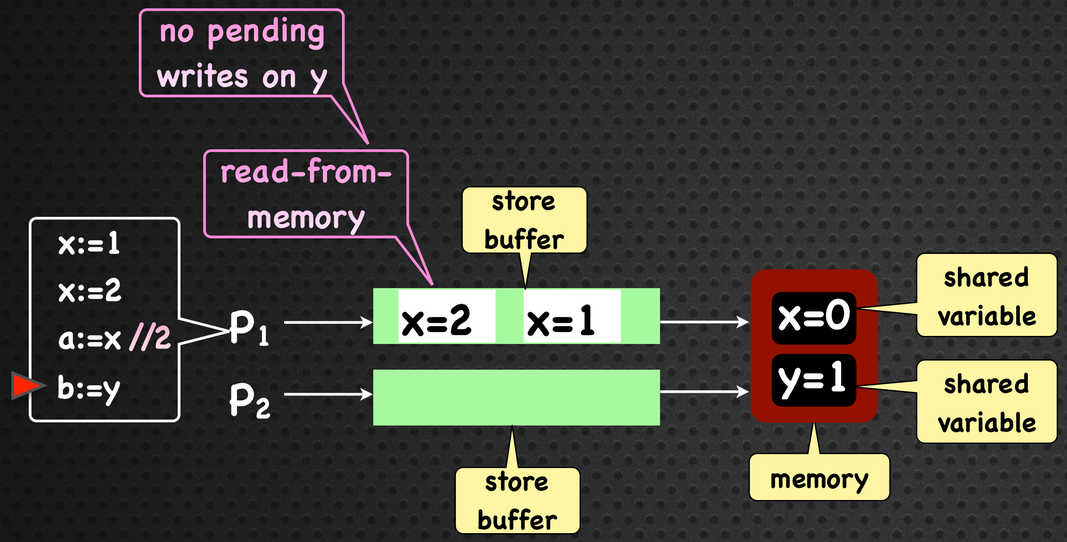


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

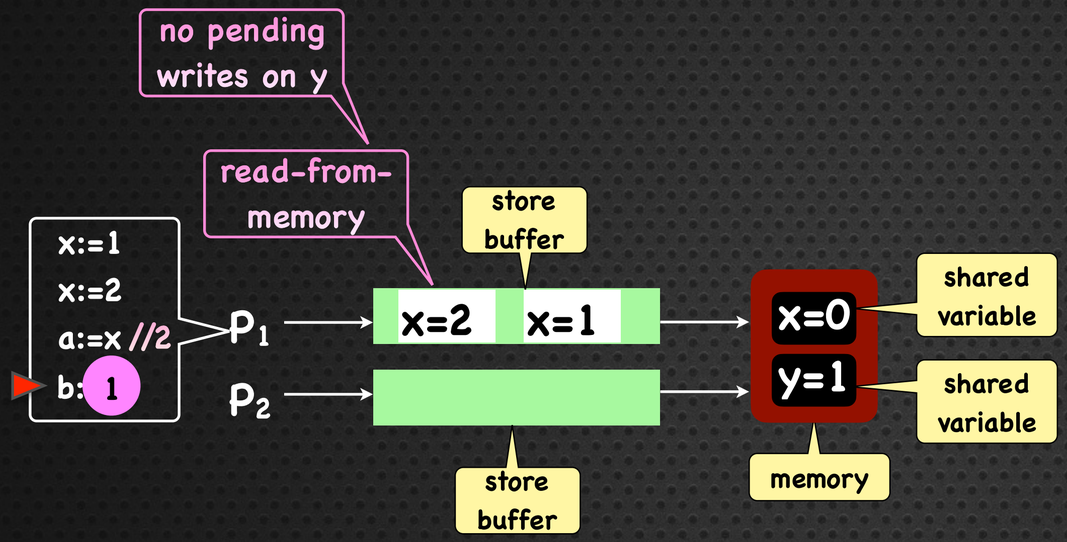


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

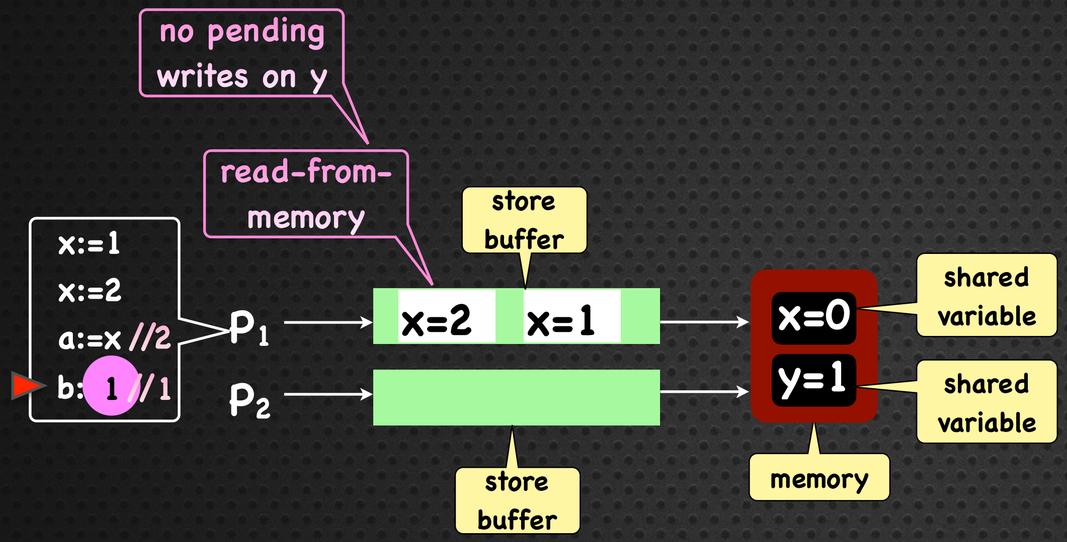


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

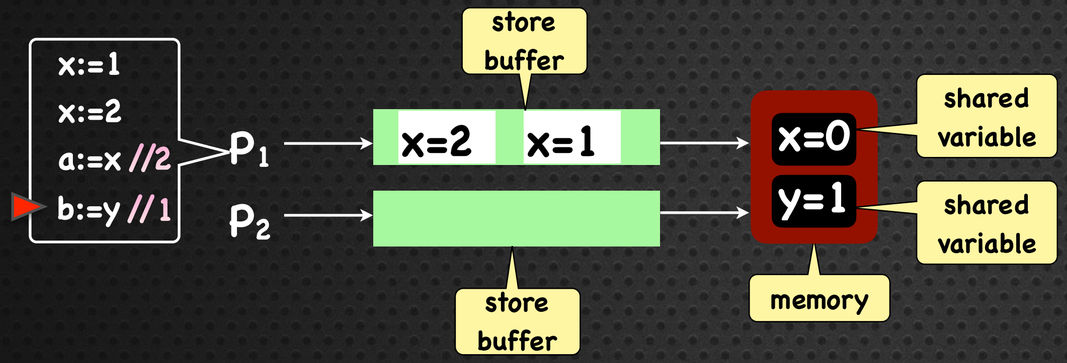
- non-atomic writes
- read locally or from memory



Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

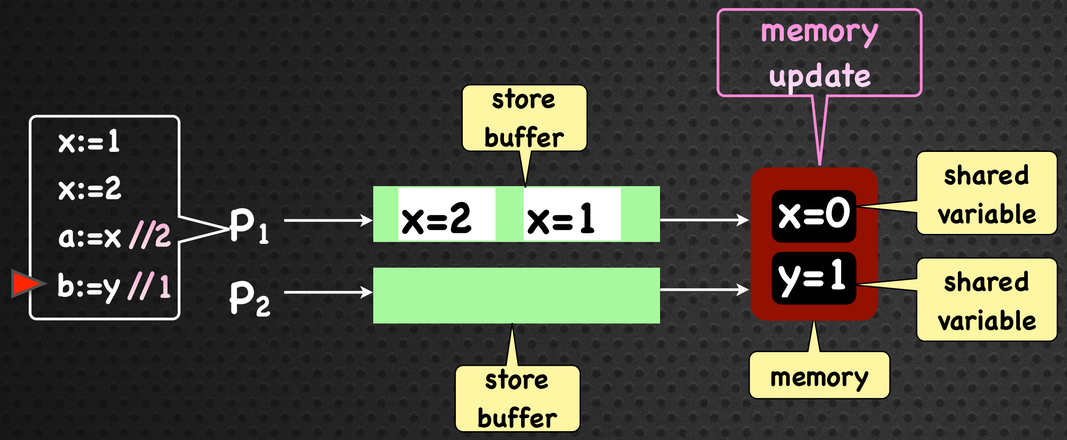


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

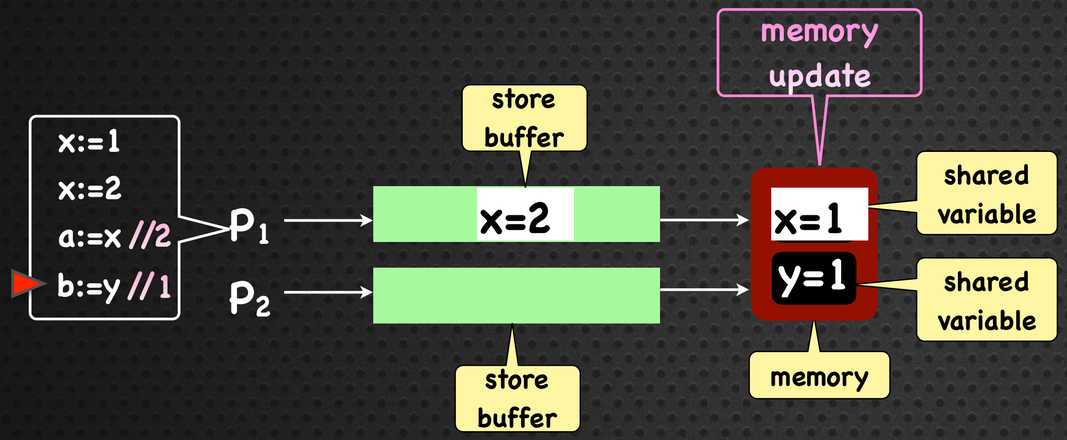


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

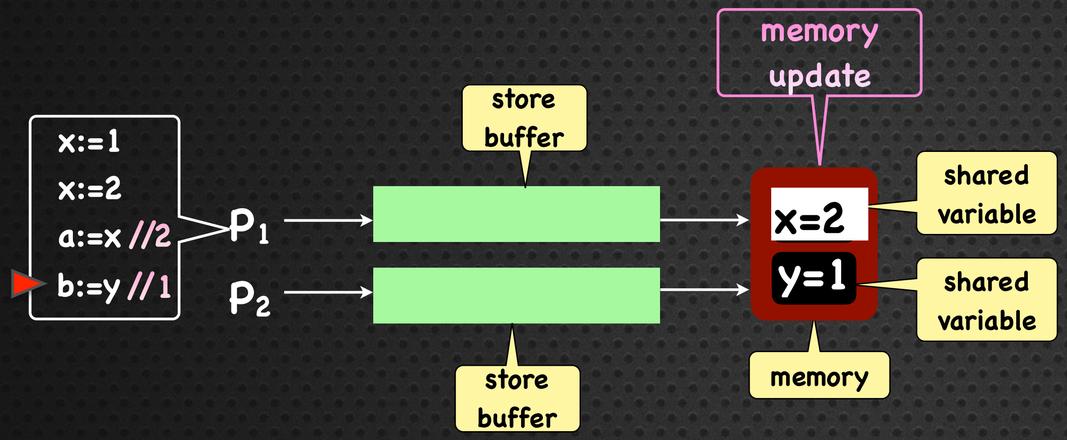


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory



WALLPAPERSWIDE.COM

Shared Variable Concurrency

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

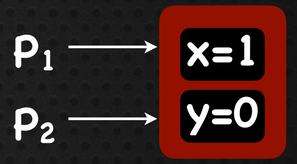
- atomic writes
- read from memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

- atomic writes
- read from memory

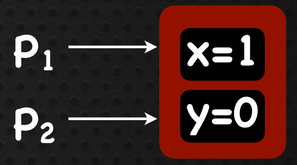


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

- atomic writes
- read from memory



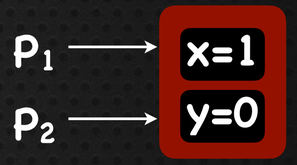
- write-to-memory
- read-from-memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

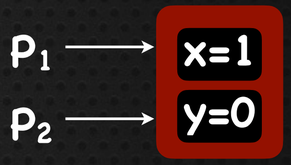


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory



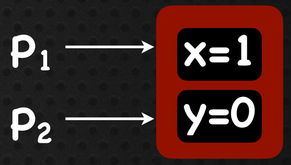
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory



WALLPAPERSWIDE.COM

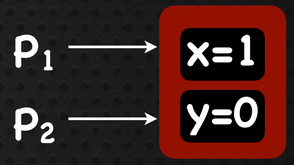
Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

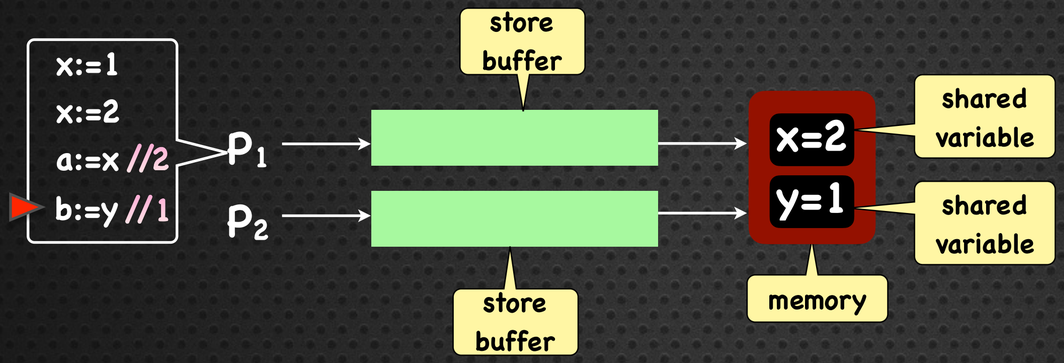


WALLPAPERSWIDE.COM

Shared Variable Concurrency

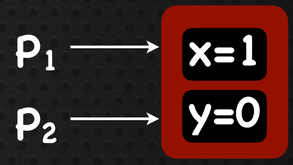
Total Store Order (TSO)

- non-atomic writes
- read locally or from memory



Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

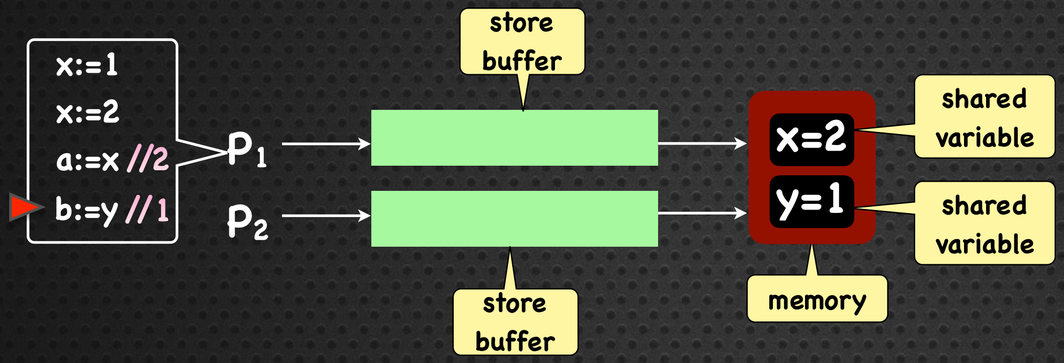
SC

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

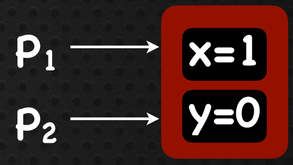
- non-atomic writes
- read locally or from memory



- write-to-buffer
- read-own-write
- read-from-memory
- update

Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

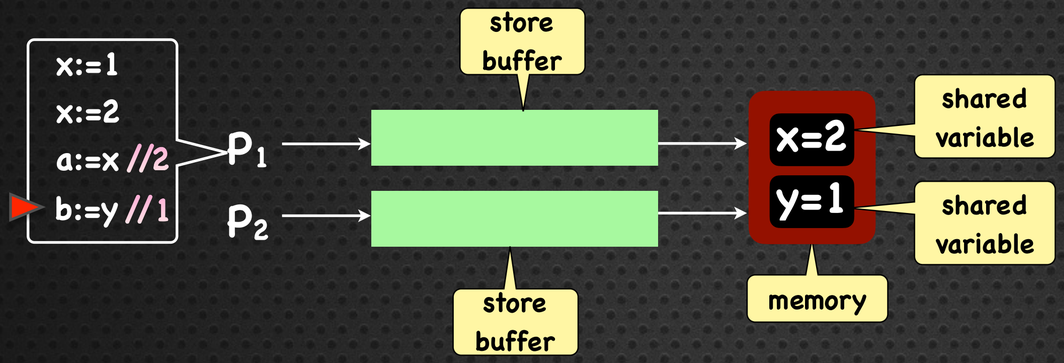
SC

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

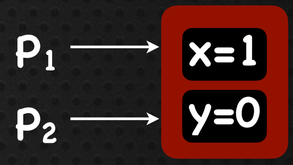


- write-to-buffer
- read-own-write
- read-from-memory
- update

TSO

Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

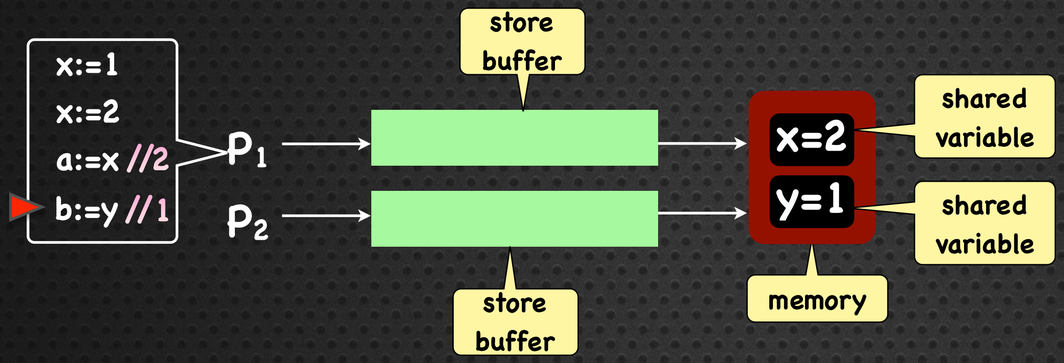
SC

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

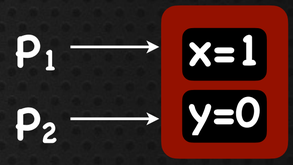


- write-to-buffer
- read-own-write
- read-from-memory
- update

TSO

Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

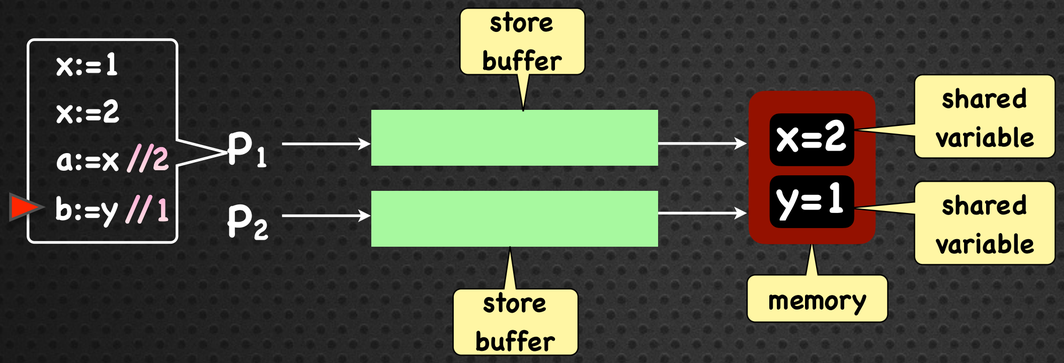
SC

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory



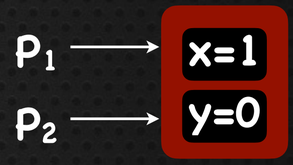
- write-to-buffer
- read-own-write
- read-from-memory
- update

TSO

weak memory model

Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

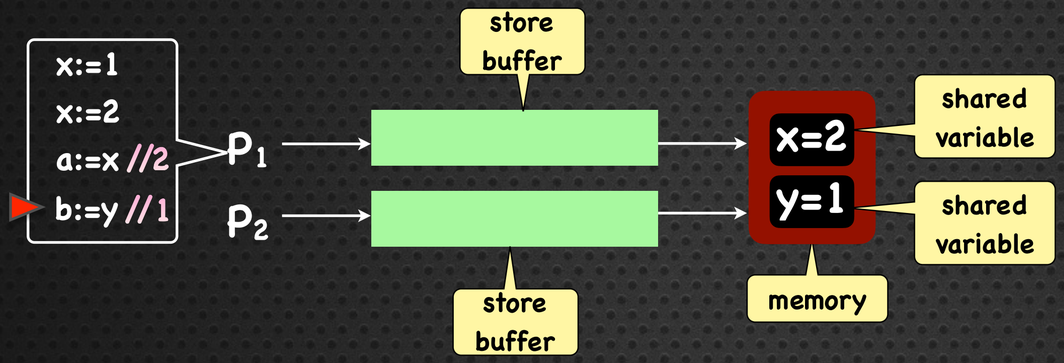
SC

WALLPAPERSWIDE.COM

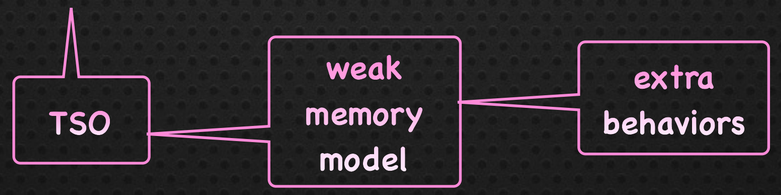
Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

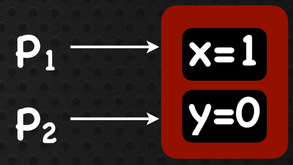


- write-to-buffer
- read-own-write
- read-from-memory
- update



Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

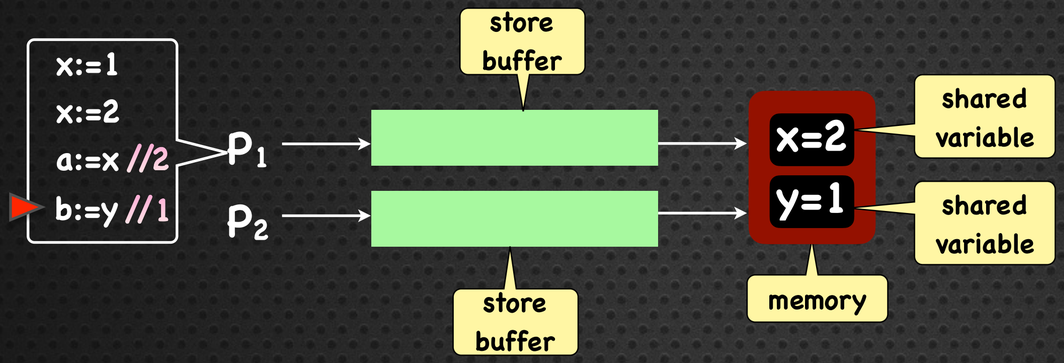


WALLPAPERSWIDE.COM

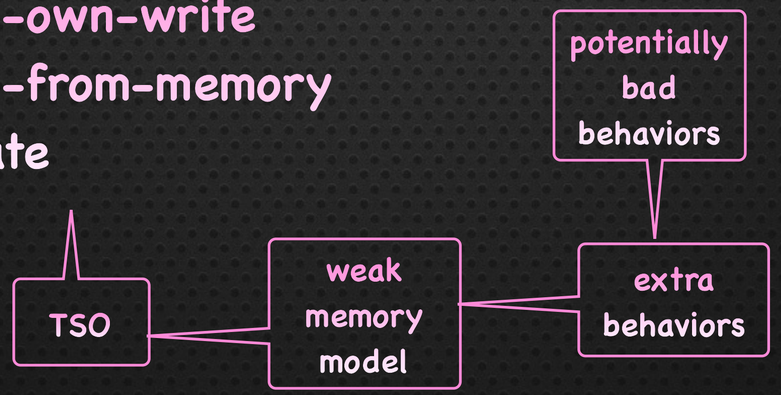
Shared Variable Concurrency

Total Store Order (TSO)

- non-atomic writes
- read locally or from memory

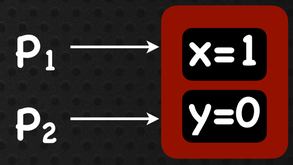


- write-to-buffer
- read-own-write
- read-from-memory
- update



Sequential Consistency (SC)

- atomic writes
- read from memory



- write-to-memory
- read-from-memory

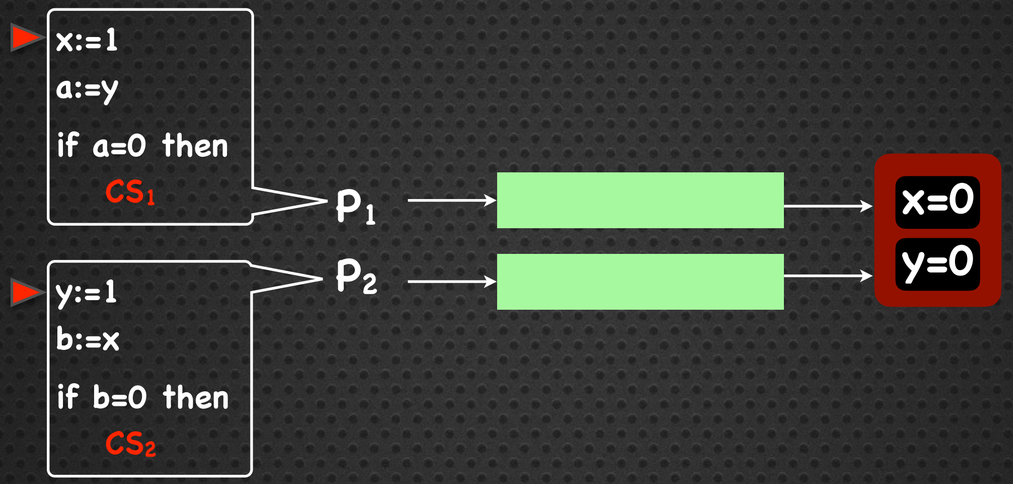


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS)

assertion

potentially bad behaviors

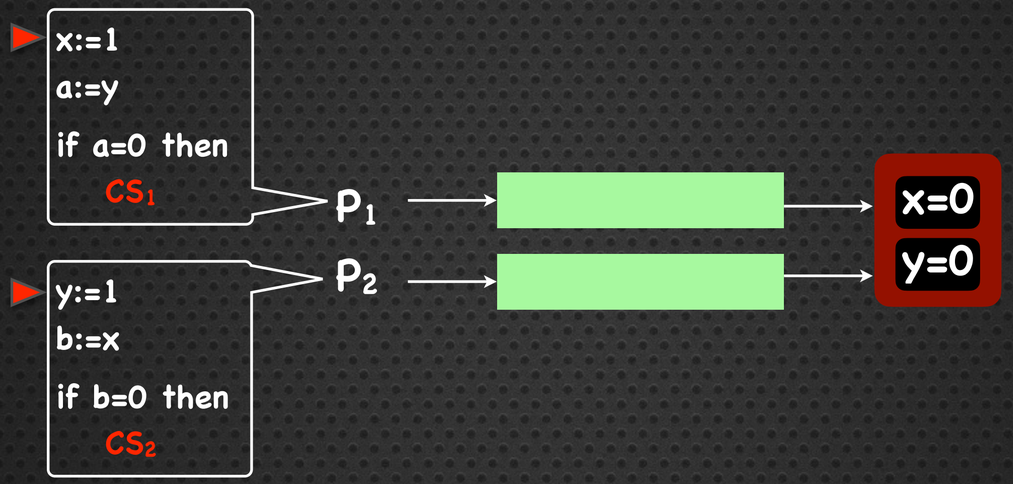
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

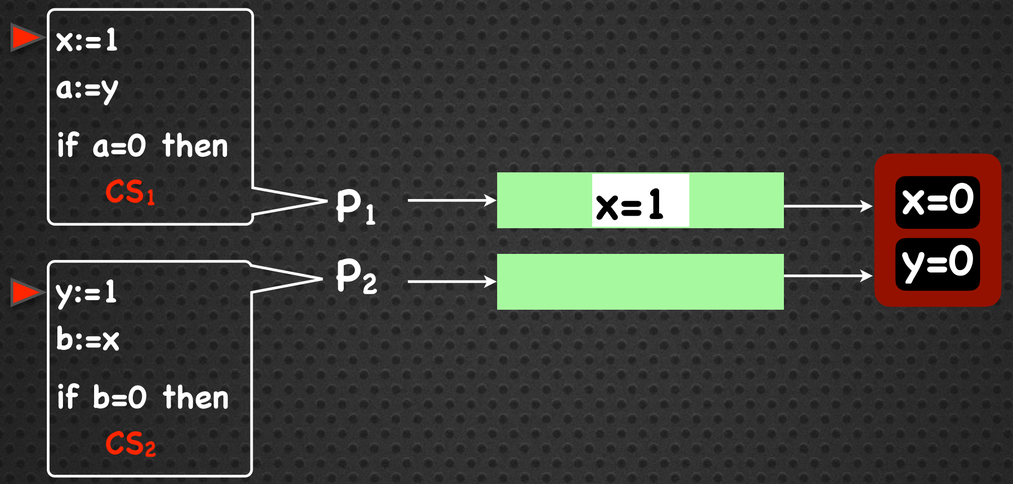
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

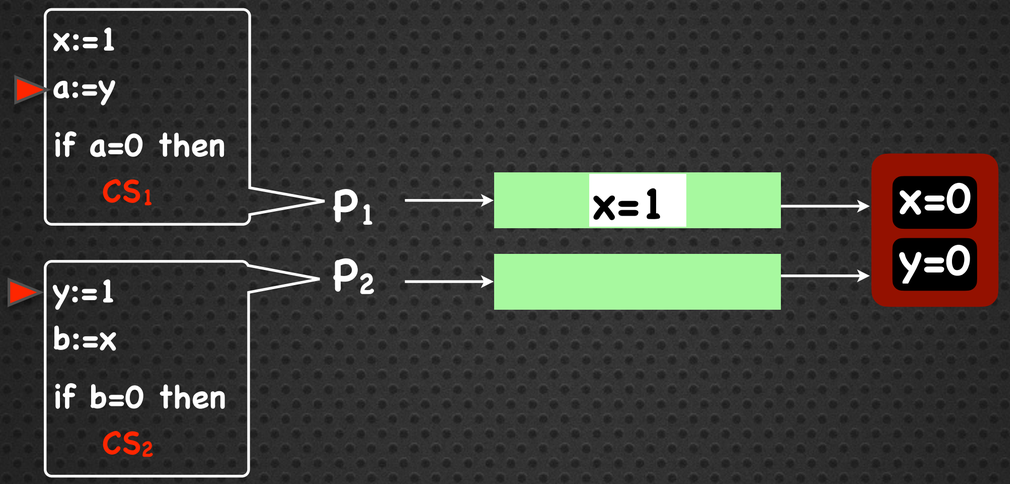
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

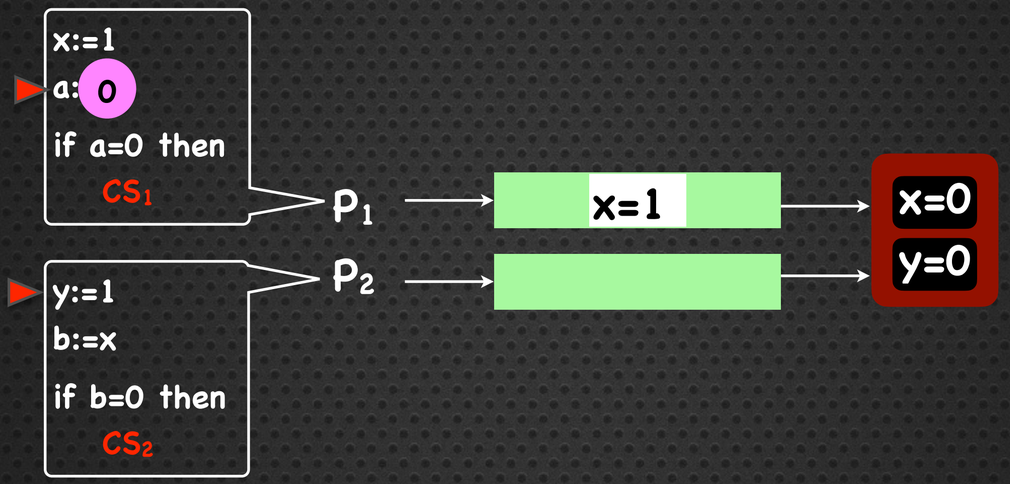
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

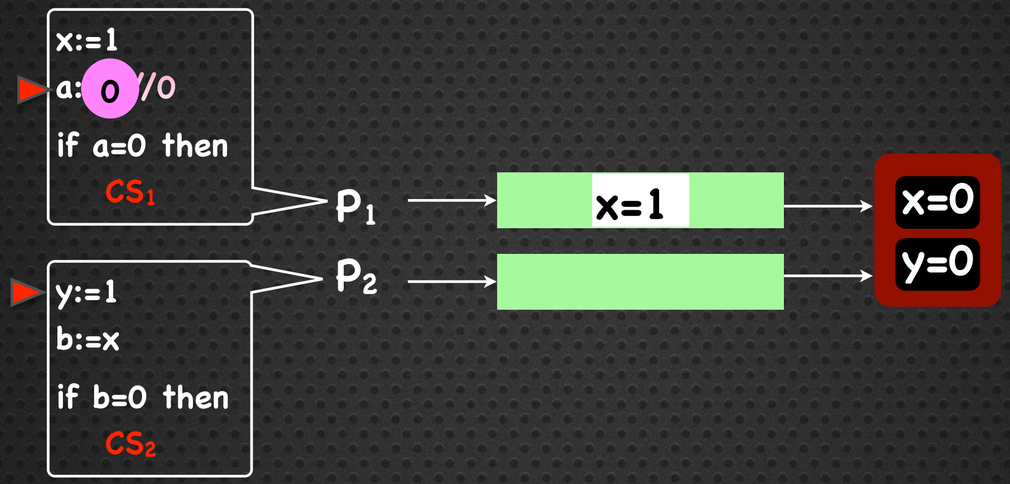
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

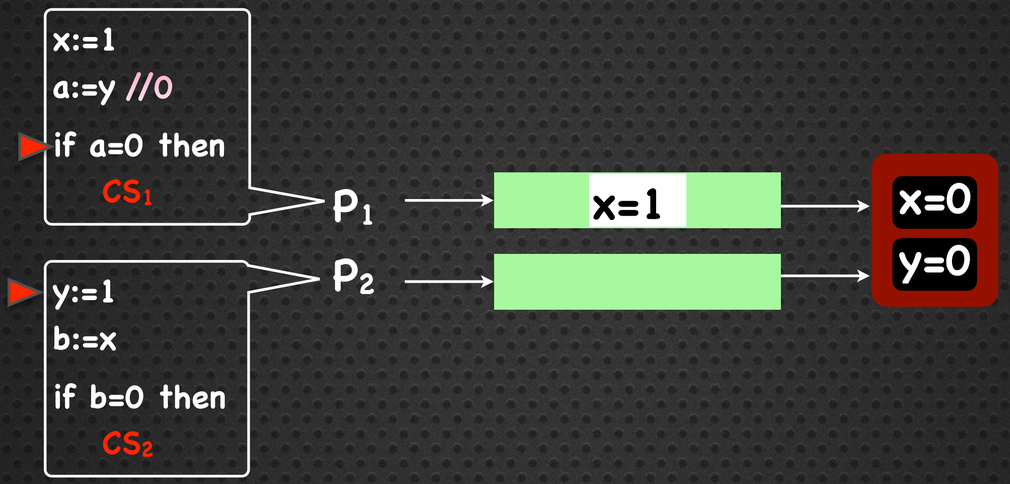
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

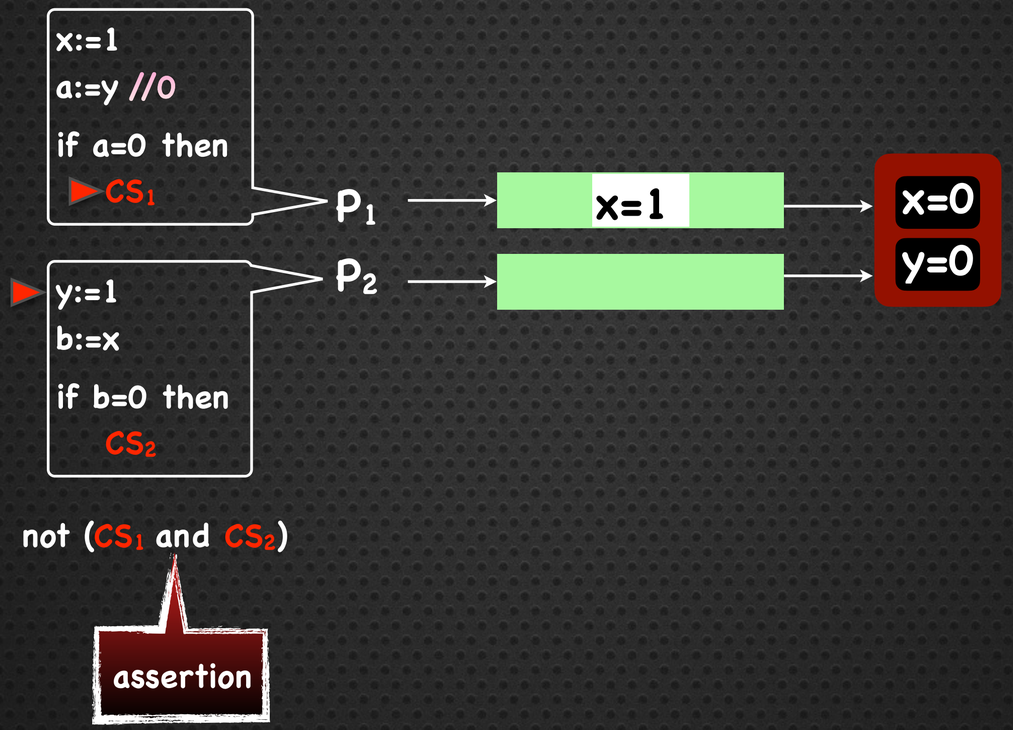
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

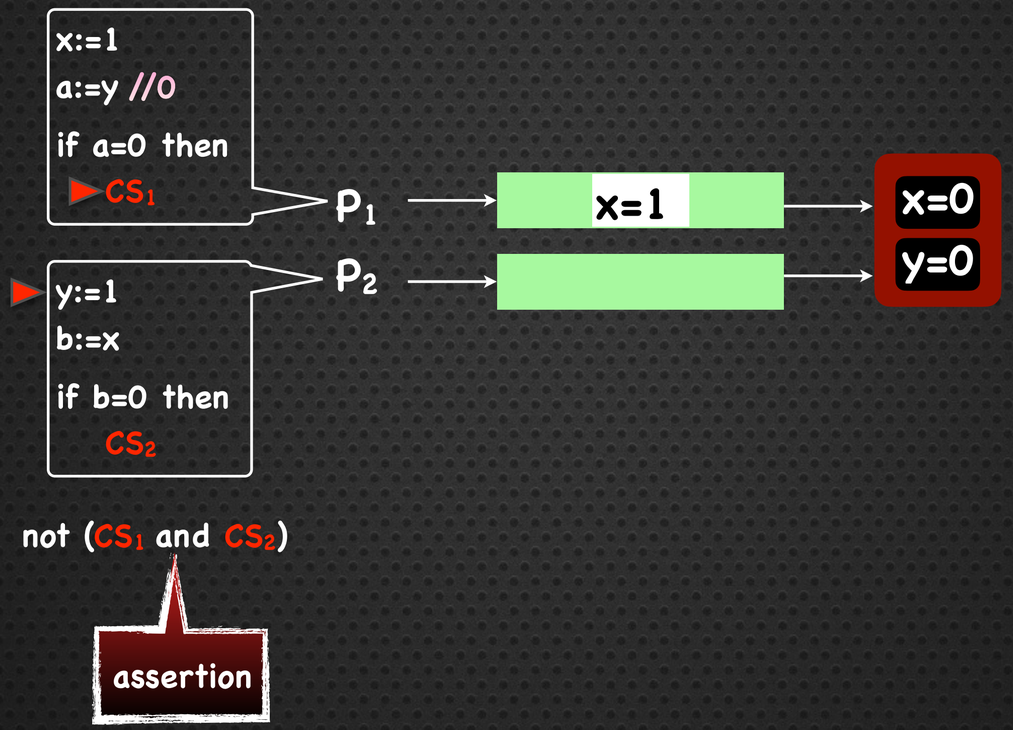
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

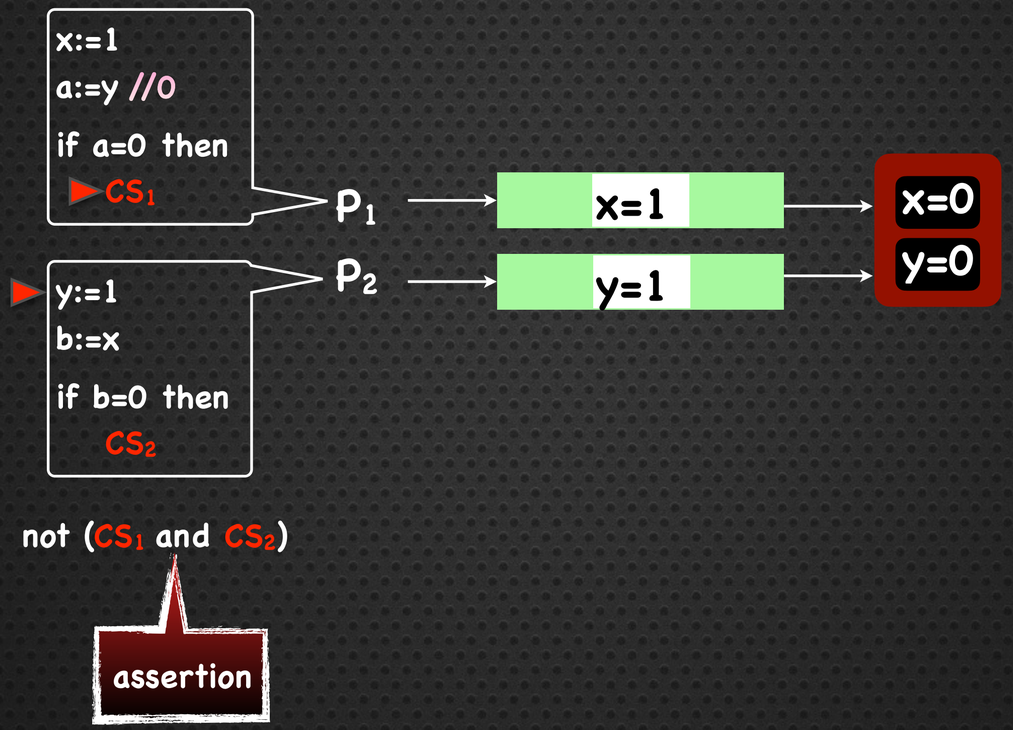
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

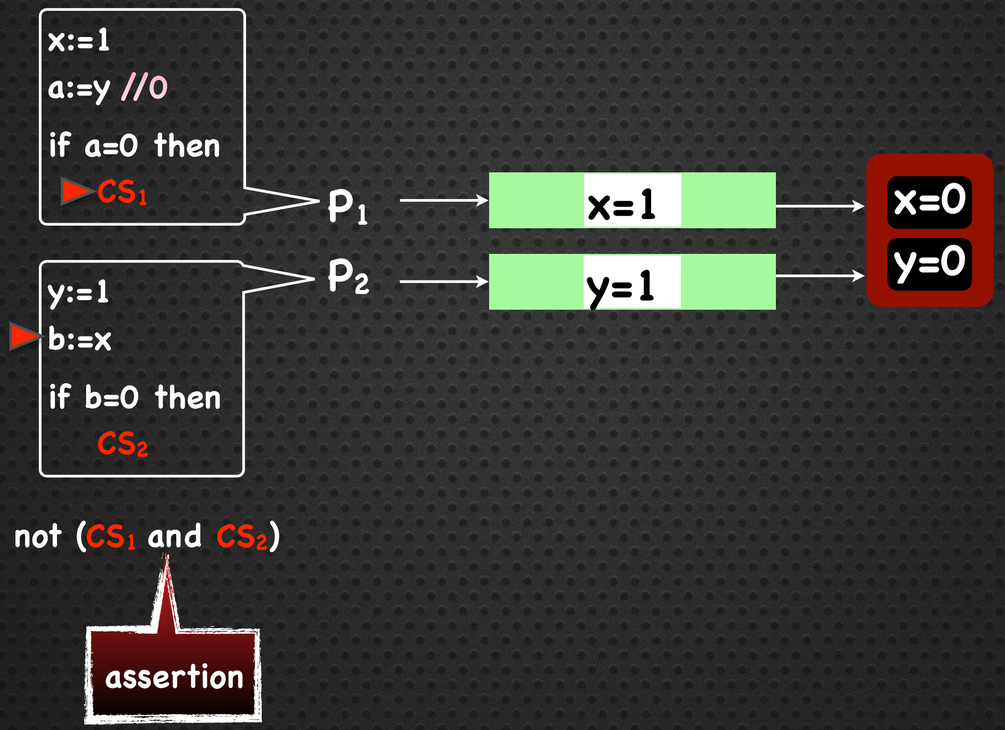
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

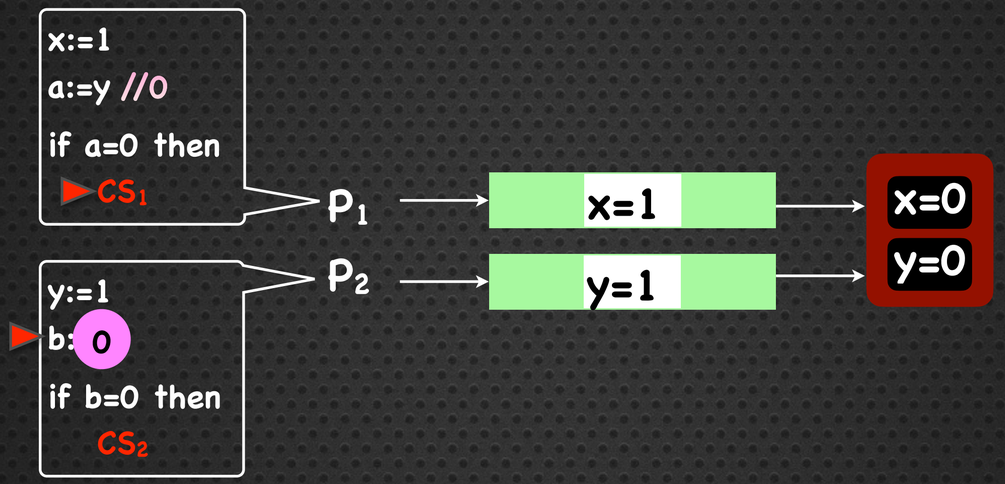
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

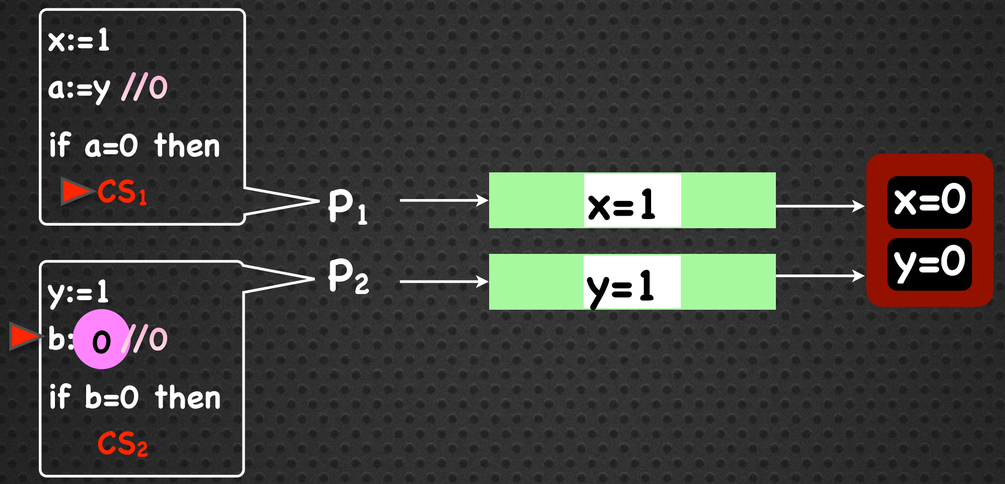
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

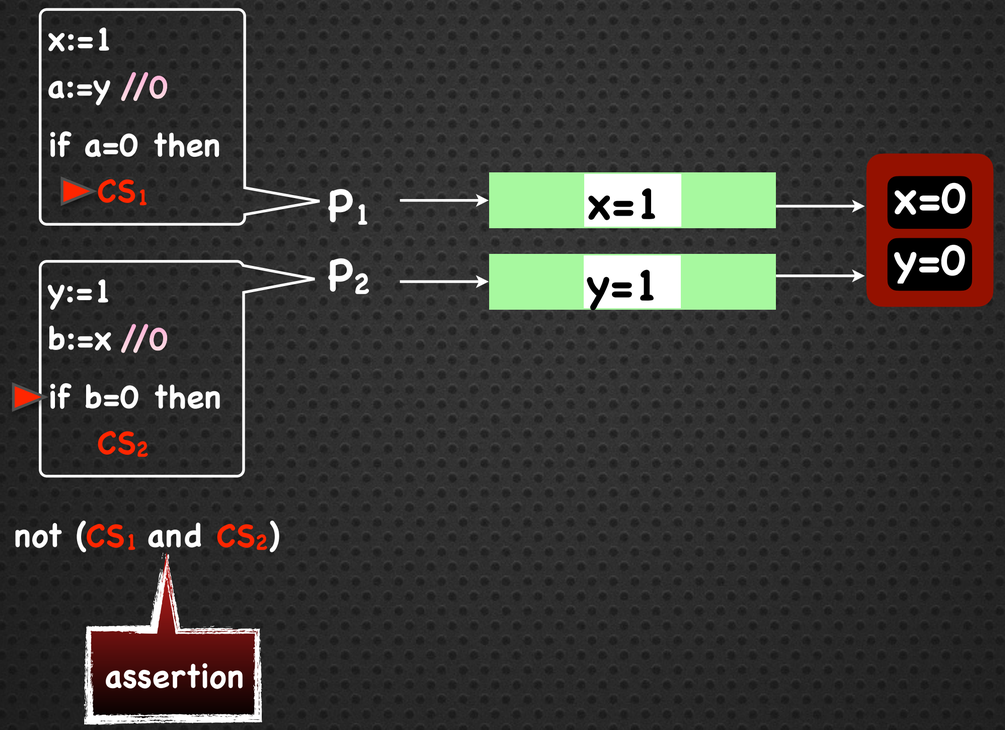
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

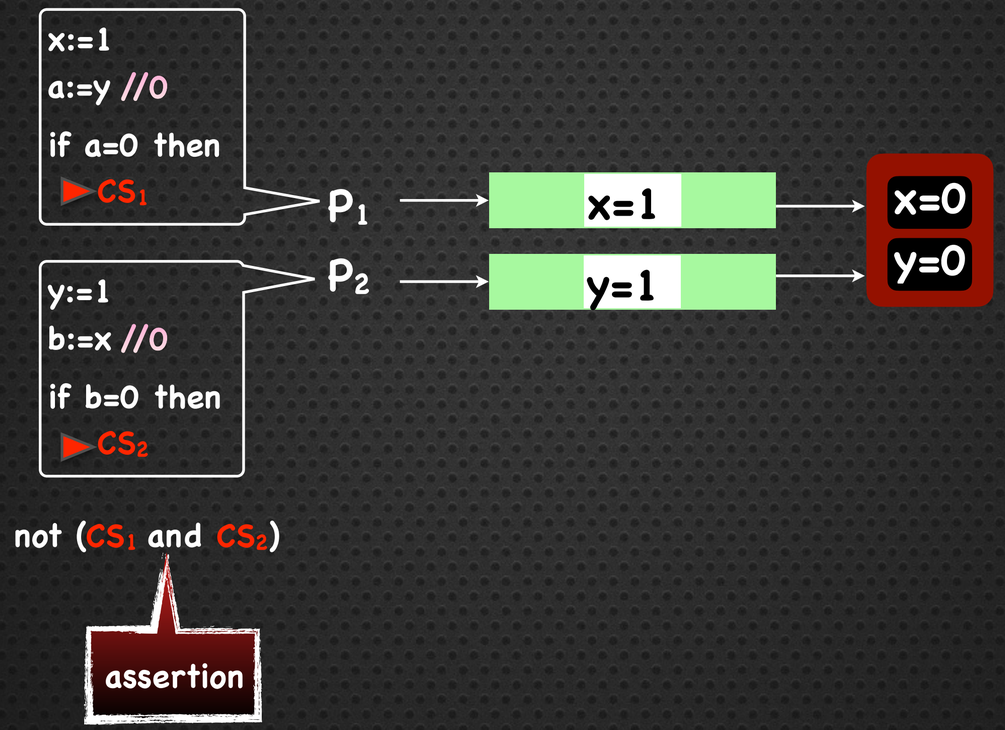
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

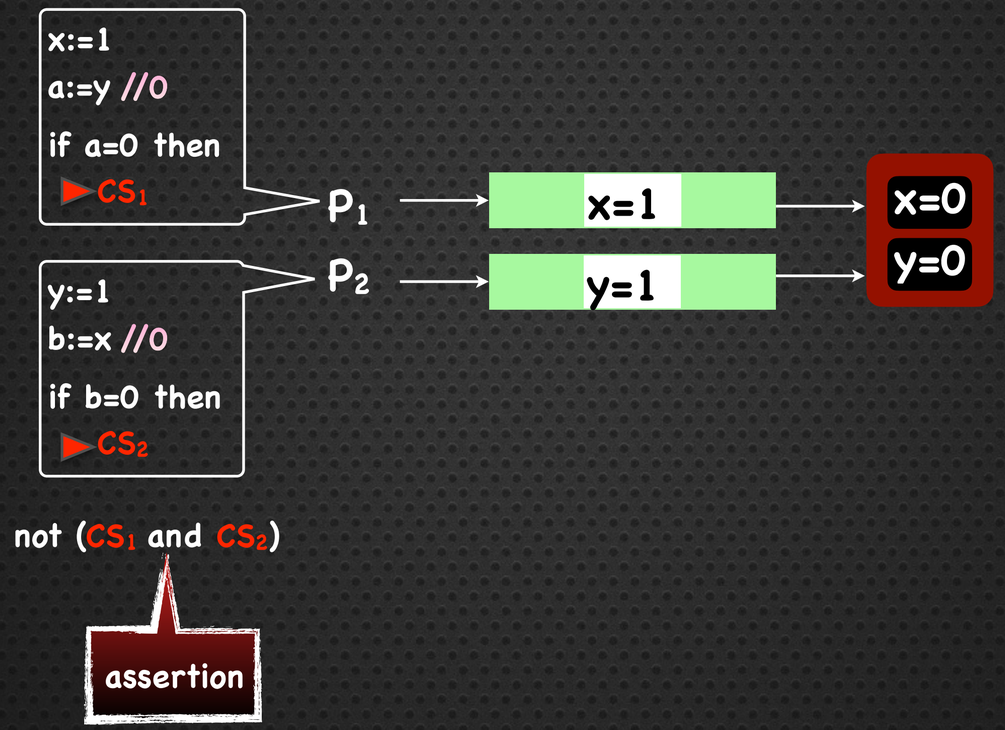
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

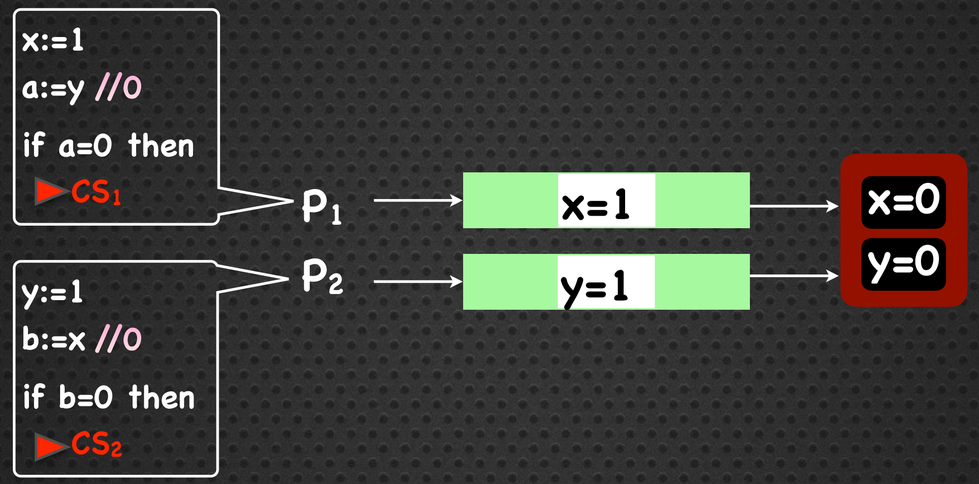
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

extra behaviors

weak memory model

potentially bad behaviors

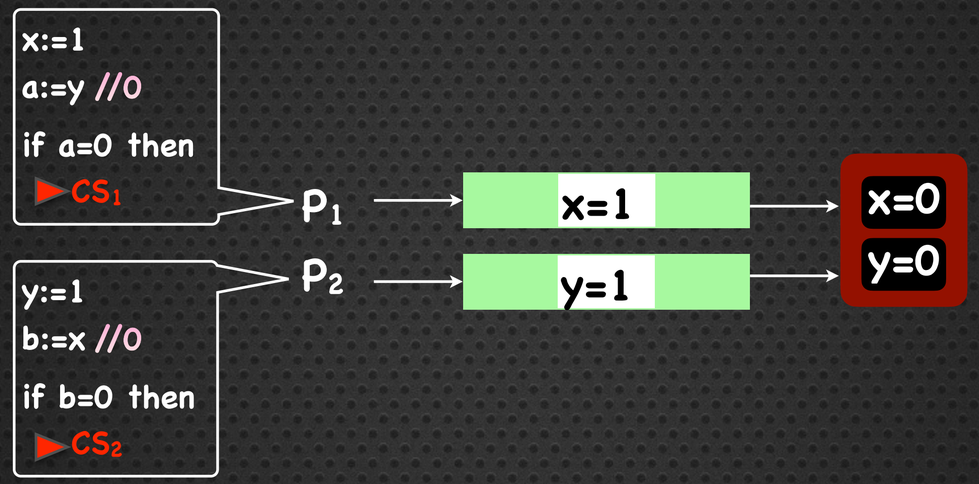
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

extra behaviors

weak memory model

TSO

potentially bad behaviors

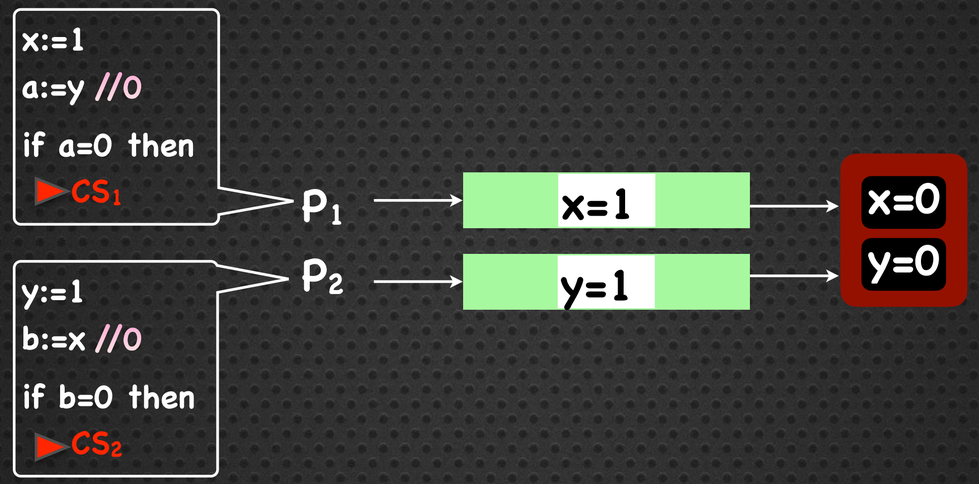
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

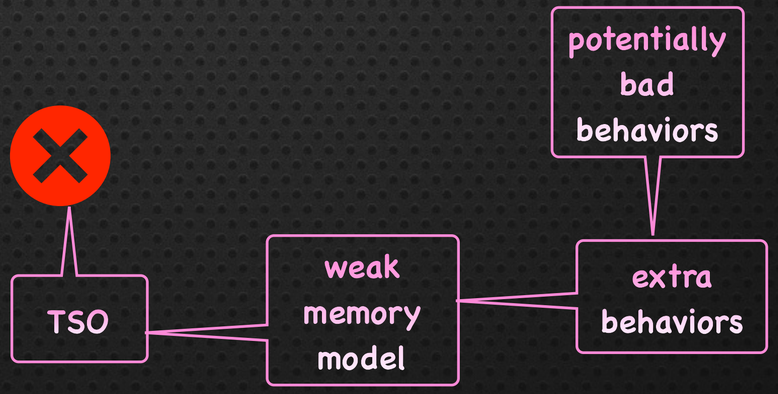
Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion



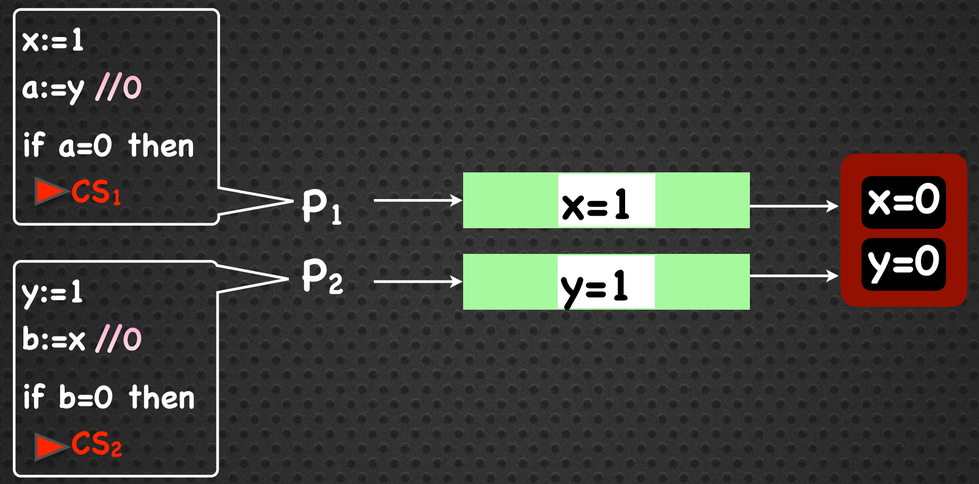


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

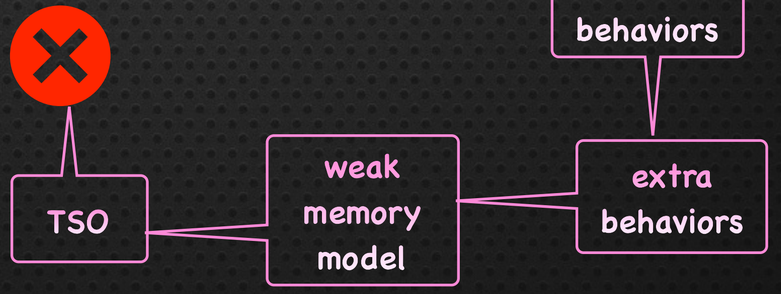
Dekker Protocol



not (CS_1 and CS_2)

assertion



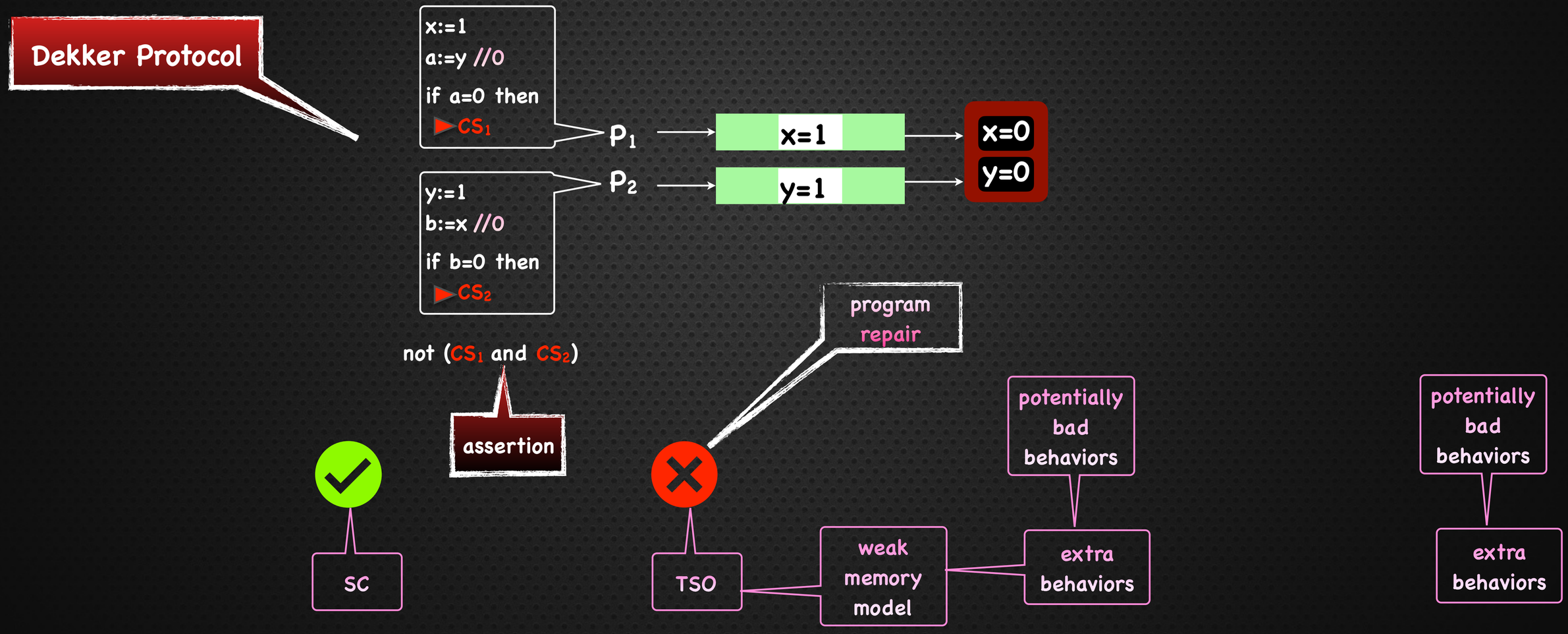




WALLPAPERSWIDE.COM

Shared Variable Concurrency

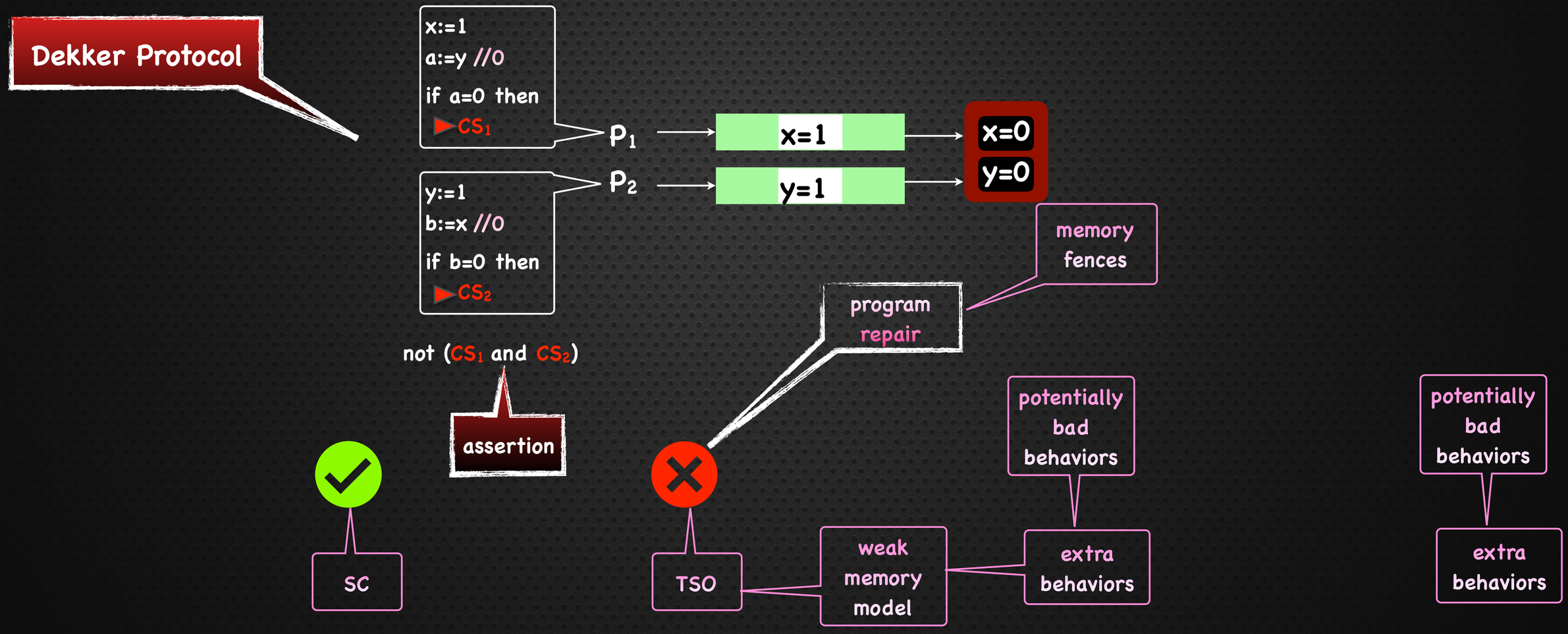
Total Store Order (TSO)



WALLPAPERSWIDE.COM

Shared Variable Concurrency

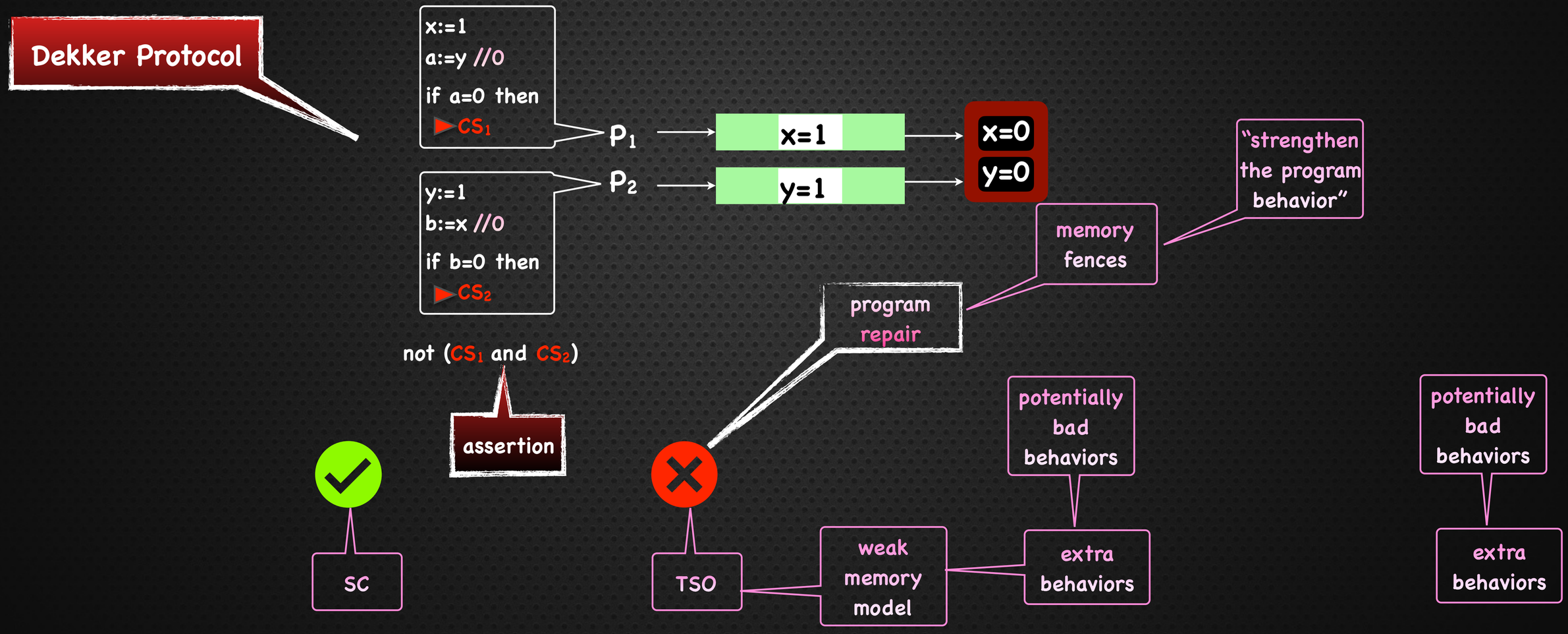
Total Store Order (TSO)



WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)



WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

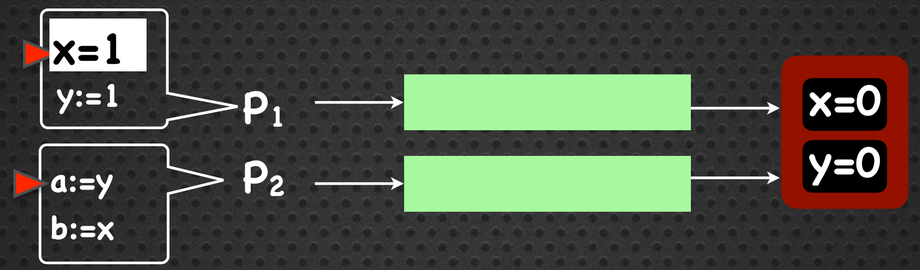
- Sequential Consistency (SC)
- Total Store Ordering (TSO) (Examples)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

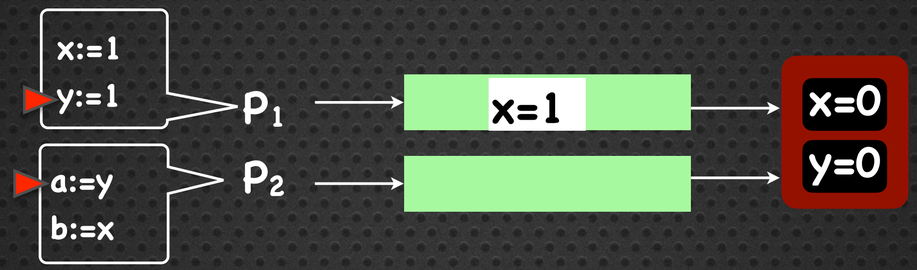
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

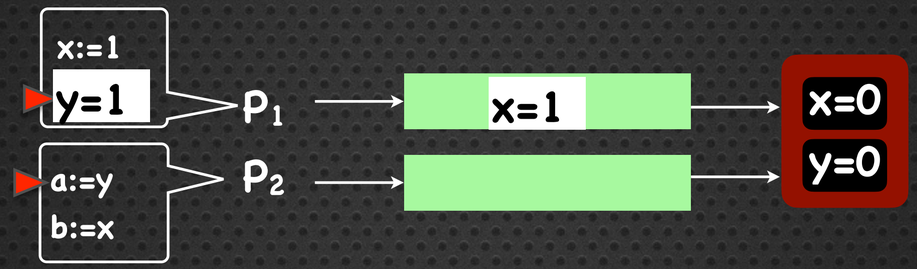
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

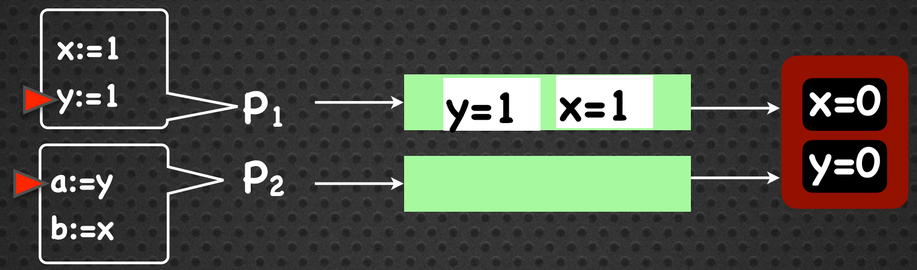
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

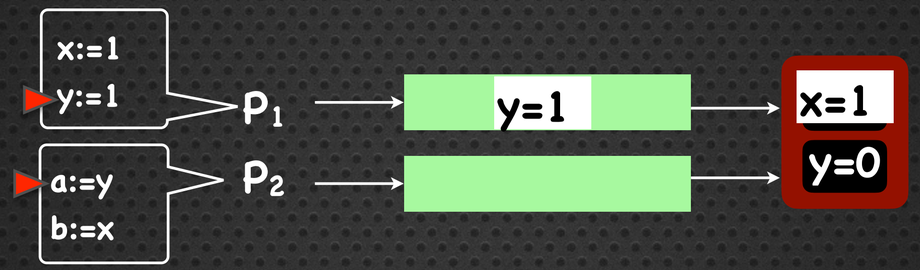
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

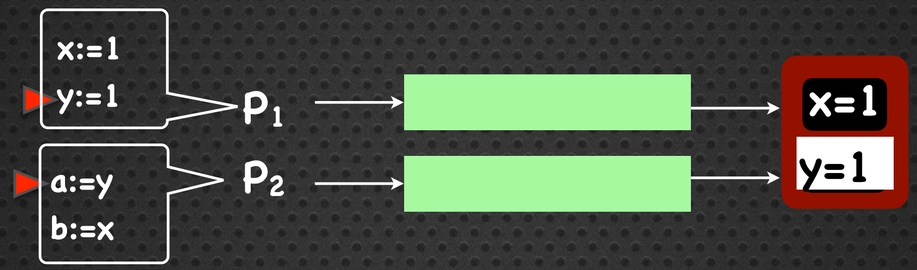
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

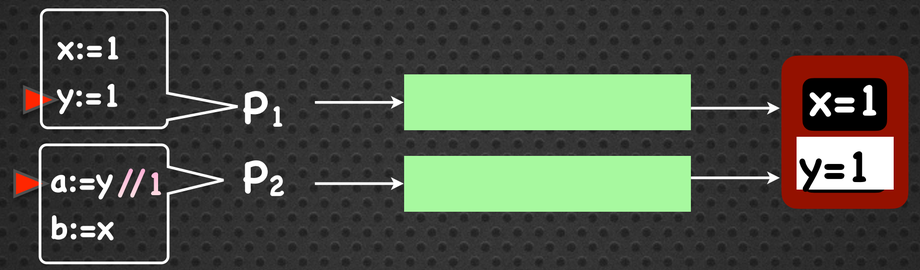
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

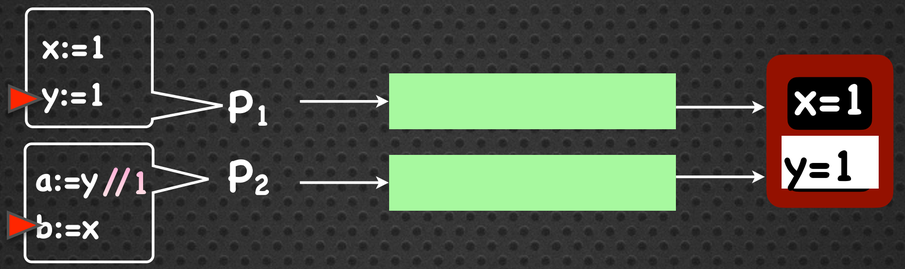
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

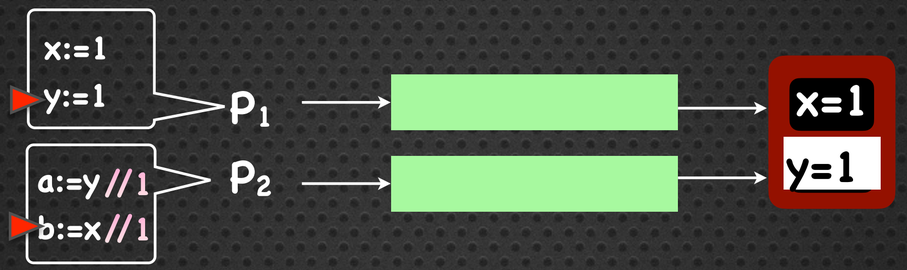
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



a=1 and b=0

potentially bad behaviors

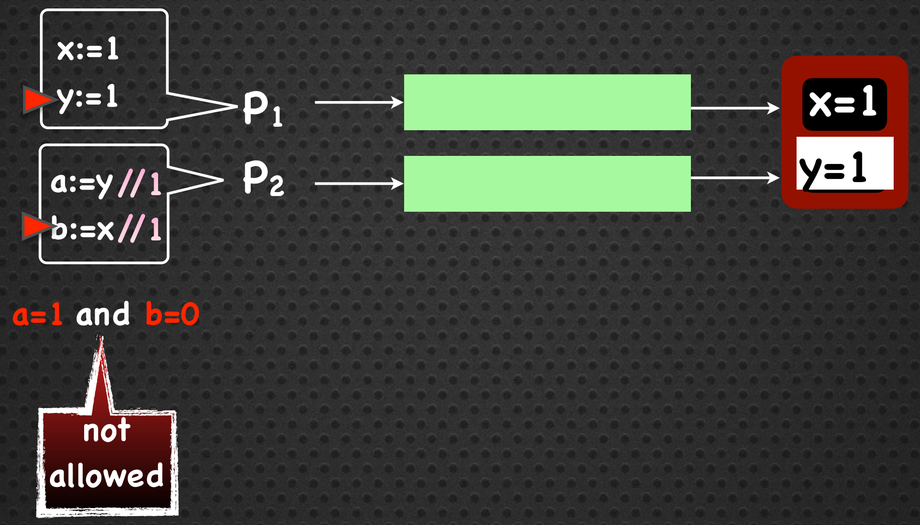
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Message Passing



potentially bad behaviors

extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

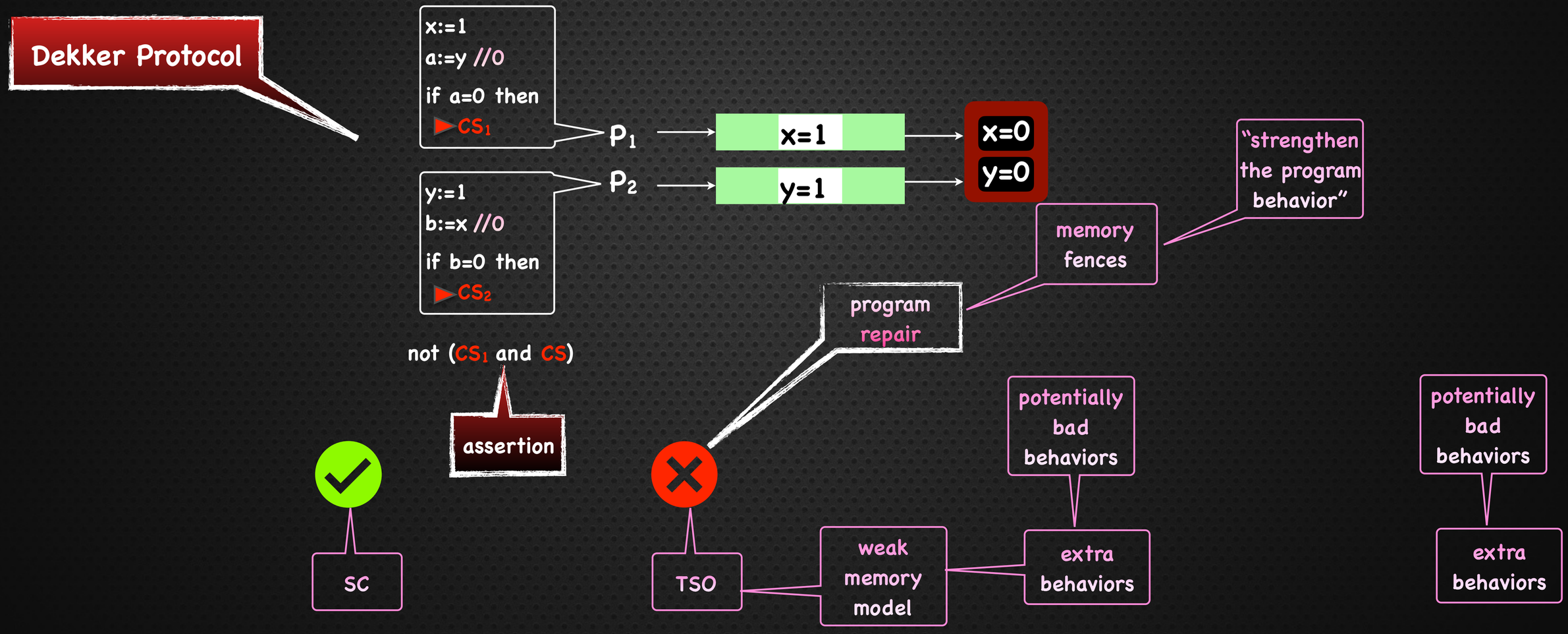
Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

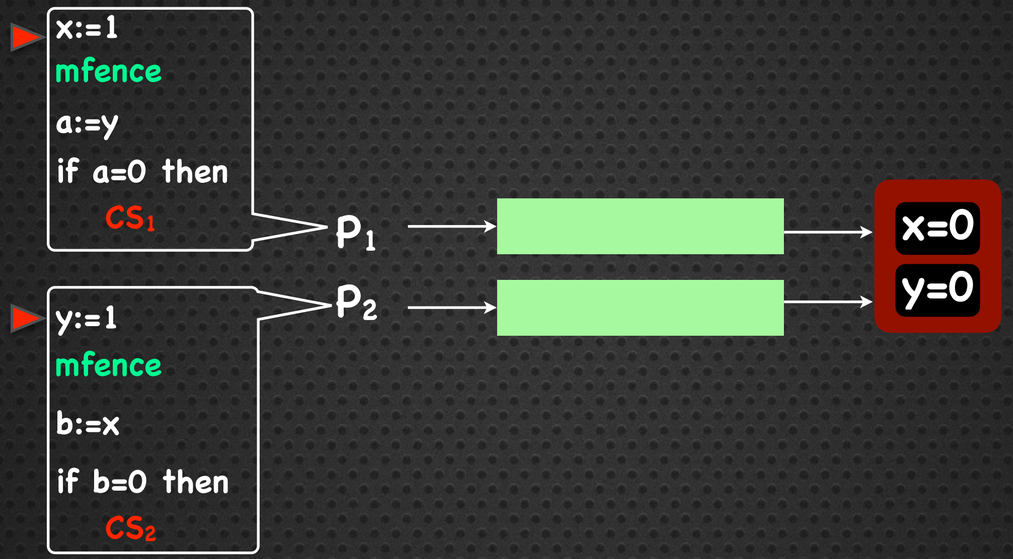


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

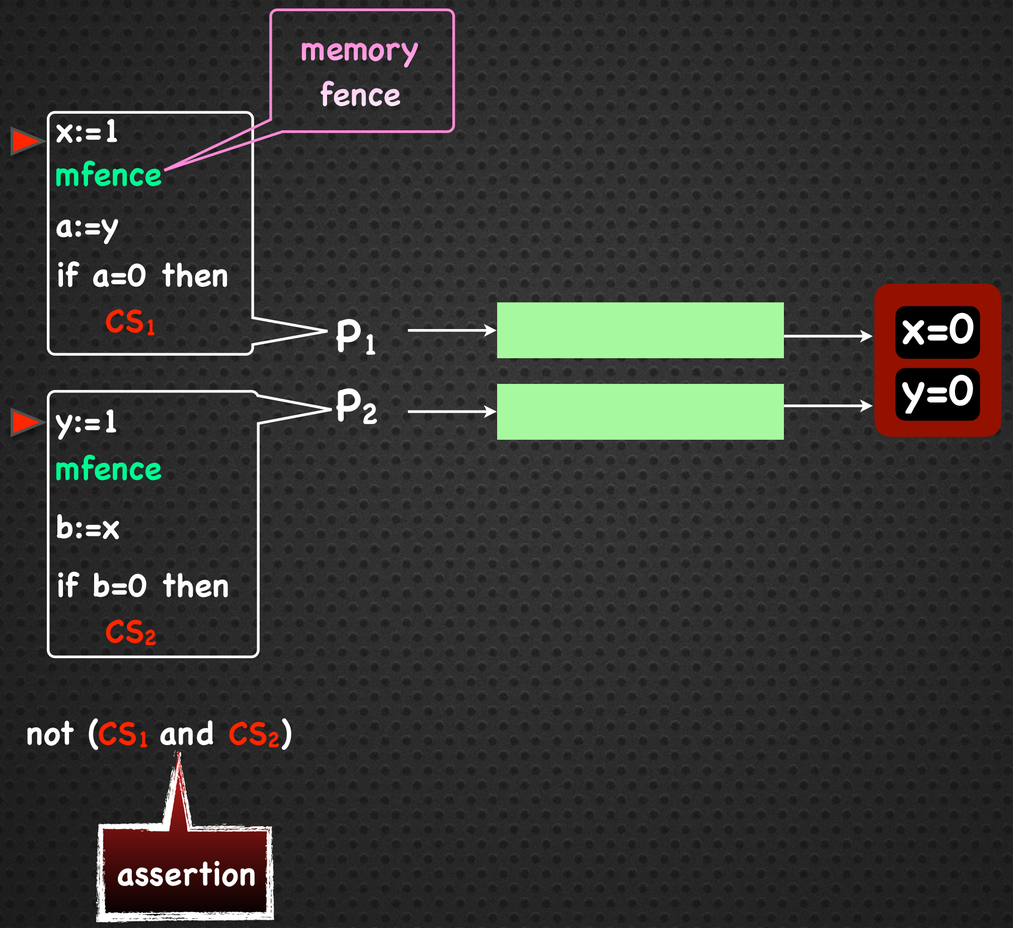
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

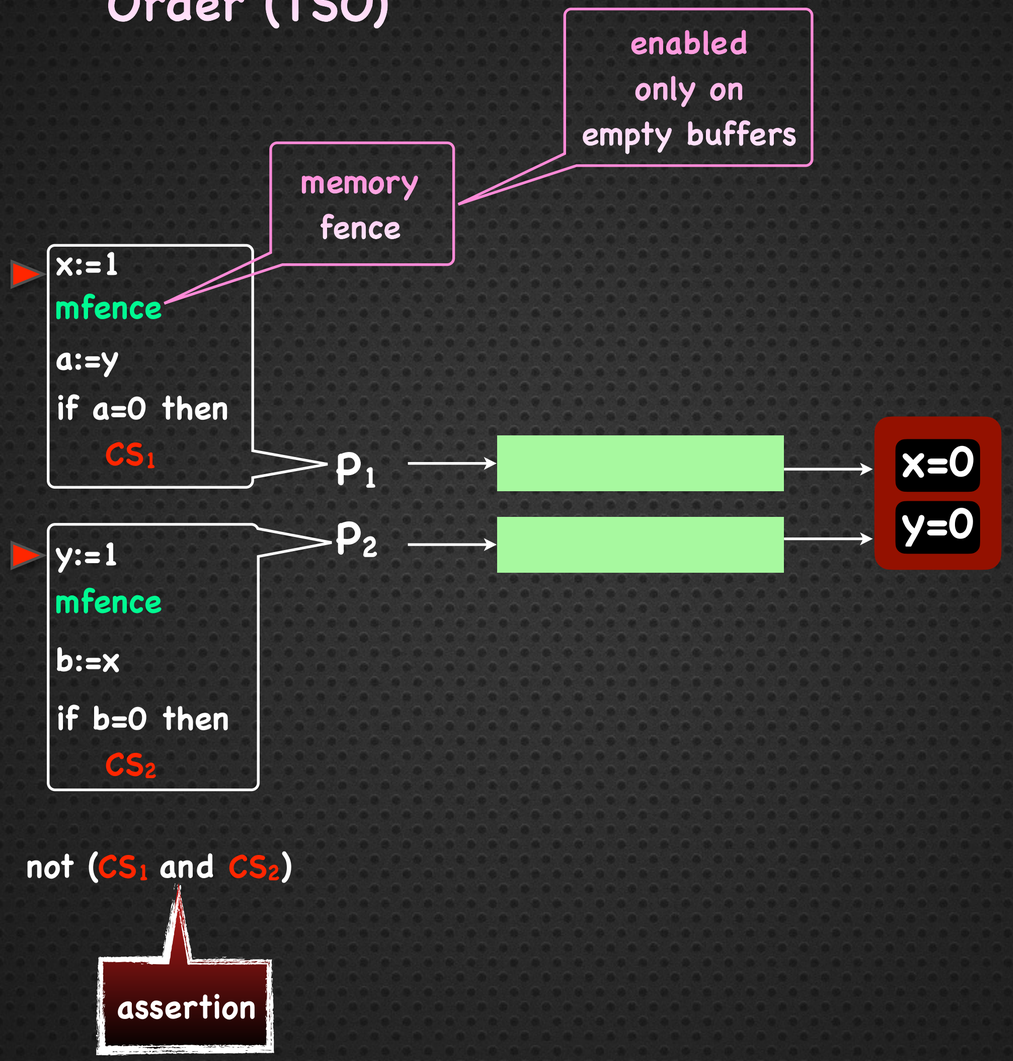
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

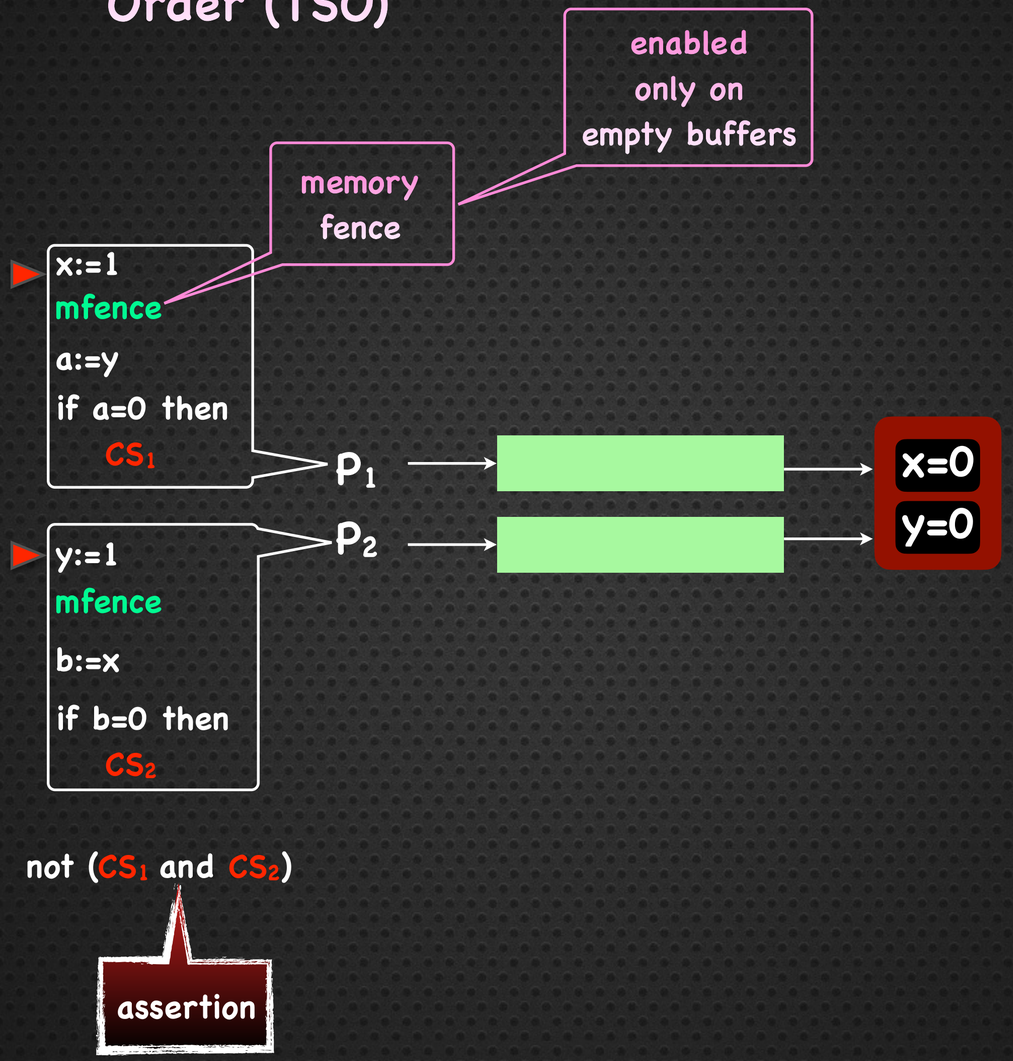
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

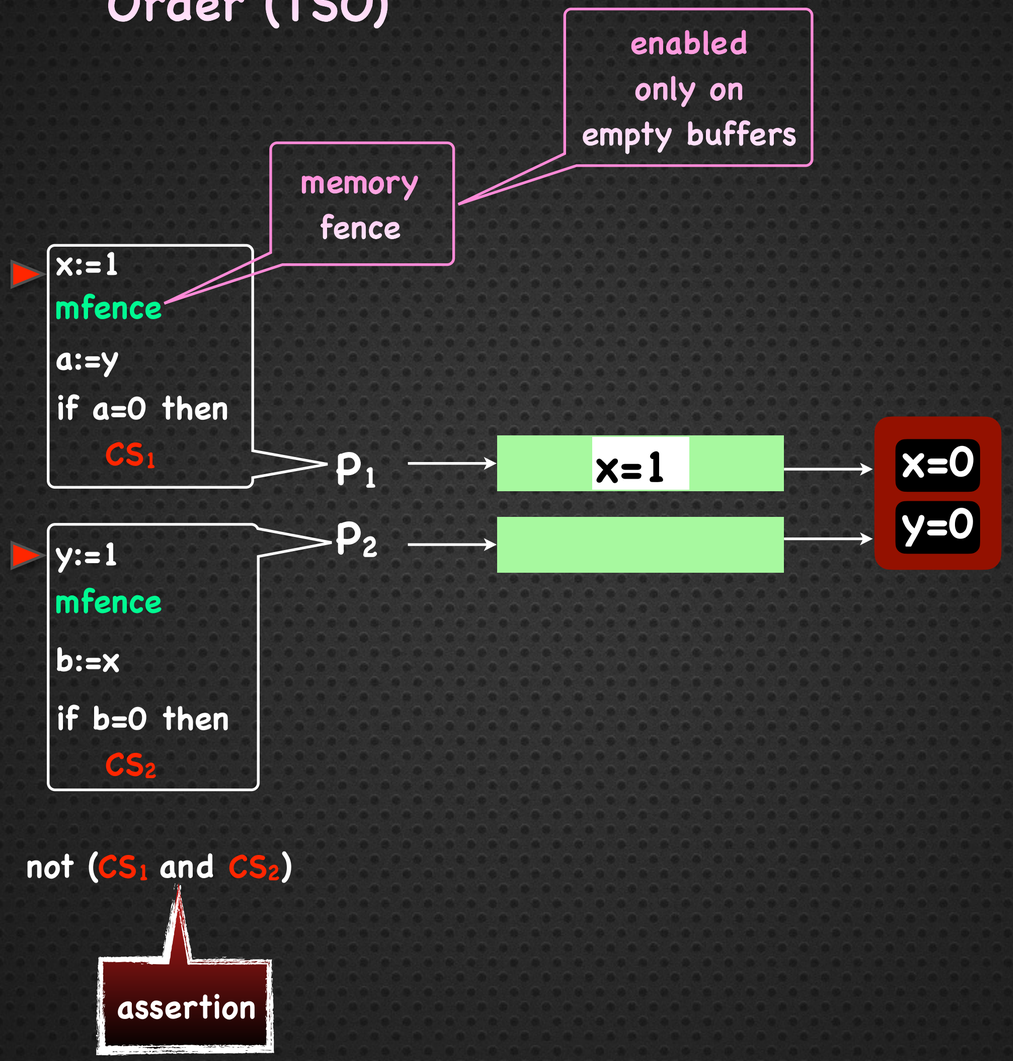
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

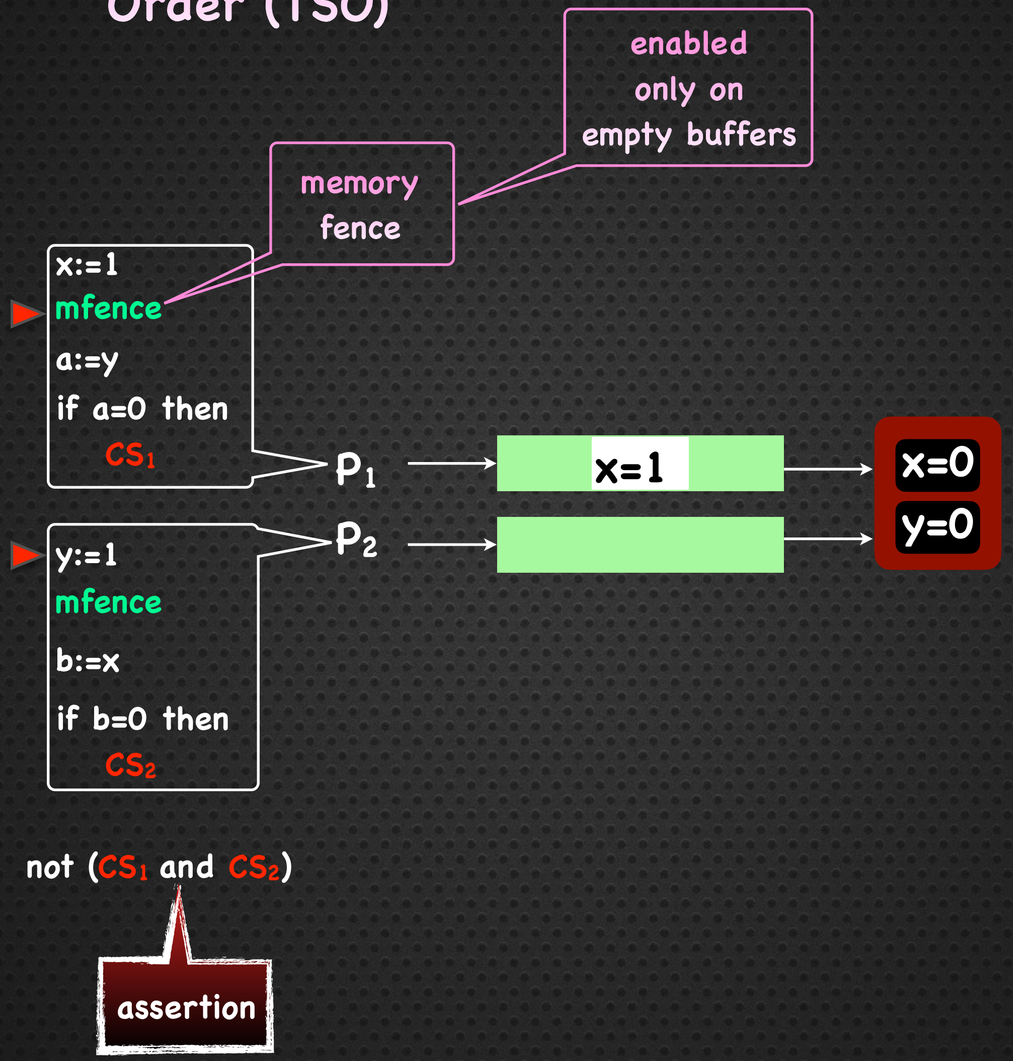
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

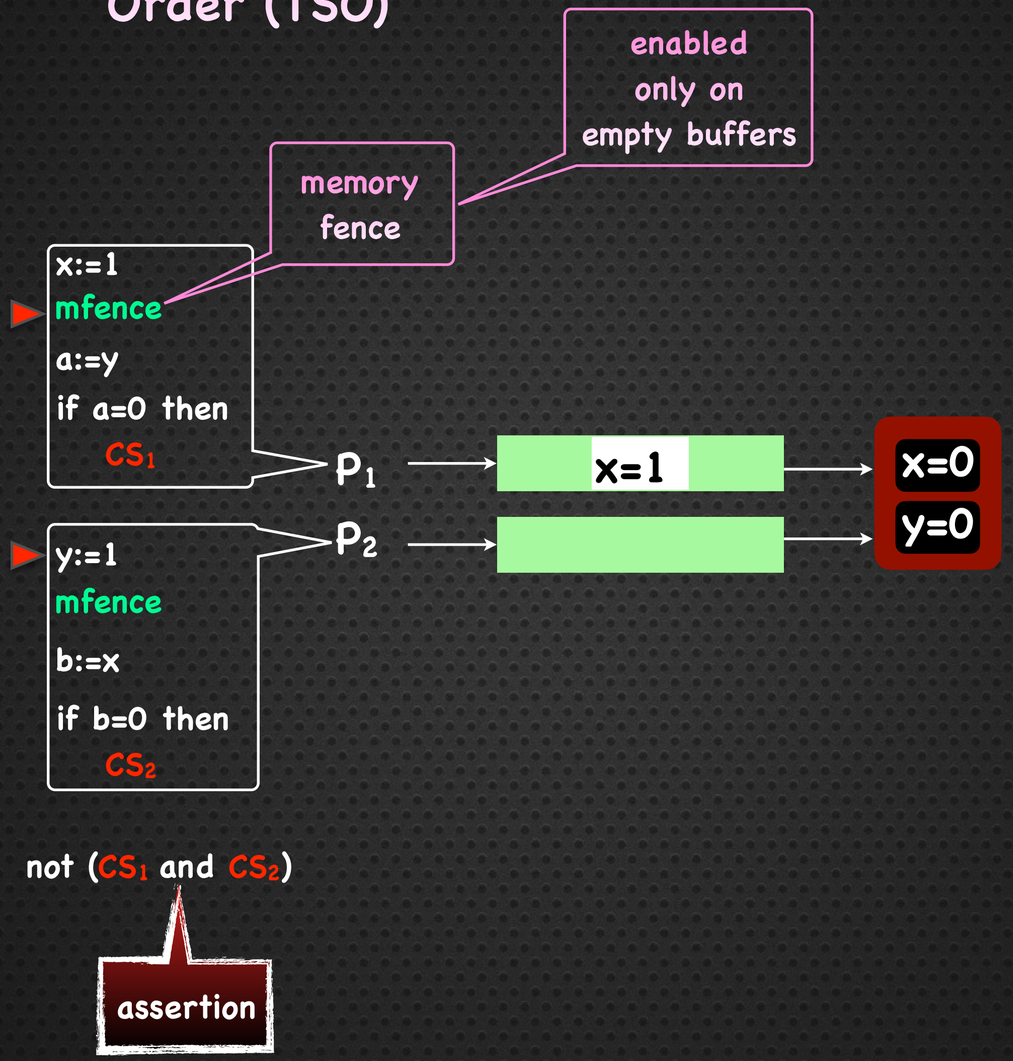
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



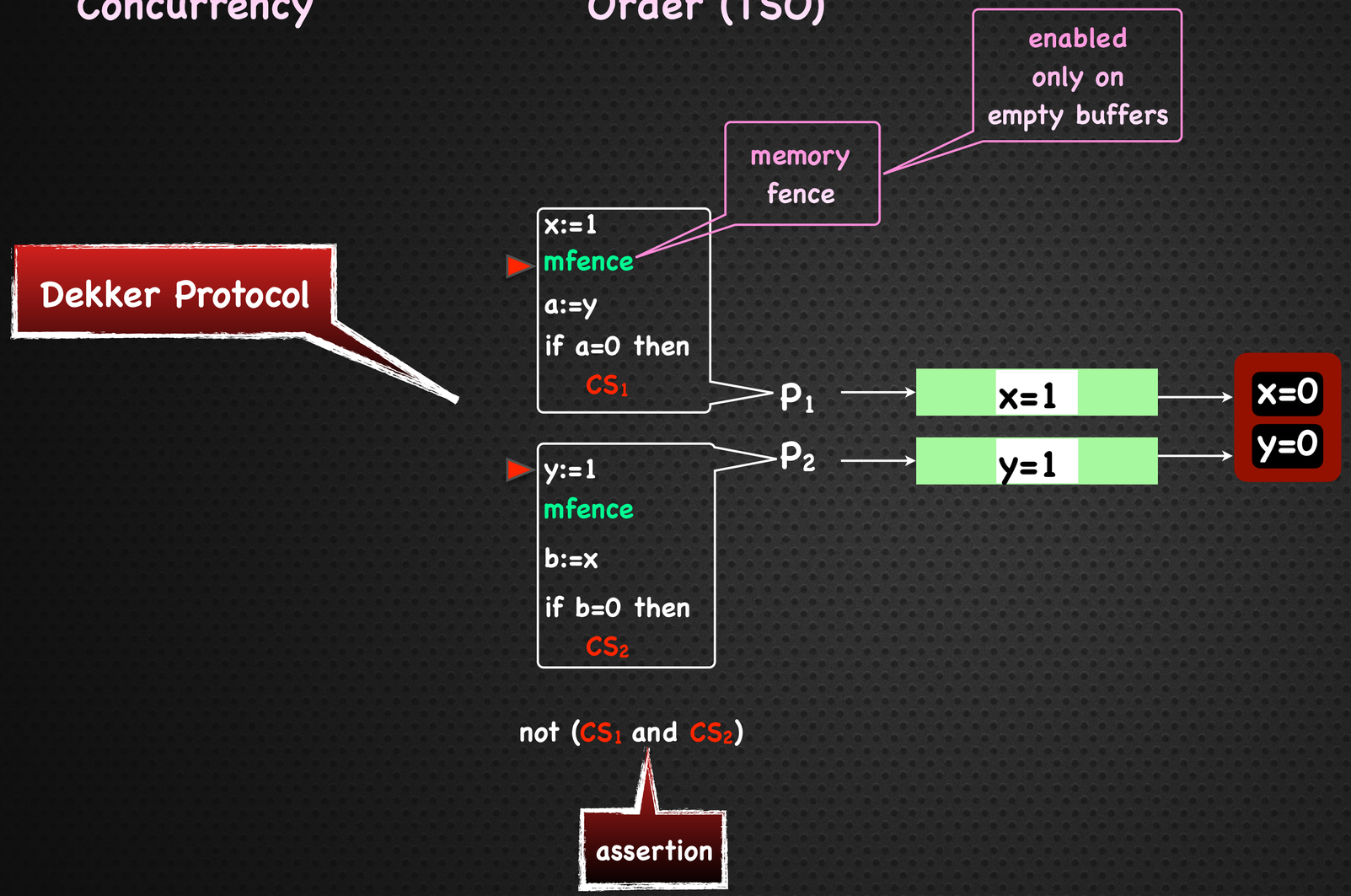
potentially bad behaviors

extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)



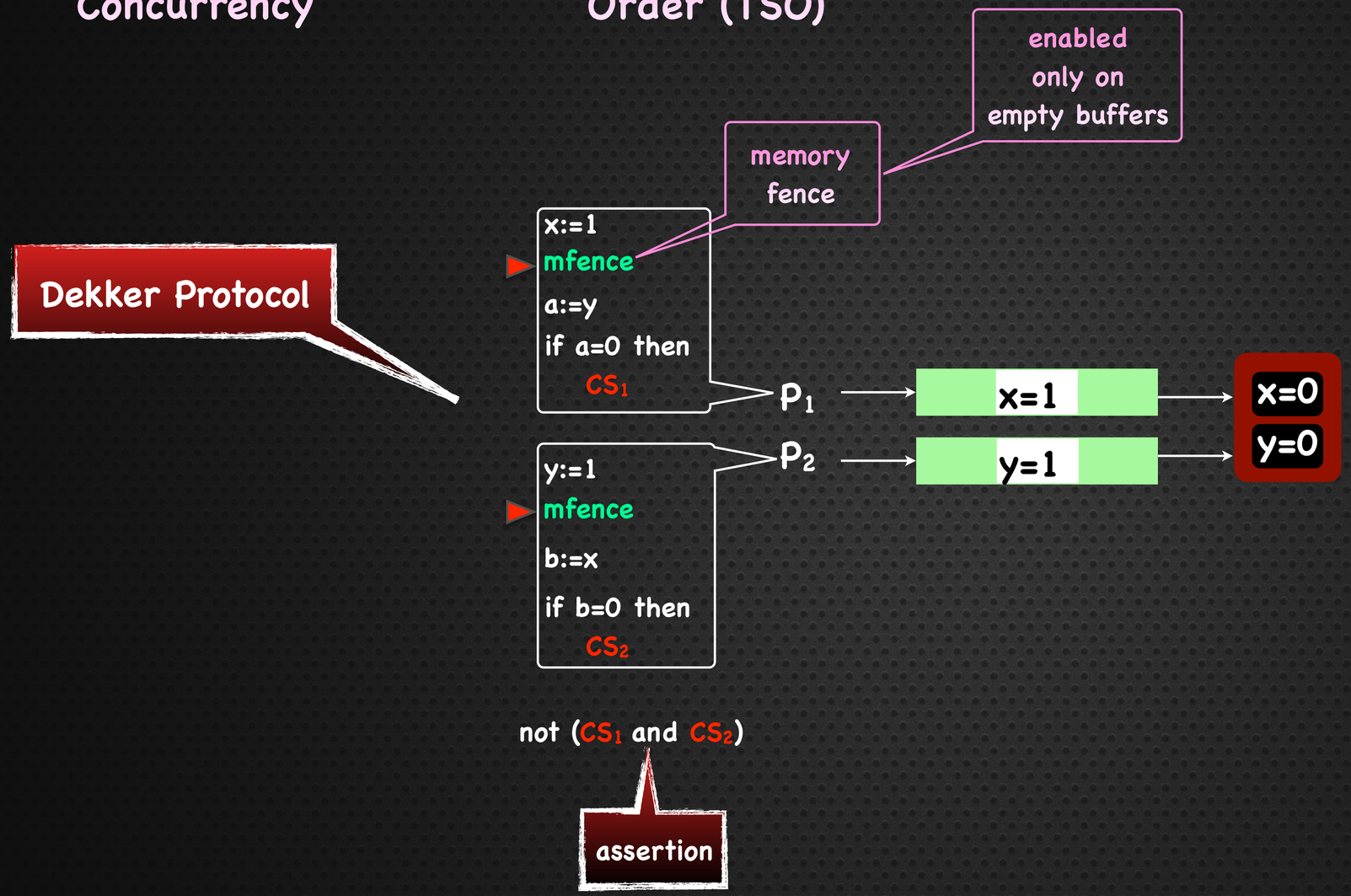
potentially bad behaviors

extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)



potentially bad behaviors

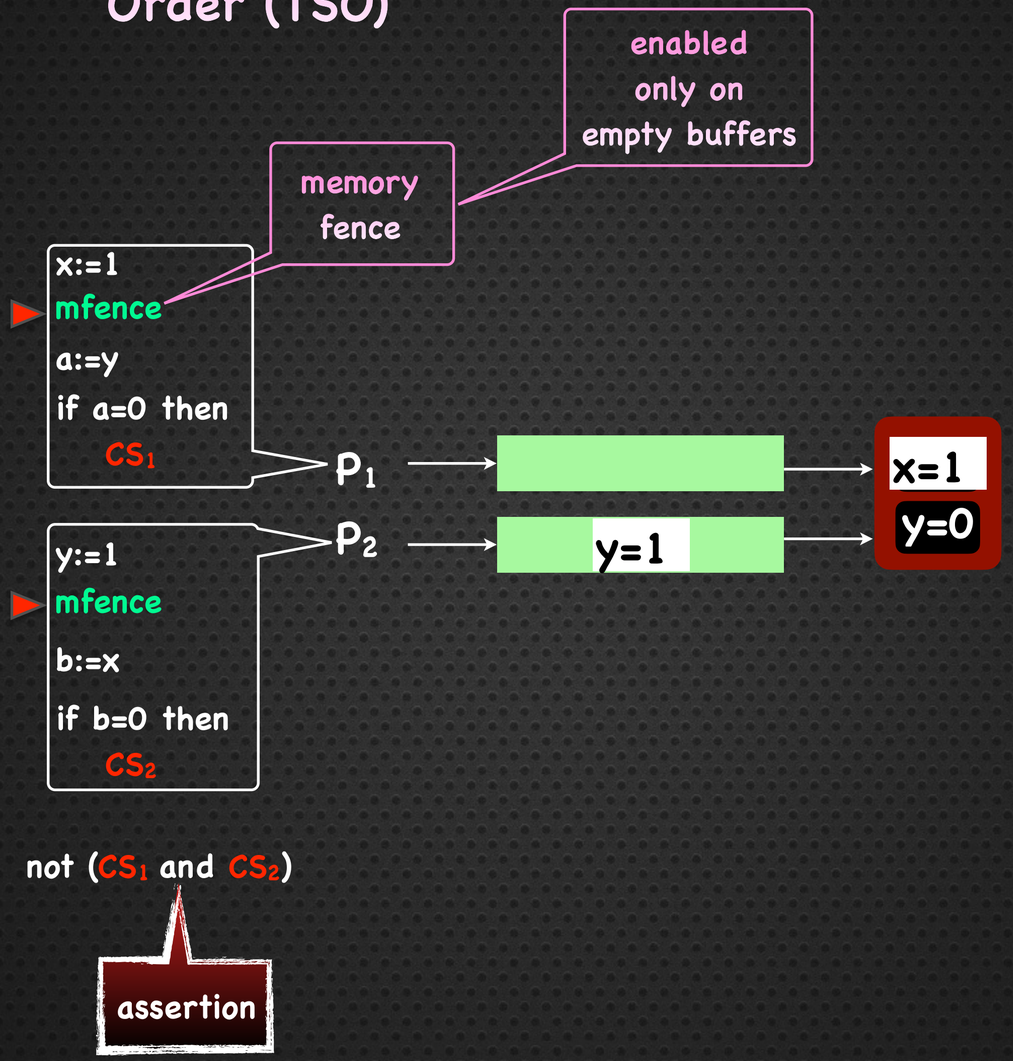
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

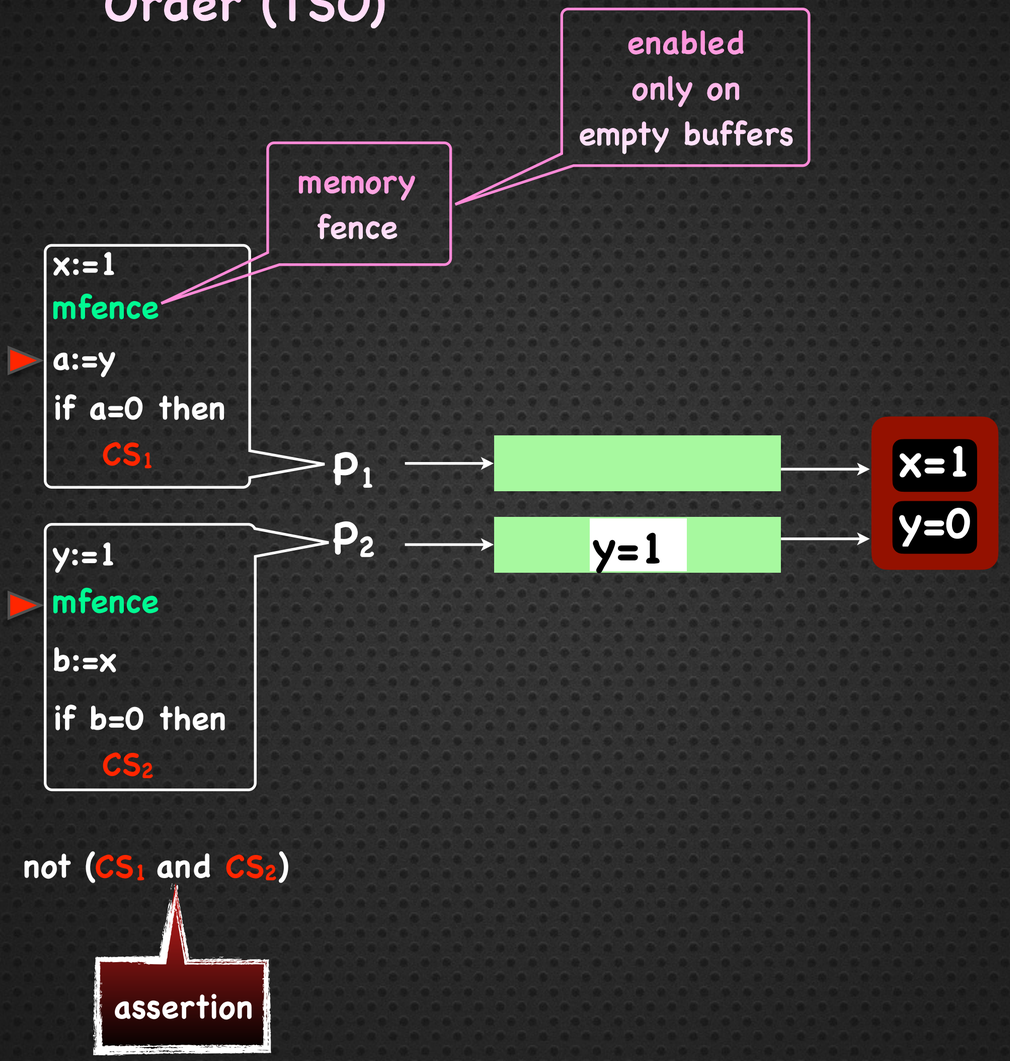
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



potentially bad behaviors

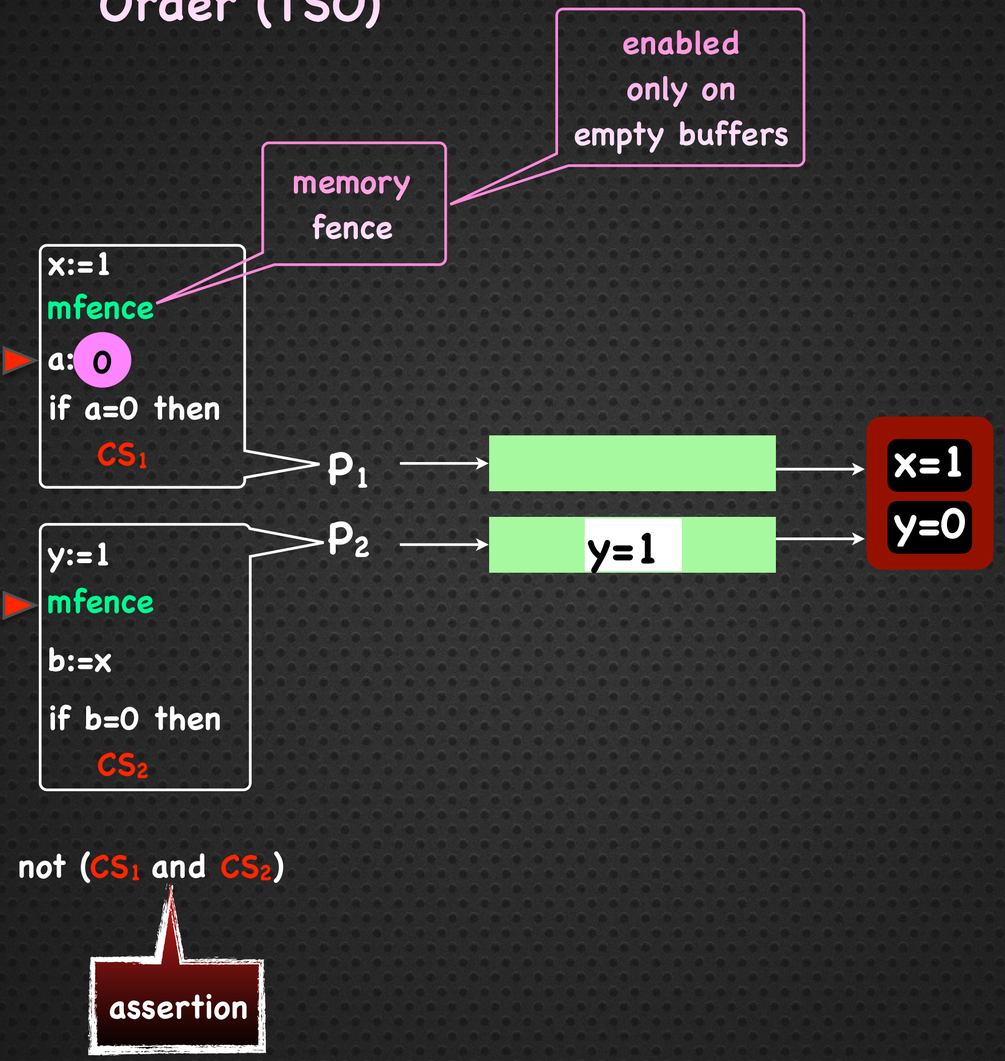
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

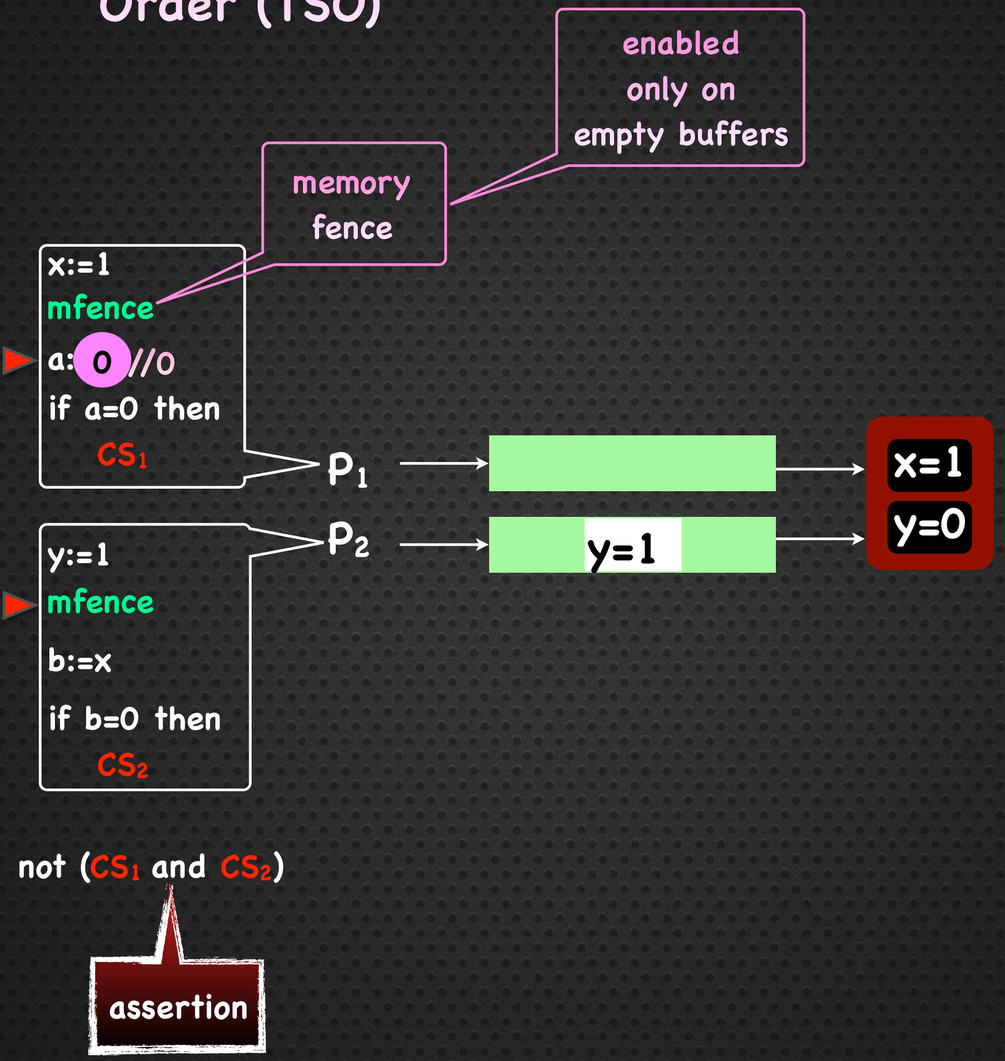
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

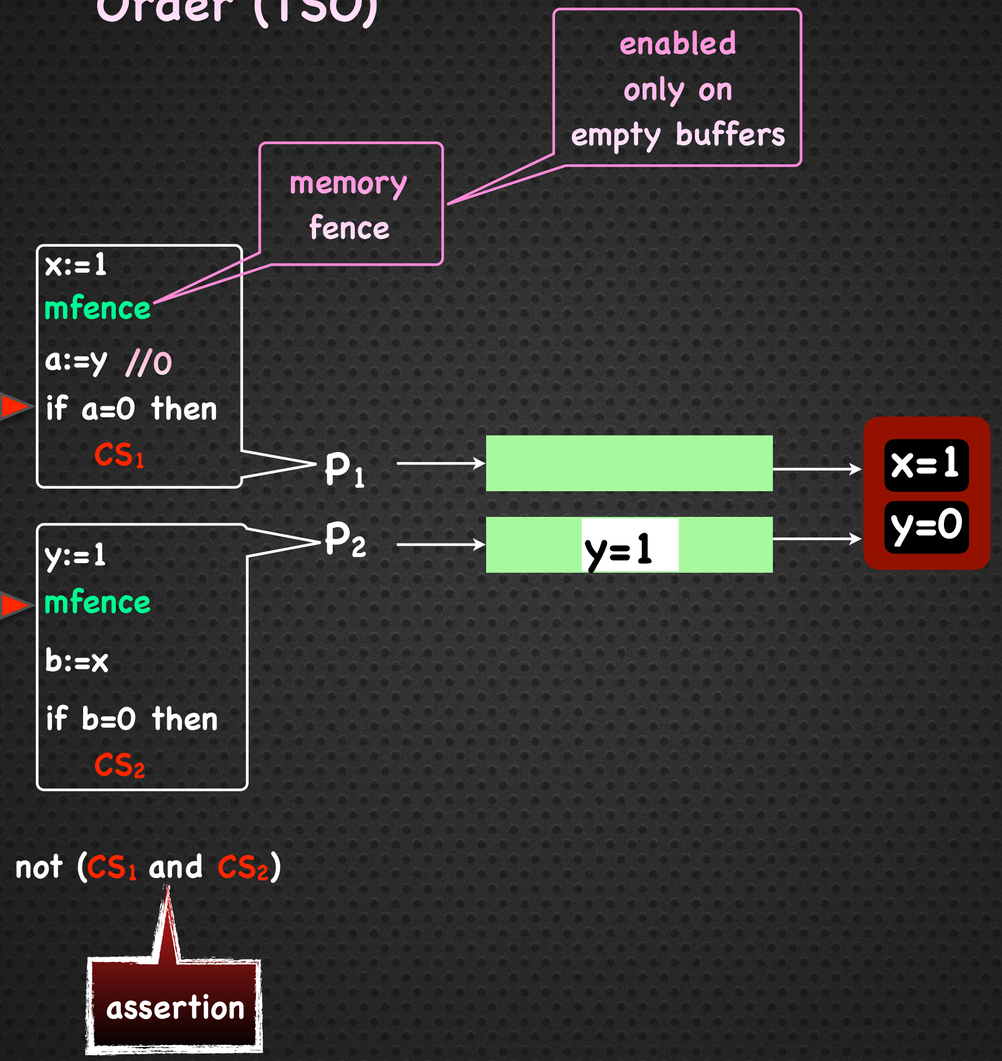
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

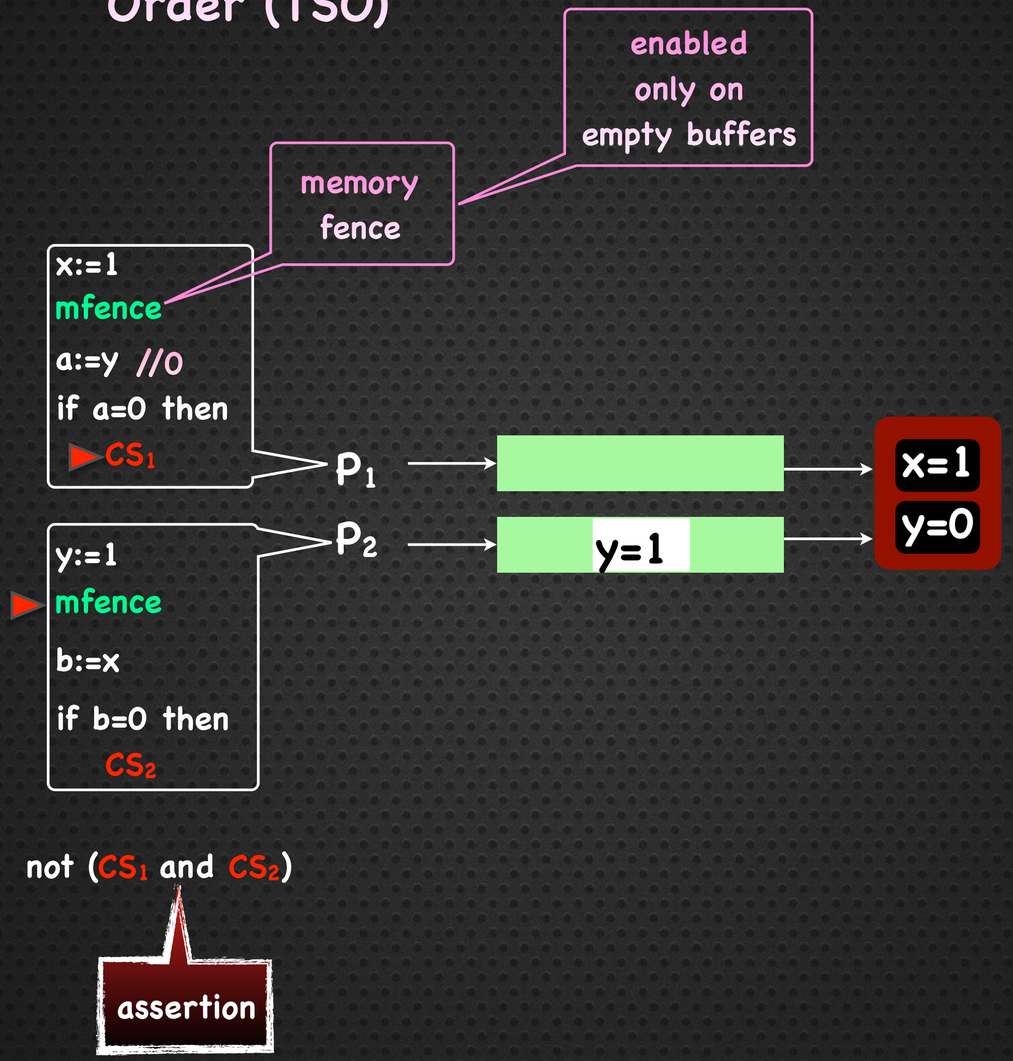
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

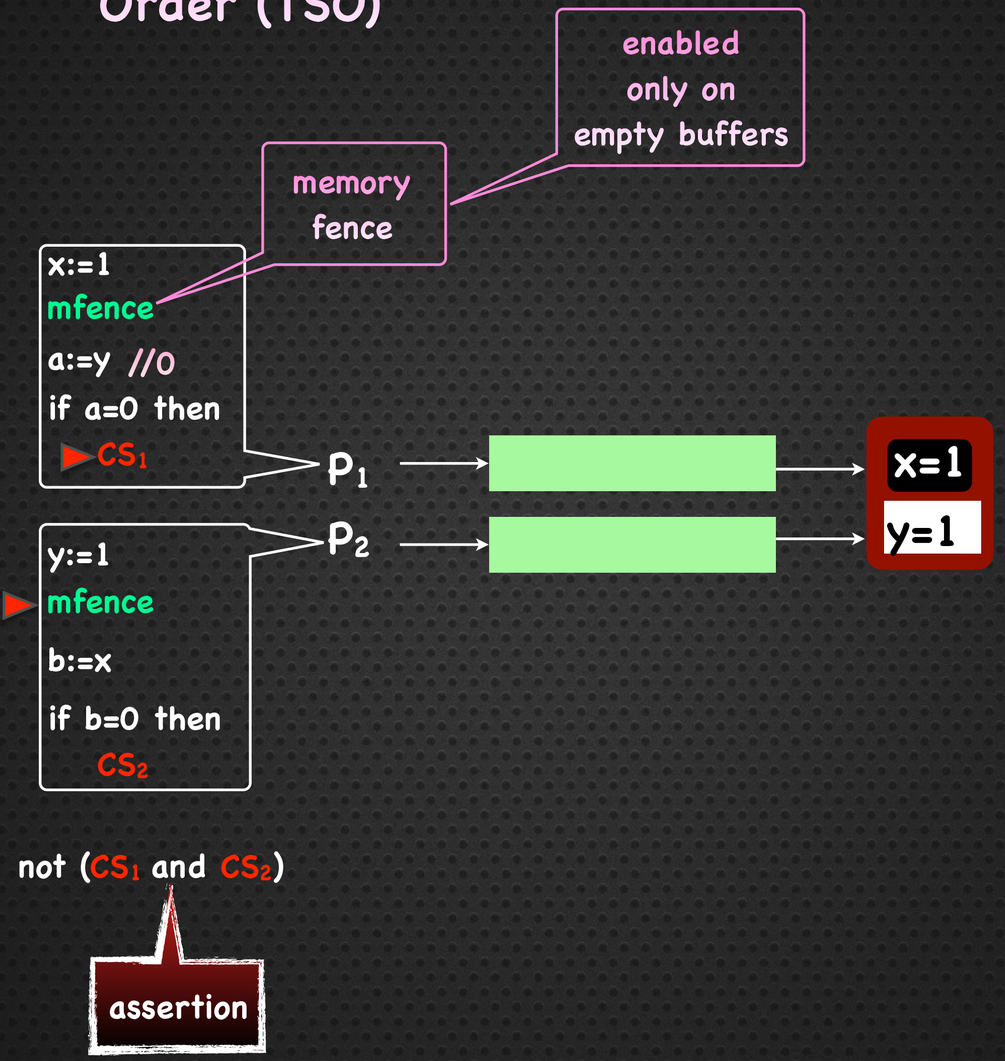
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

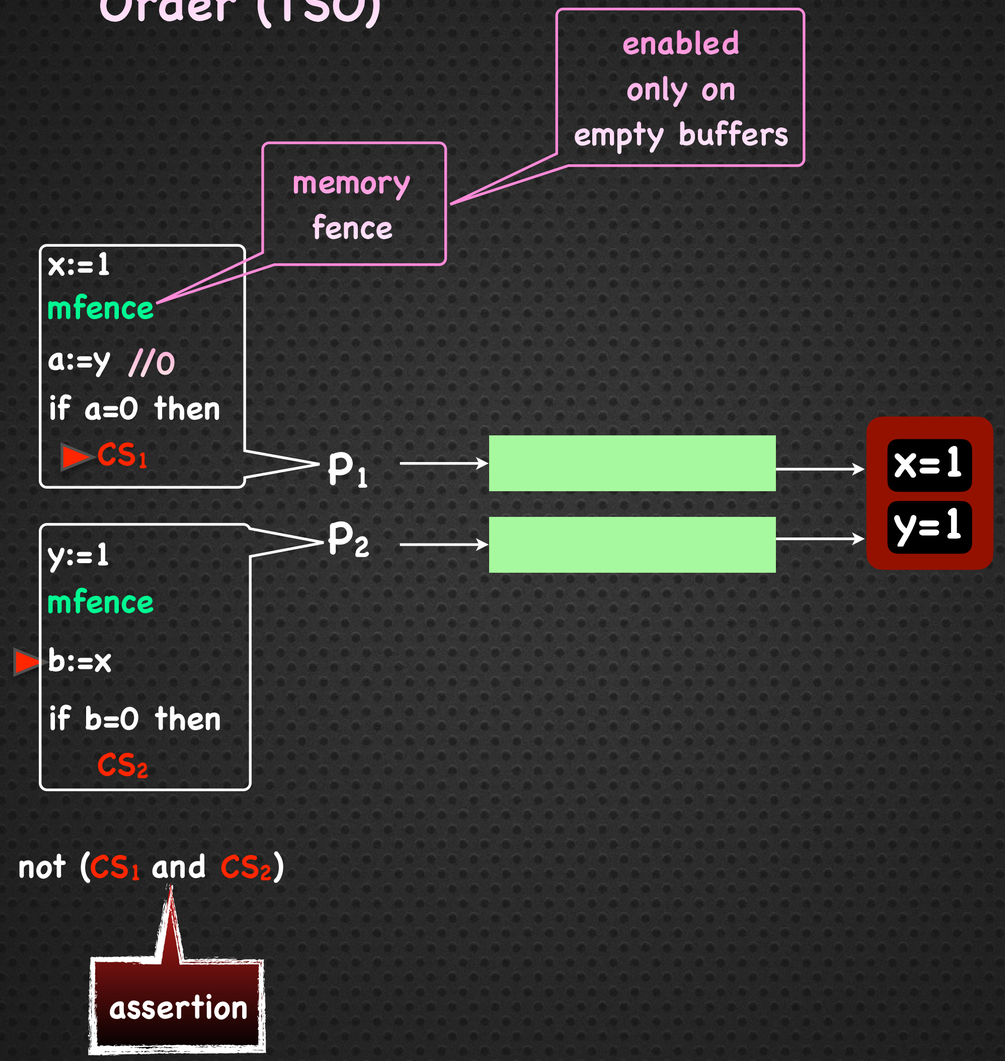
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

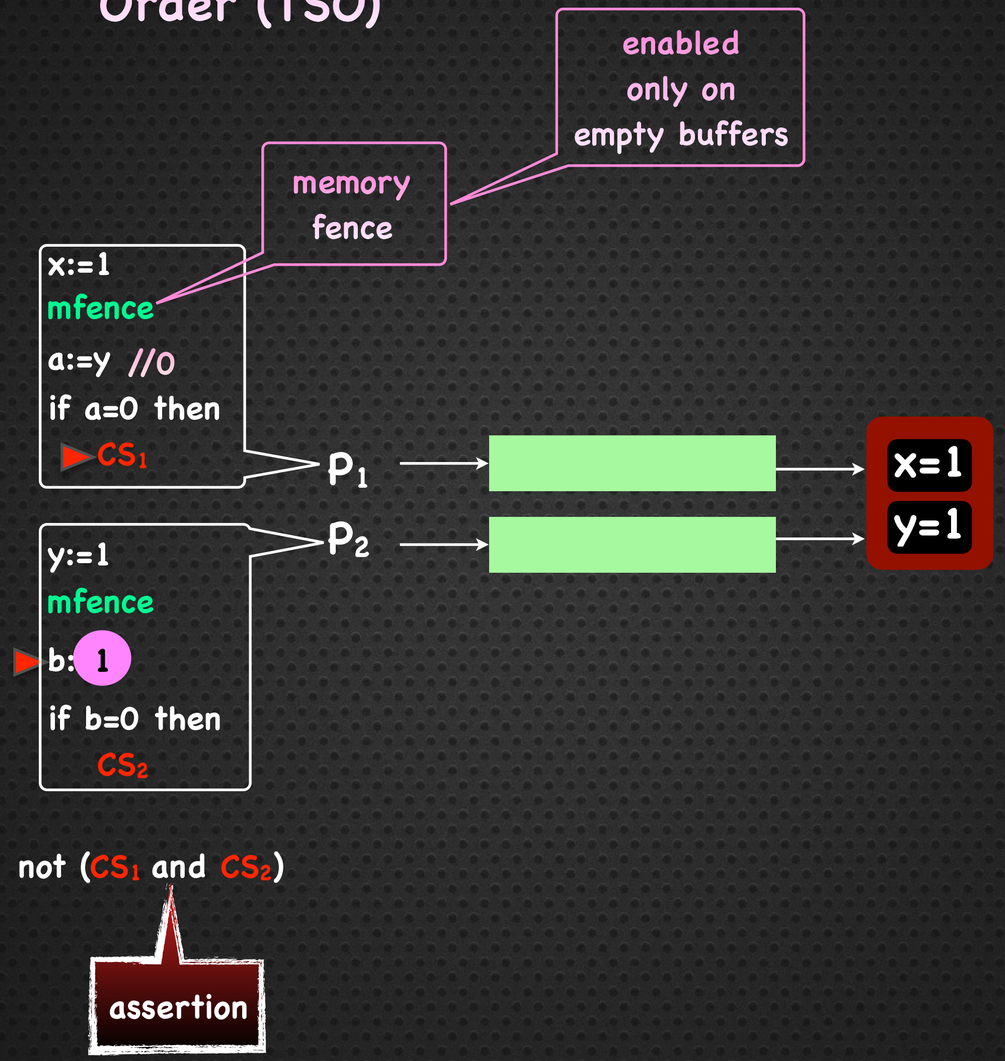
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

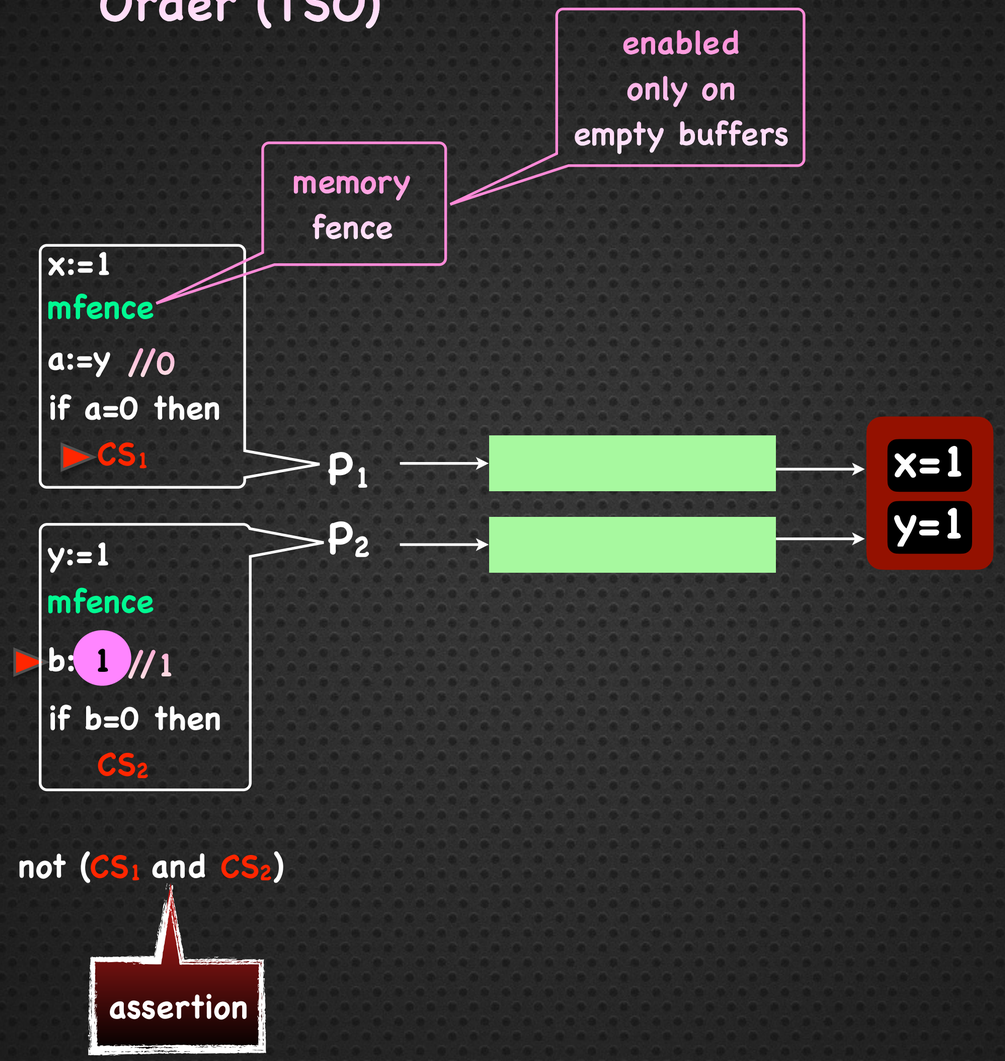
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

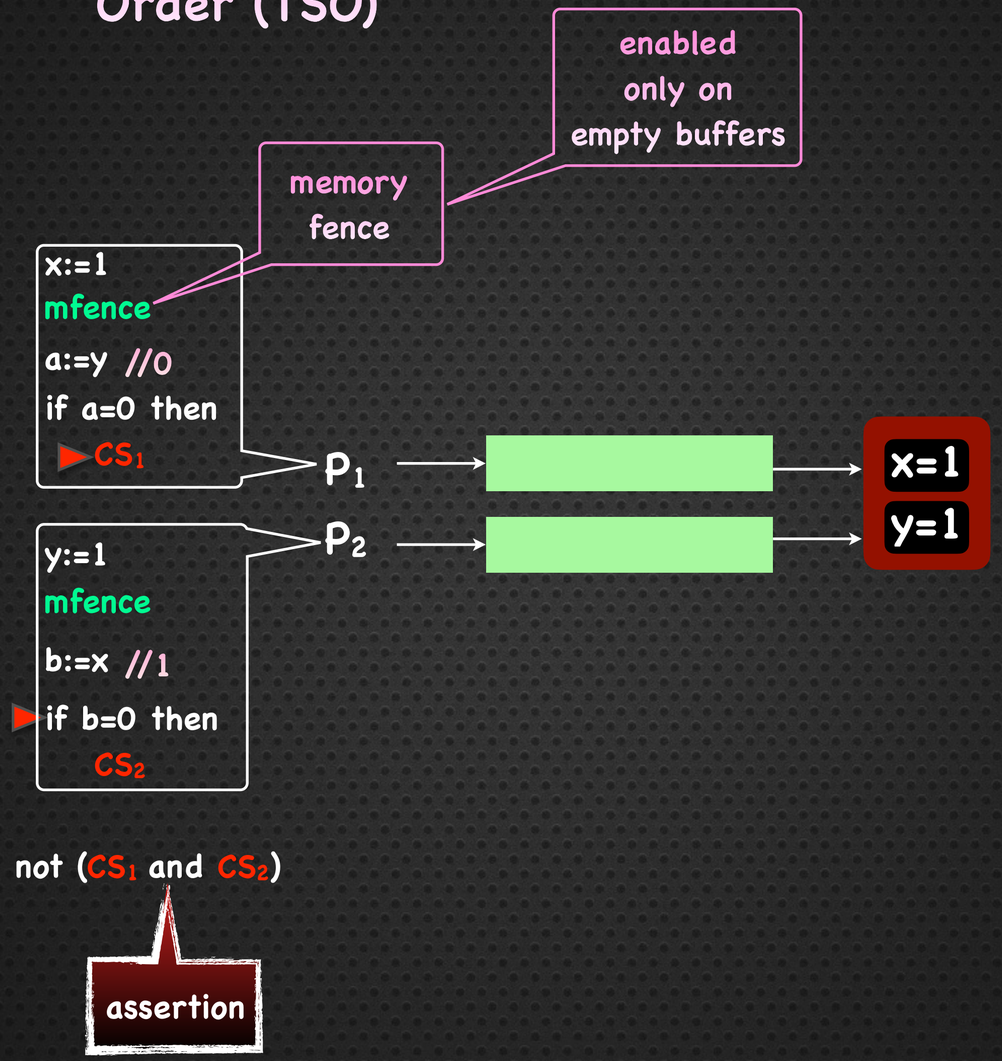
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

potentially bad behaviors

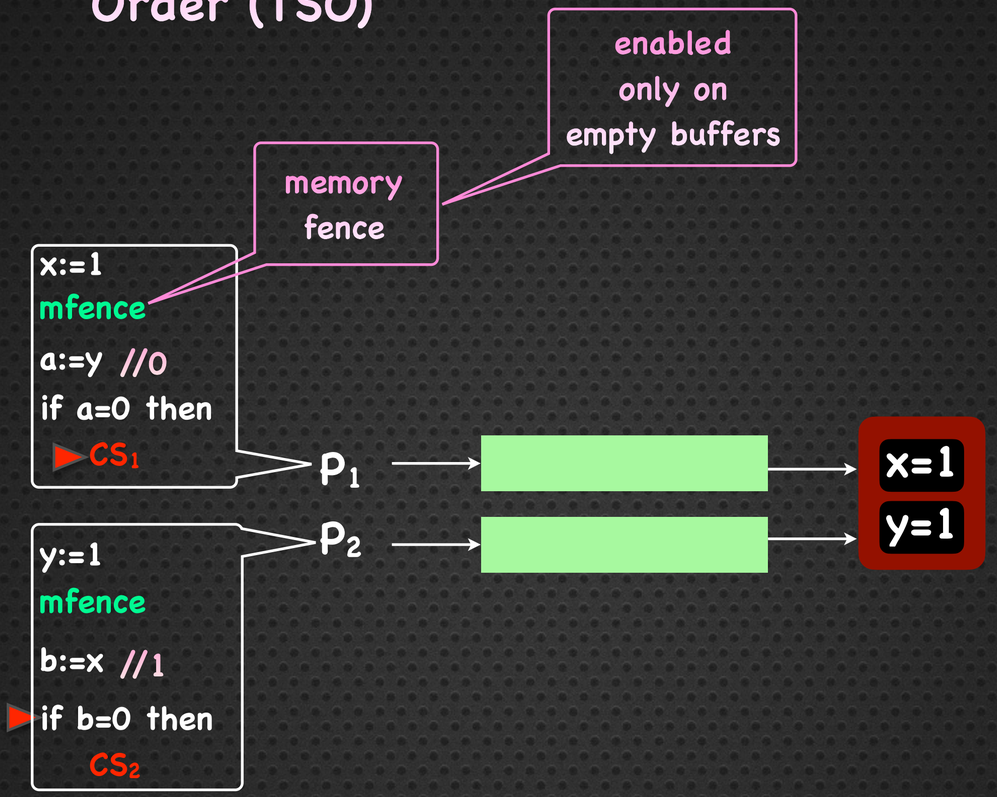
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

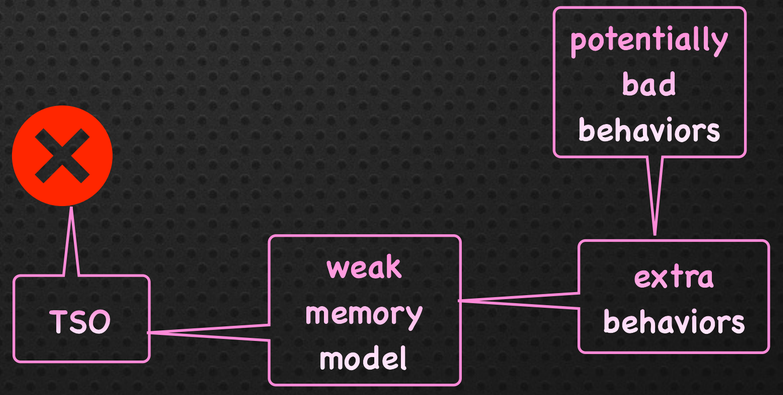
Total Store Order (TSO)

Dekker Protocol



not (CS_1 and CS_2)

assertion

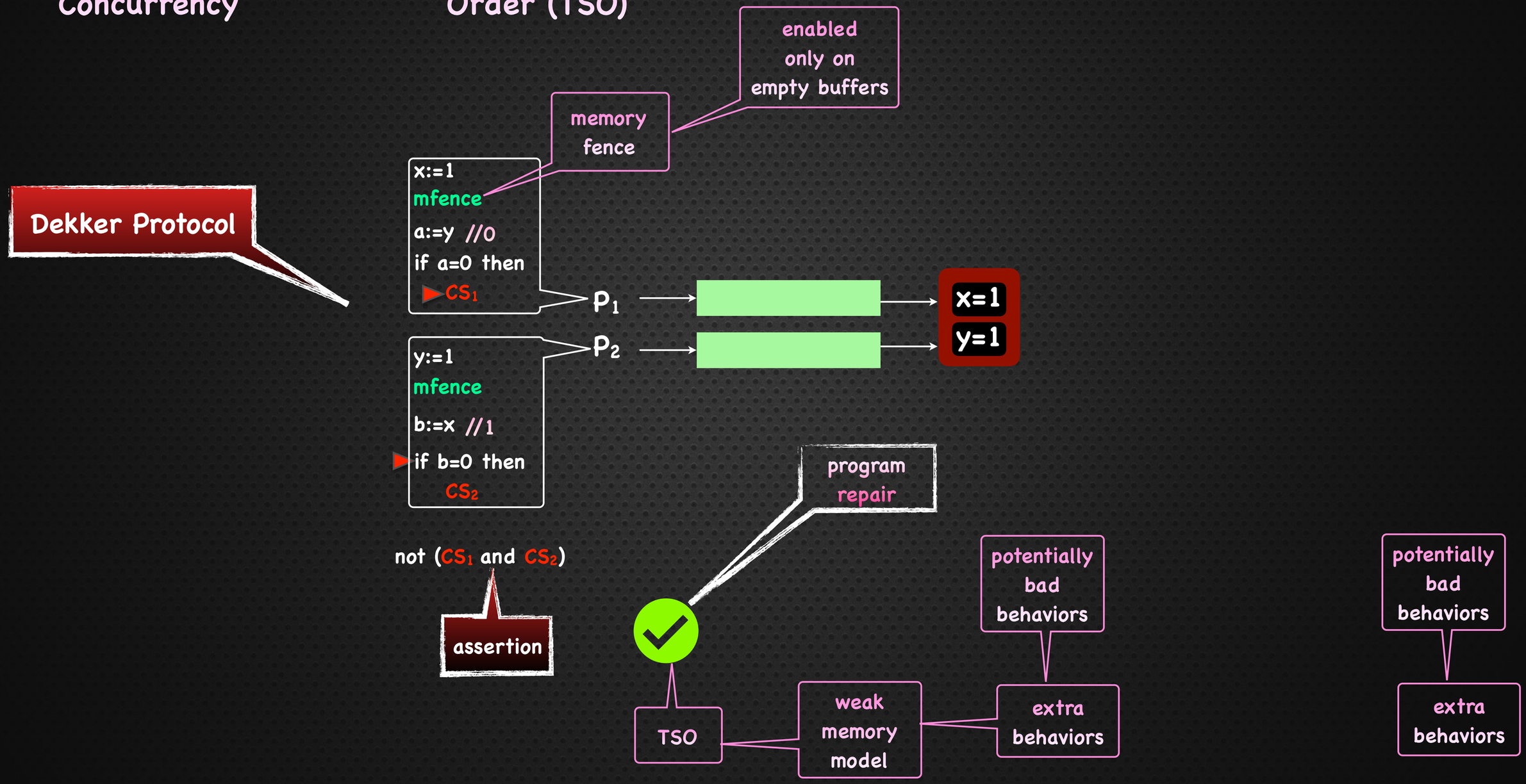




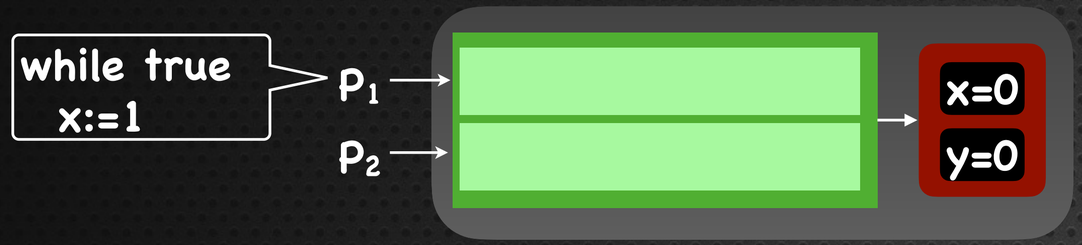
WALLPAPERSWIDE.COM

Shared Variable Concurrency

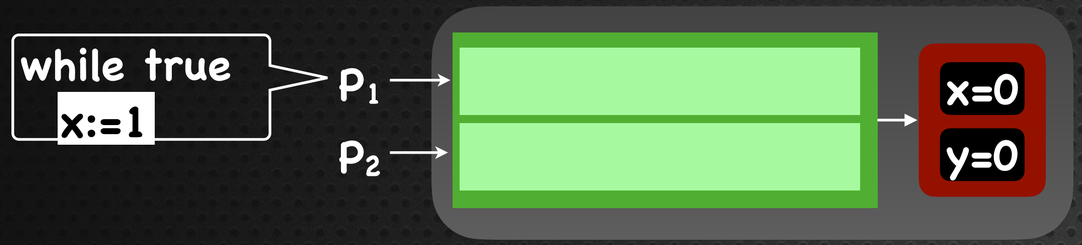
Total Store Order (TSO)



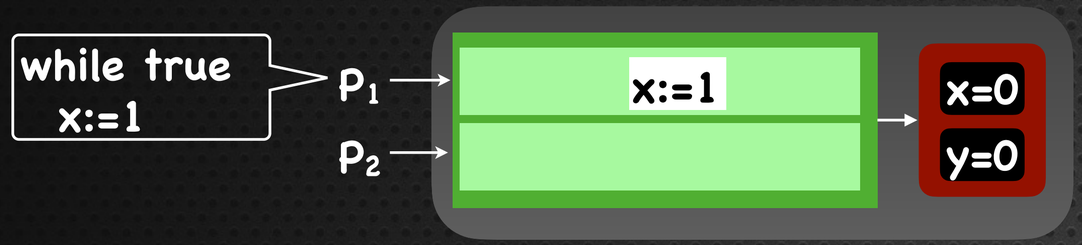
WALLPAPERSWIDE.COM



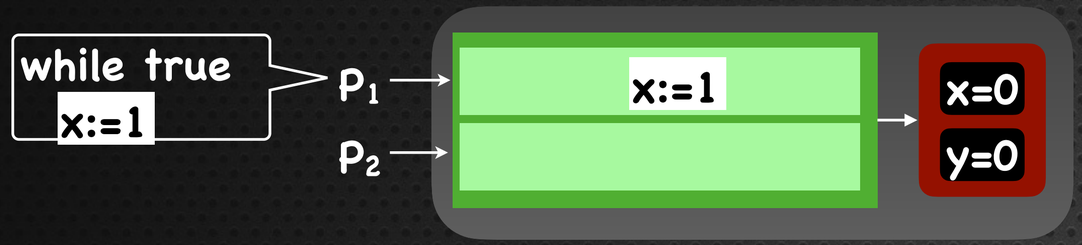
WALLPAPERSWIDE.COM



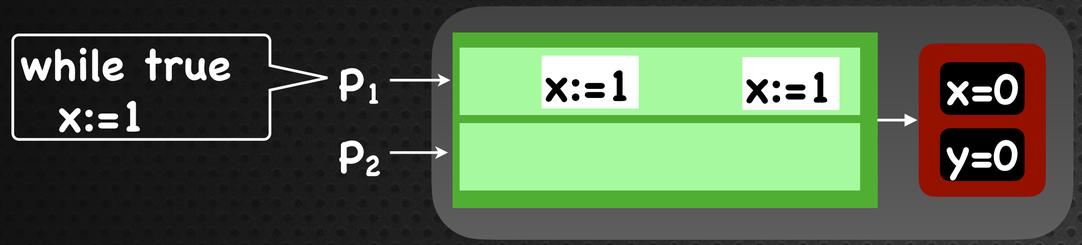
WALLPAPERSWIDE.COM



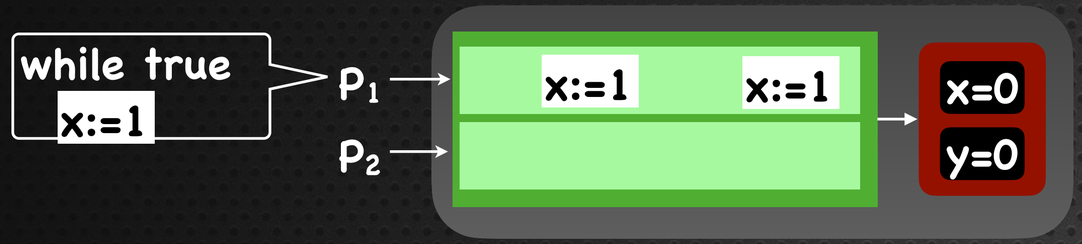
WALLPAPERSWIDE.COM



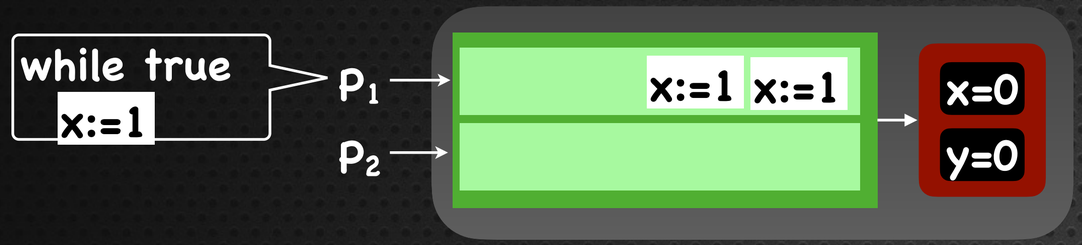
WALLPAPERSWIDE.COM



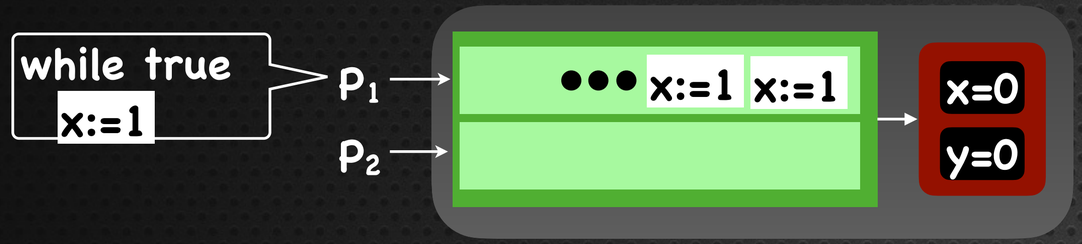
WALLPAPERSWIDE.COM



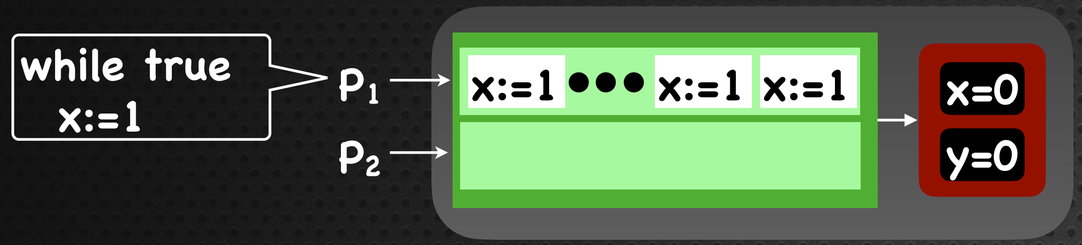
WALLPAPERSWIDE.COM



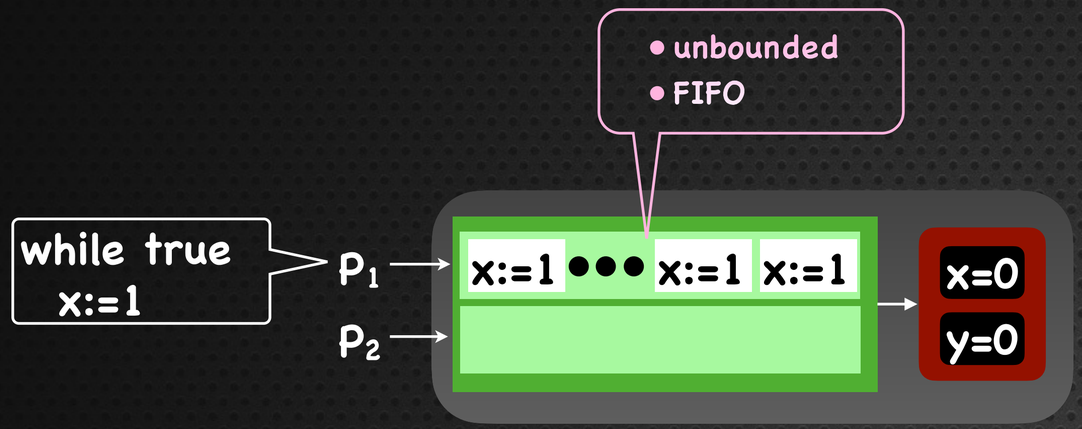
WALLPAPERSWIDE.COM



WALLPAPERSWIDE.COM



WALLPAPERSWIDE.COM



WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

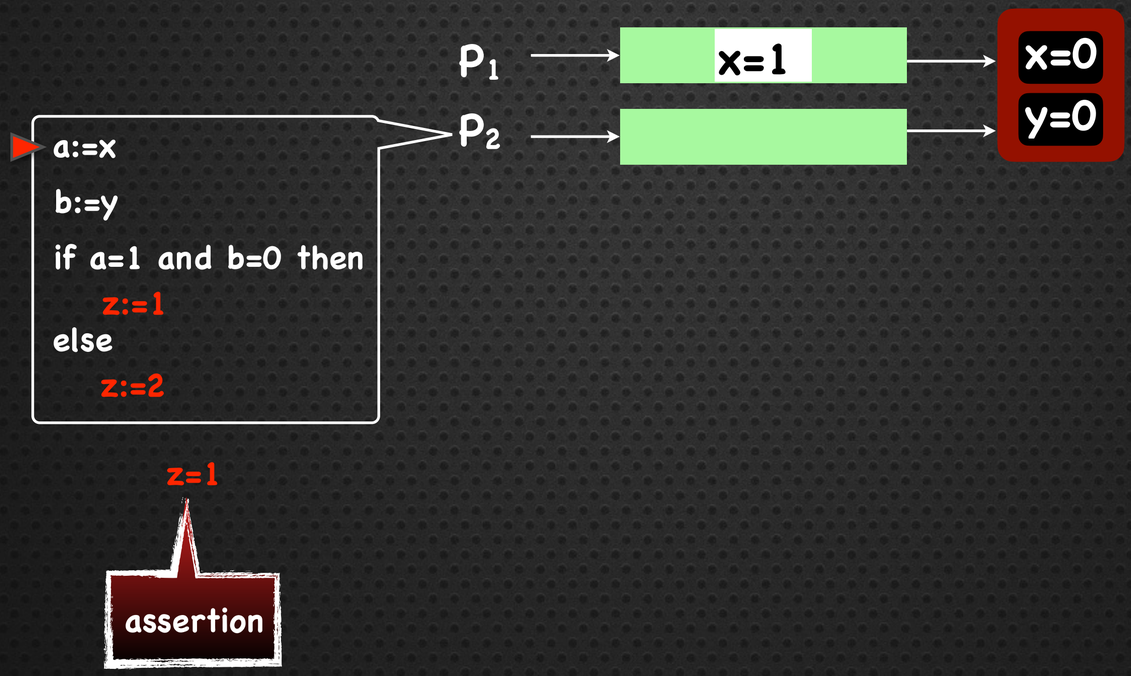
Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)



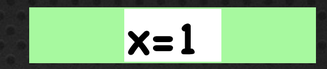
potentially bad behaviors

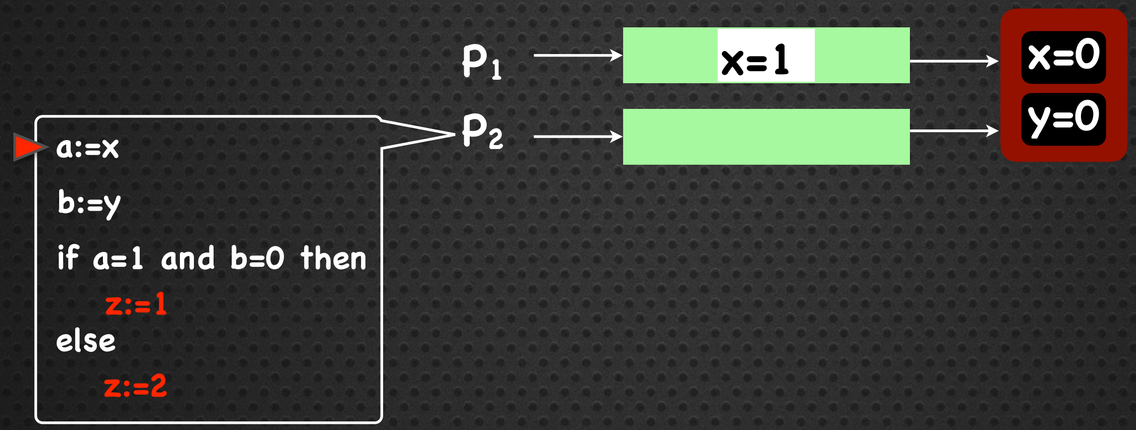
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







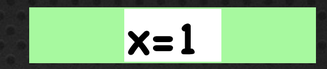
potentially bad behaviors

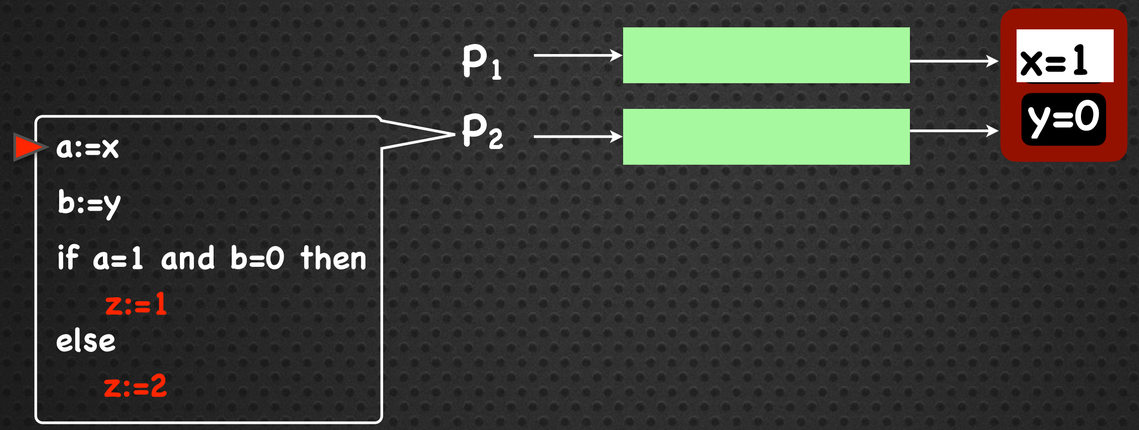
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







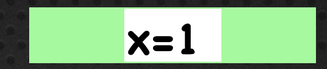
potentially bad behaviors

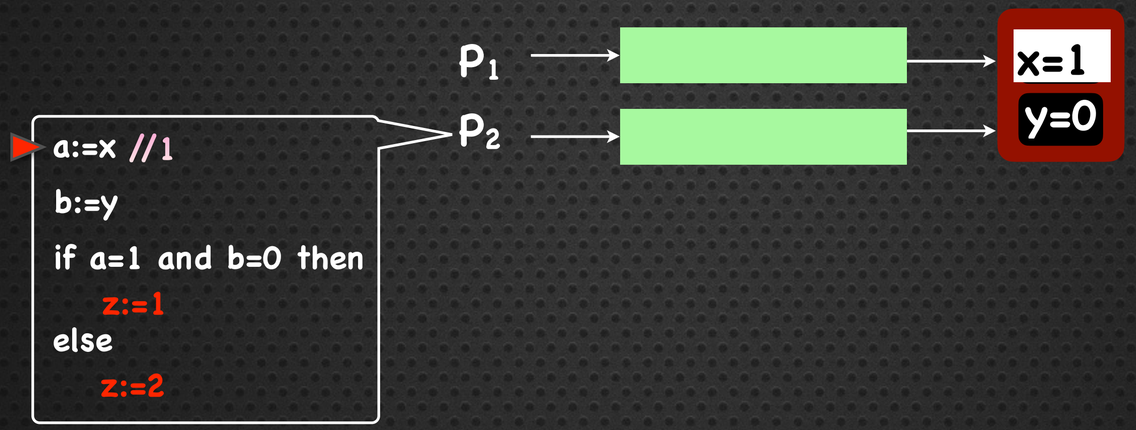
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







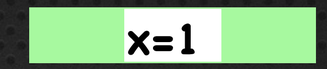
potentially bad behaviors

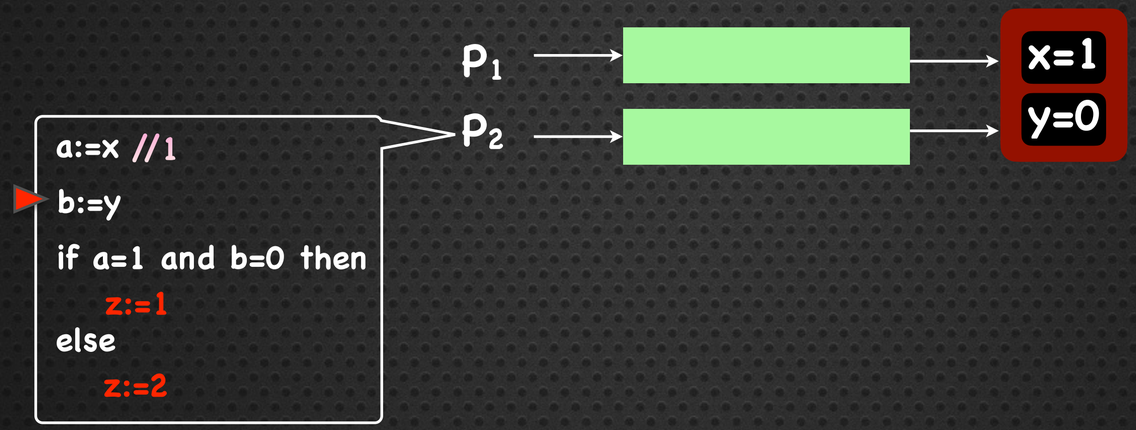
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







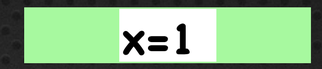
potentially bad behaviors

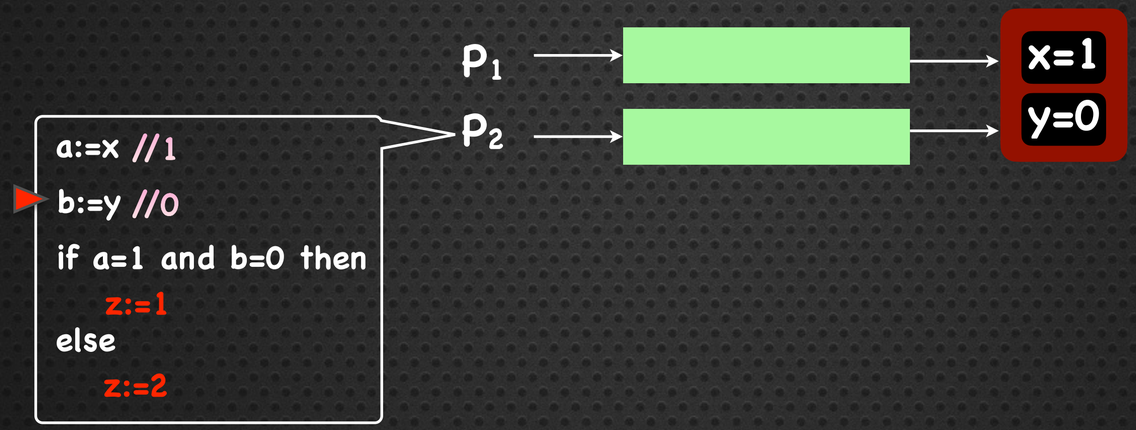
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







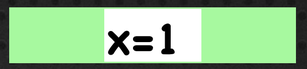
potentially bad behaviors

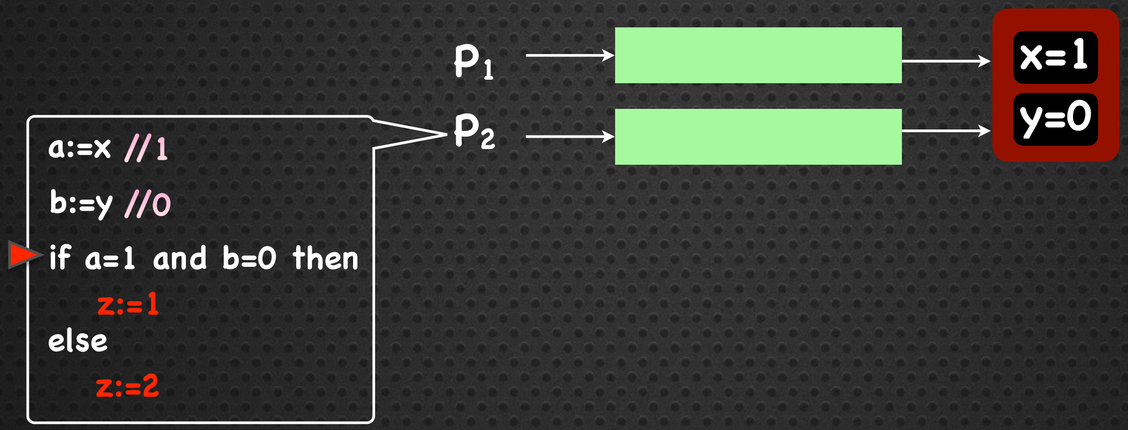
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







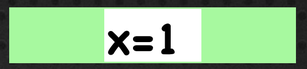
potentially bad behaviors

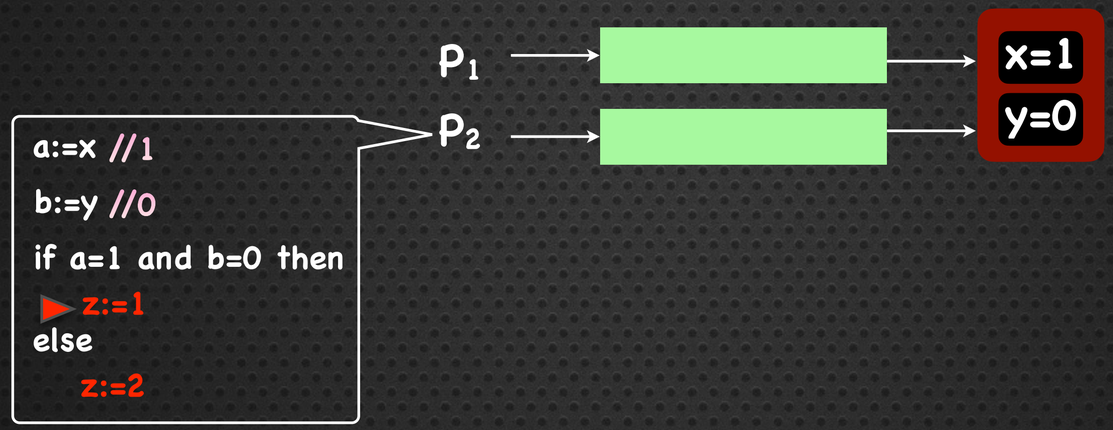
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







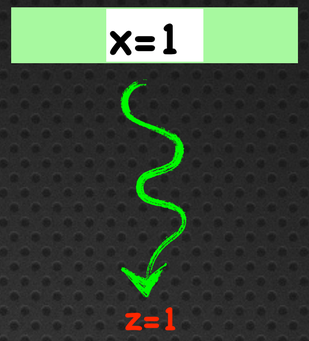
potentially bad behaviors

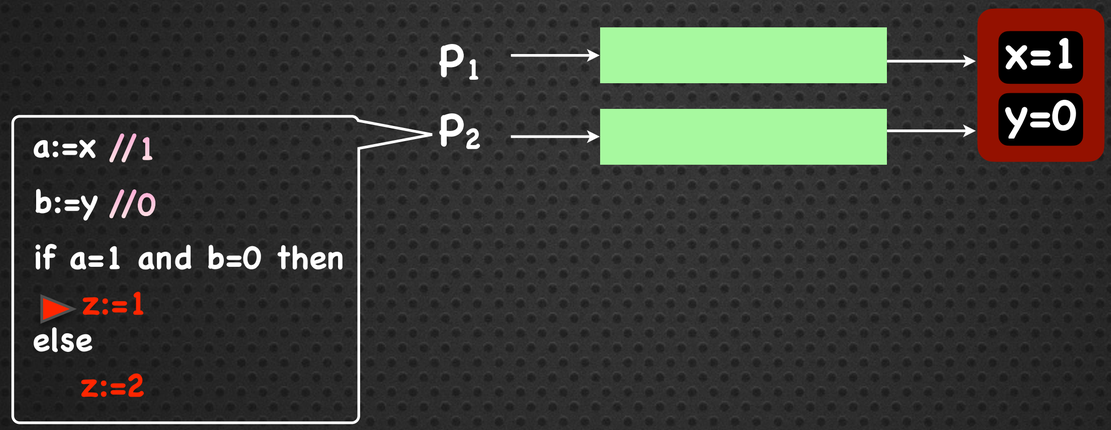
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







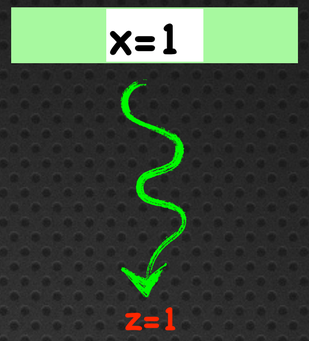
potentially bad behaviors

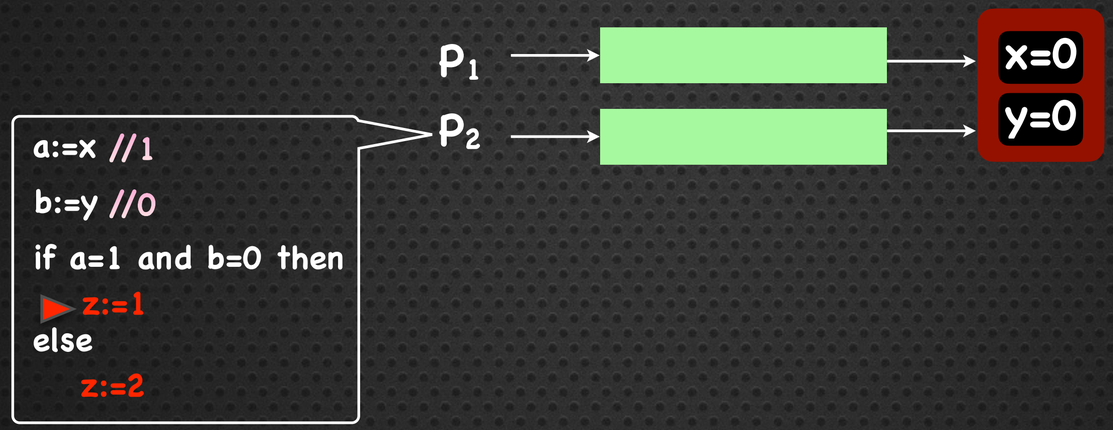
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







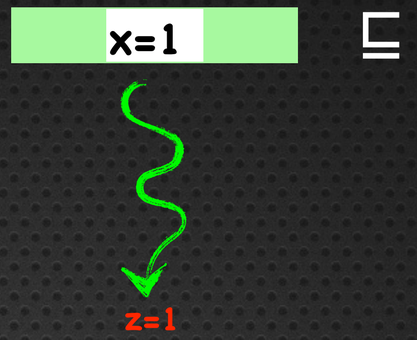
potentially bad behaviors

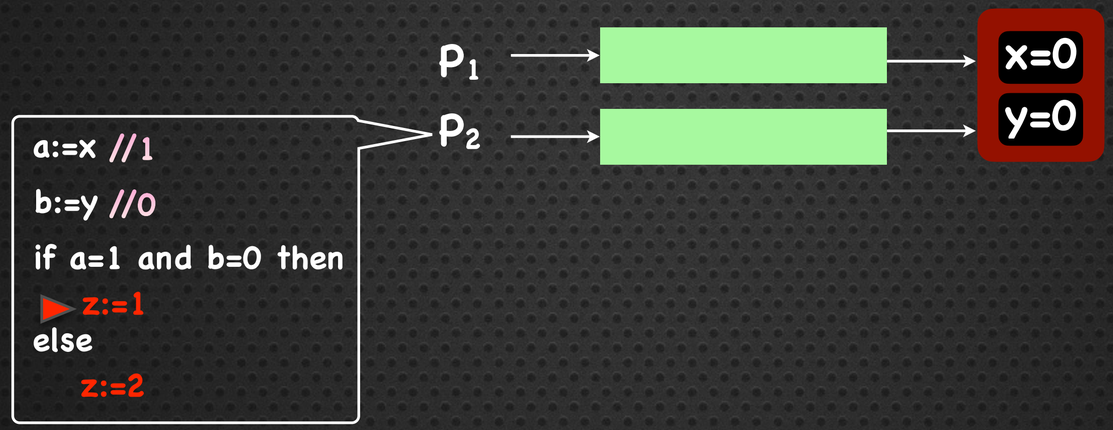
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







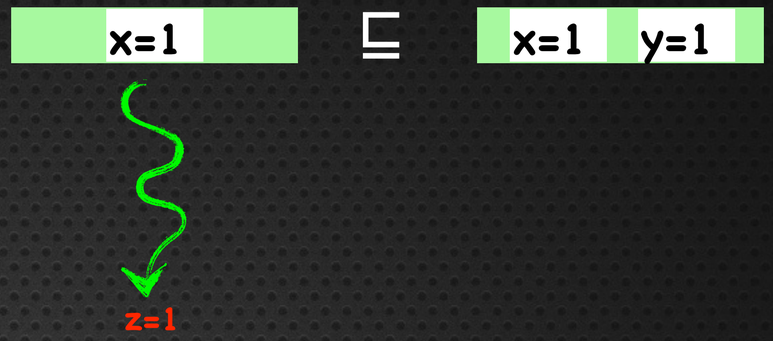
potentially bad behaviors

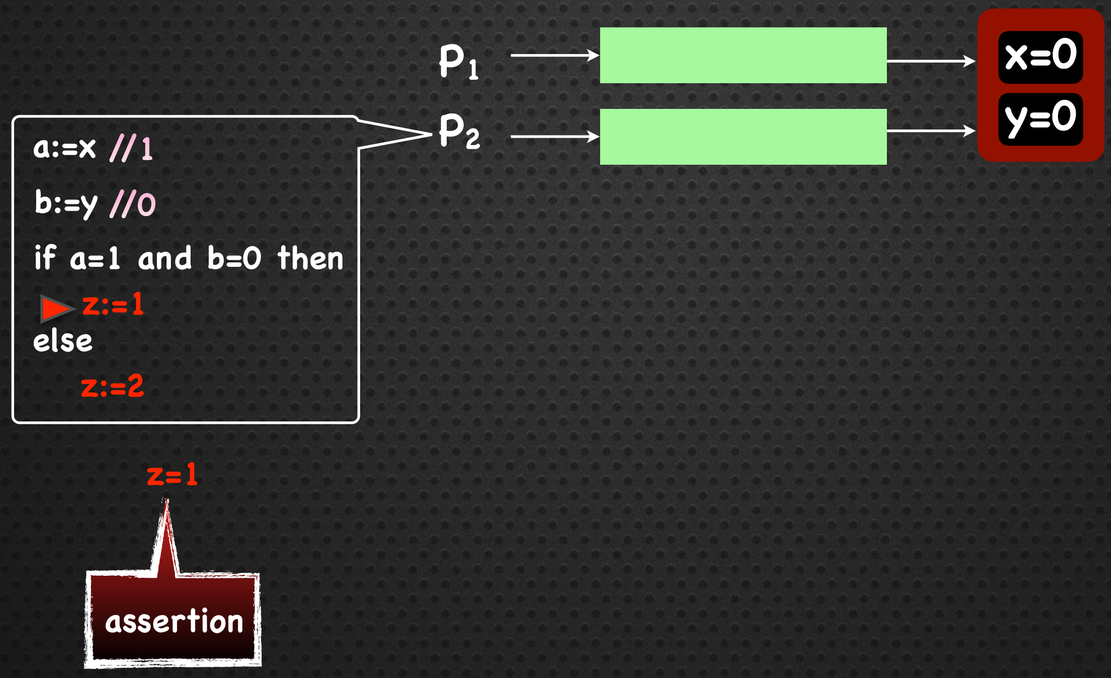
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





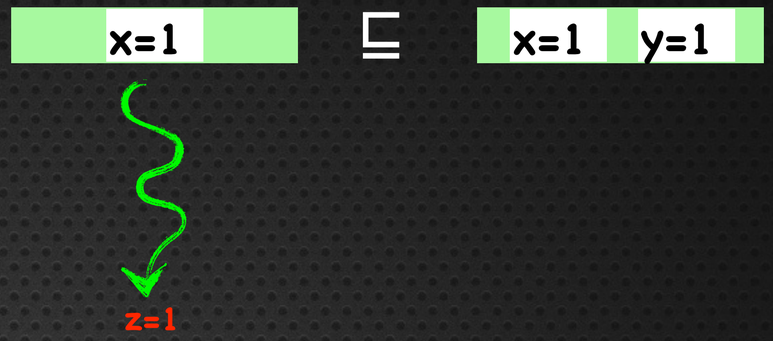
potentially bad behaviors

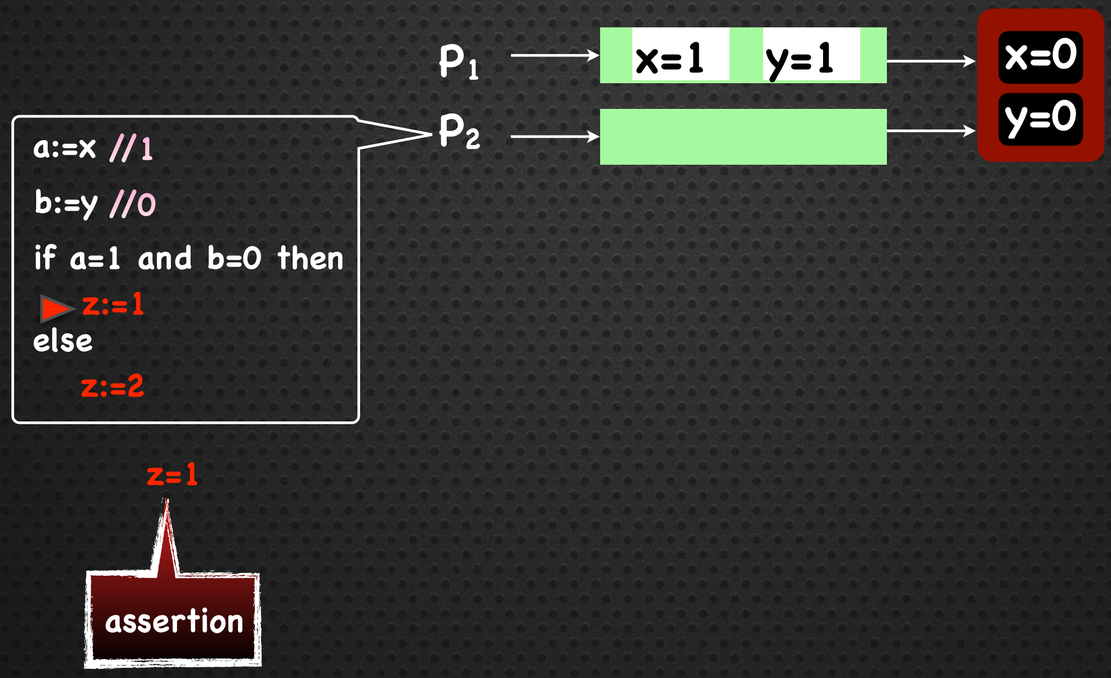
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





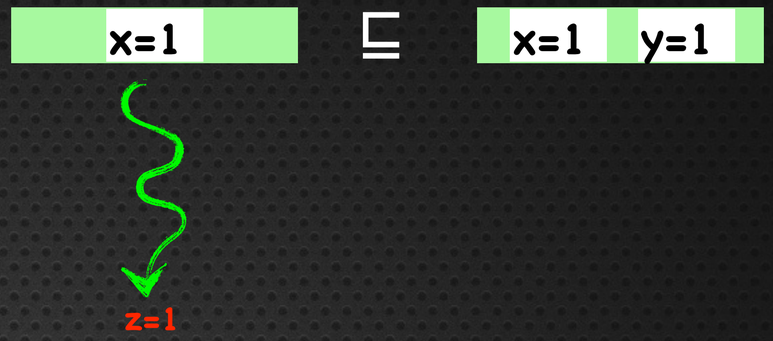
potentially bad behaviors

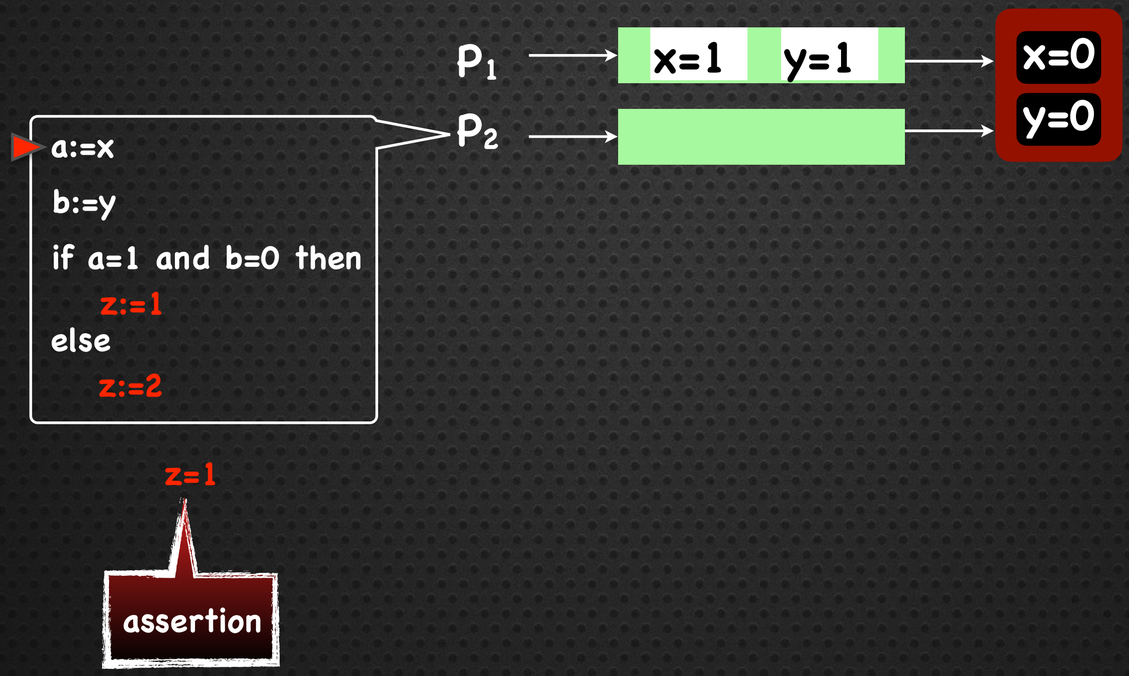
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





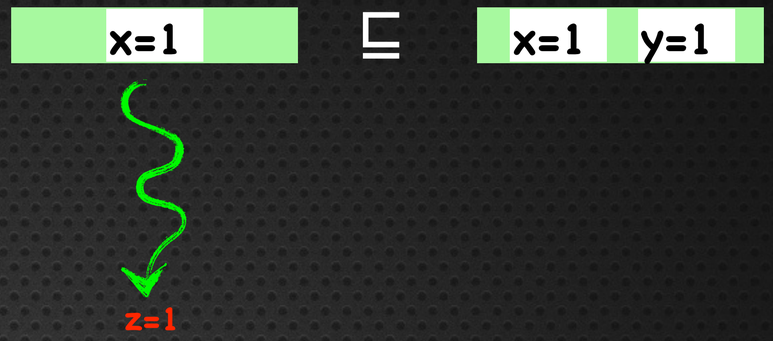
potentially bad behaviors

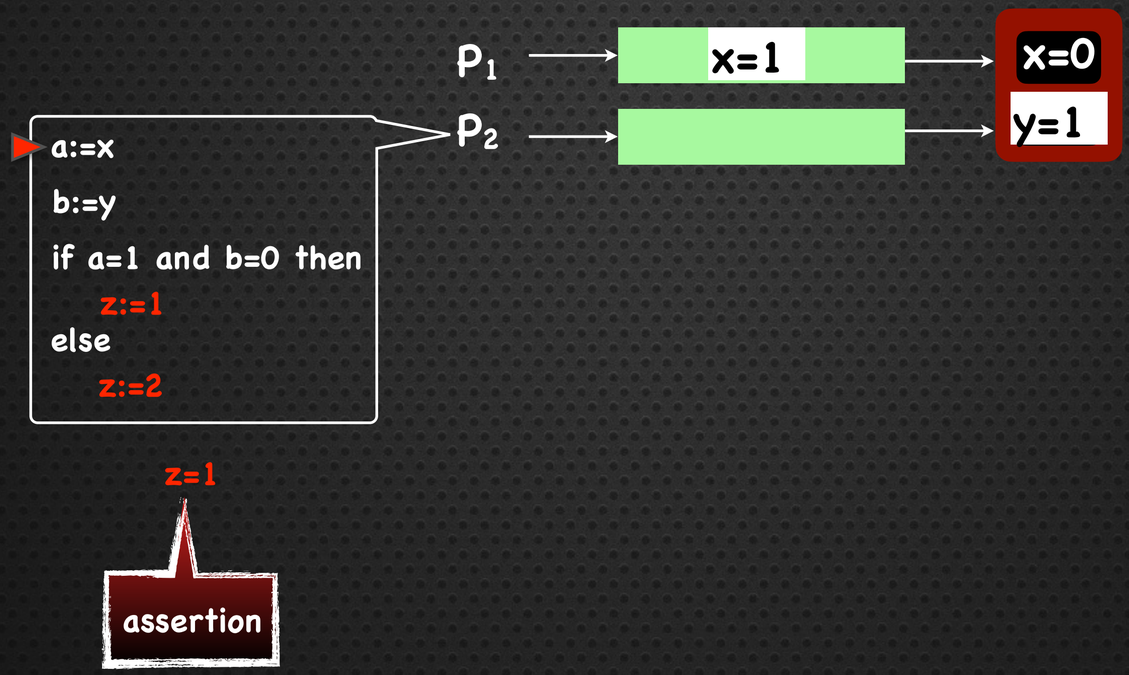
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





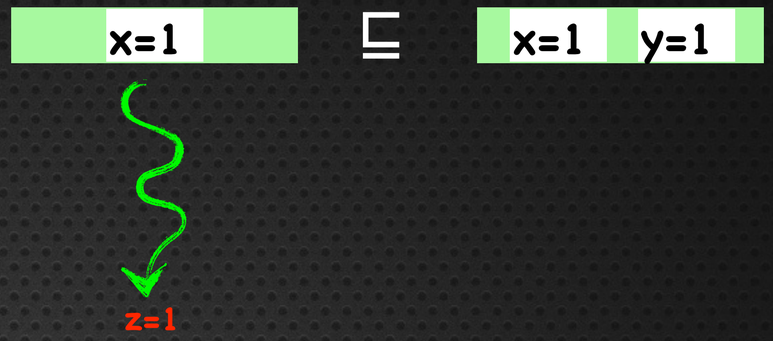
potentially bad behaviors

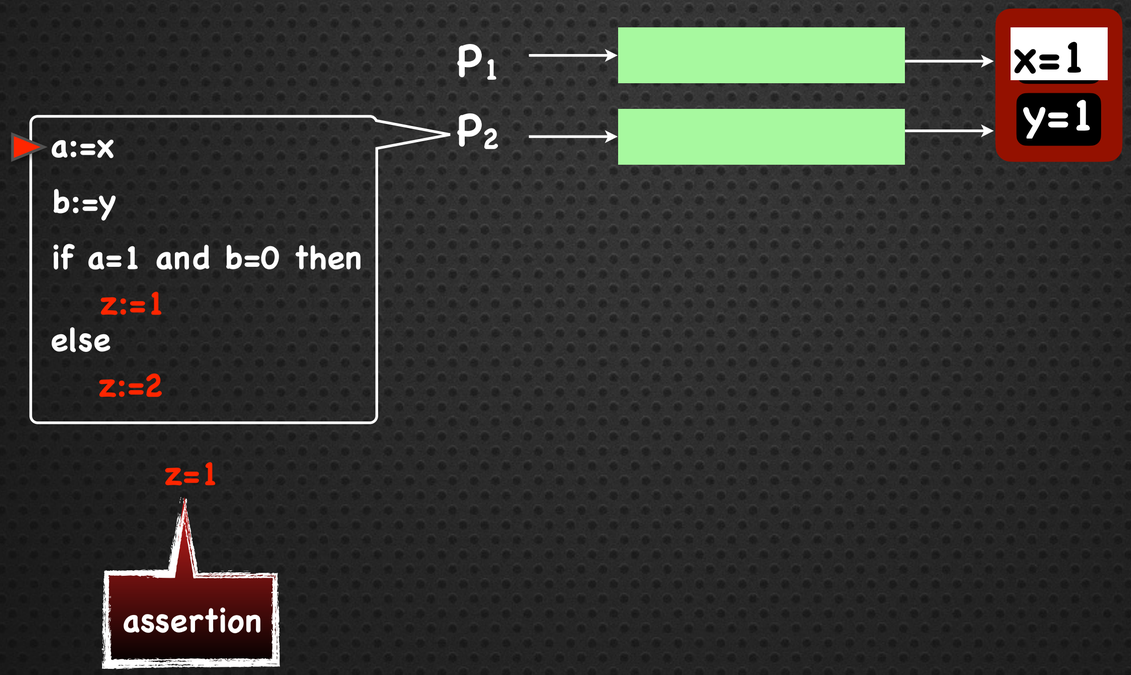
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





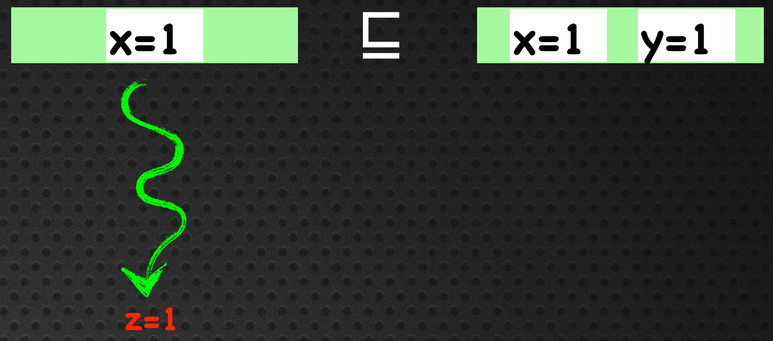
potentially bad behaviors

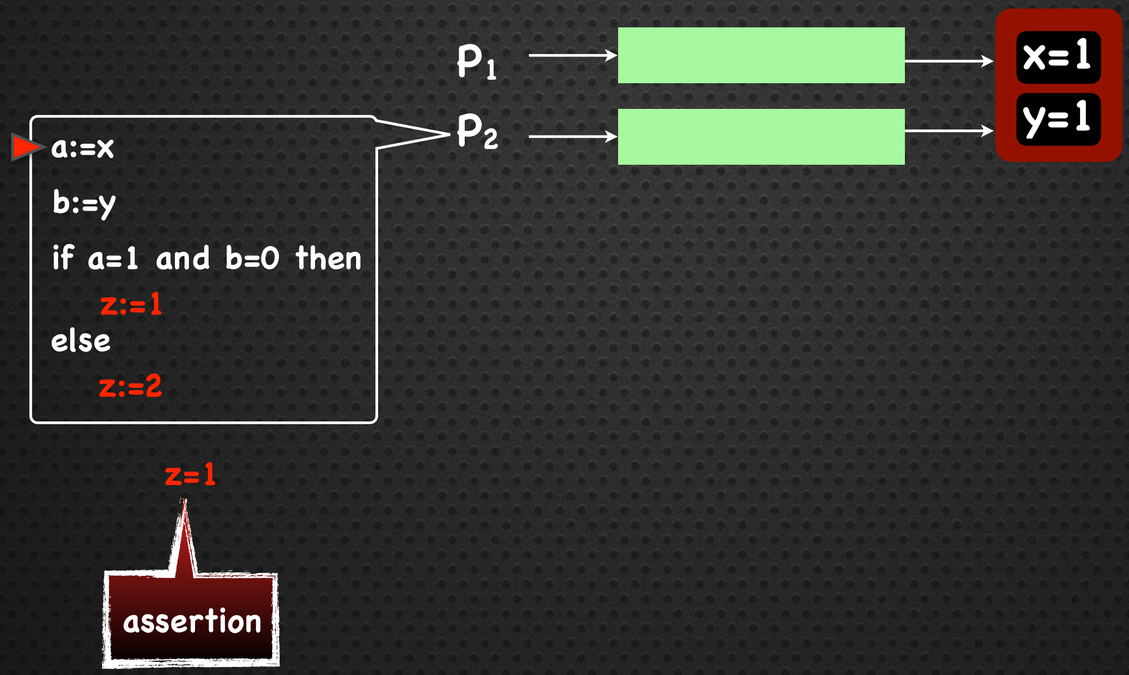
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





potentially bad behaviors

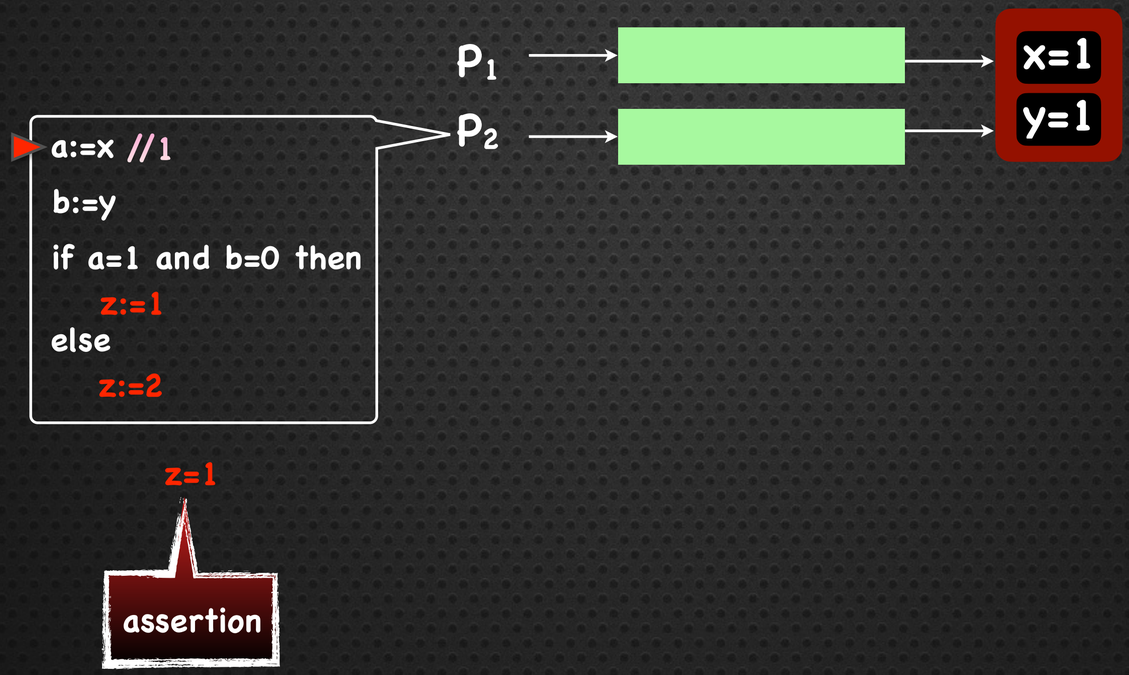
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





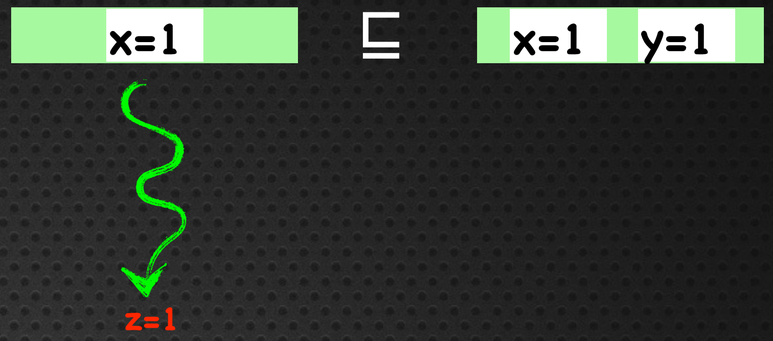
potentially bad behaviors

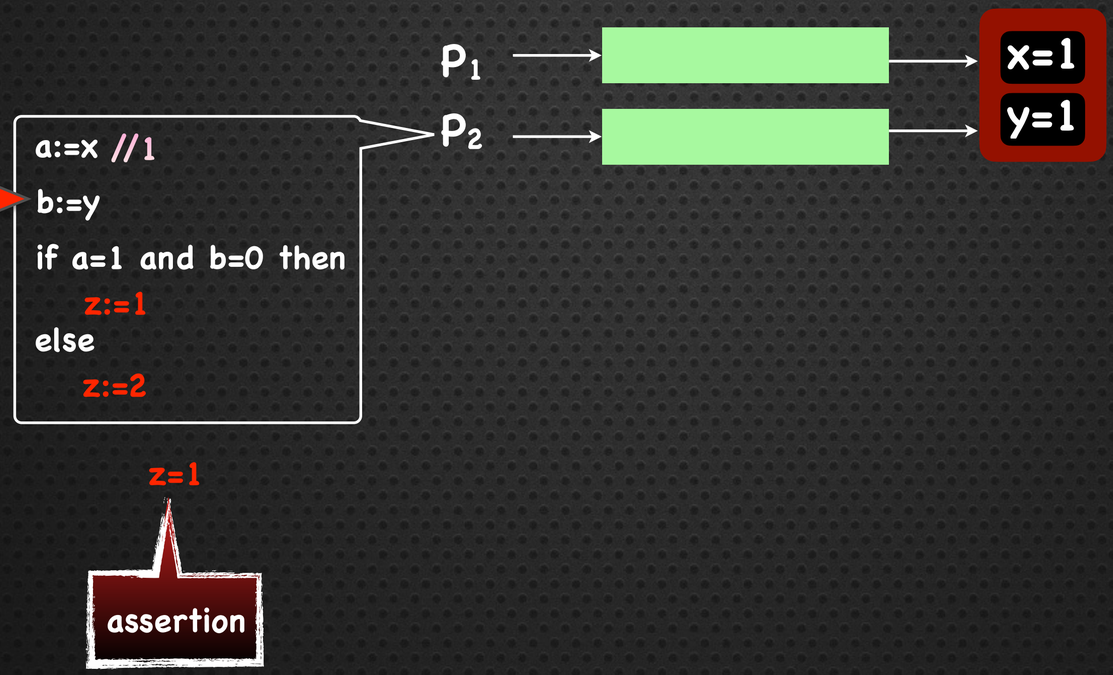
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





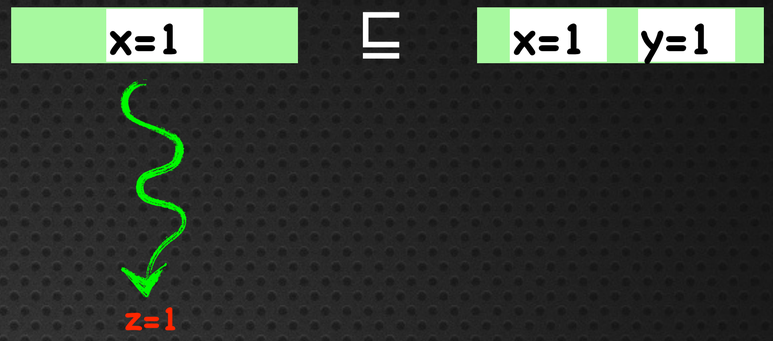
potentially bad behaviors

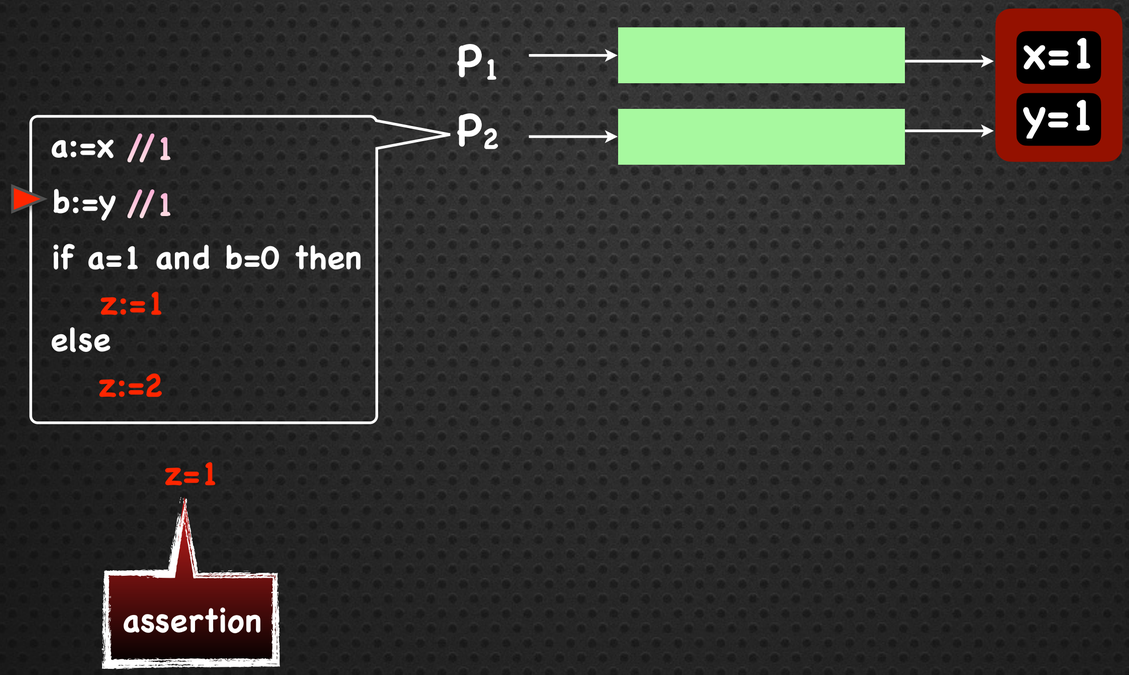
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





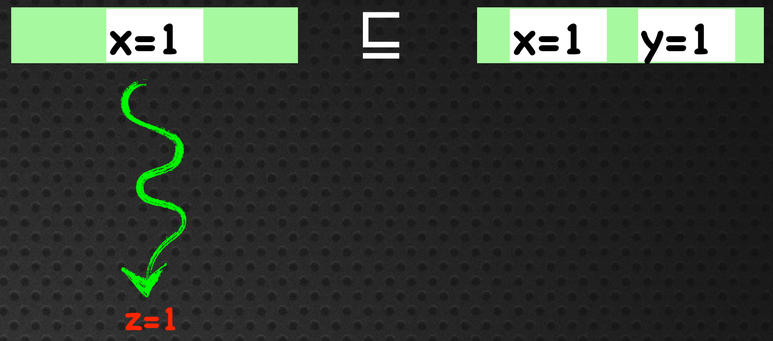
potentially bad behaviors

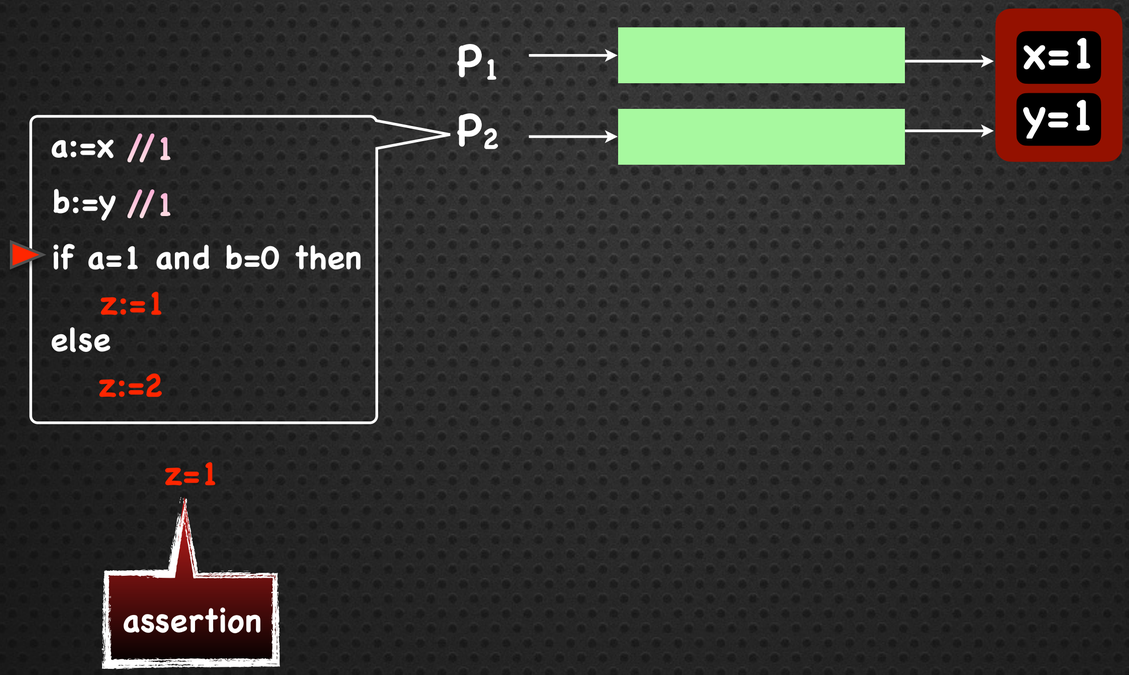
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





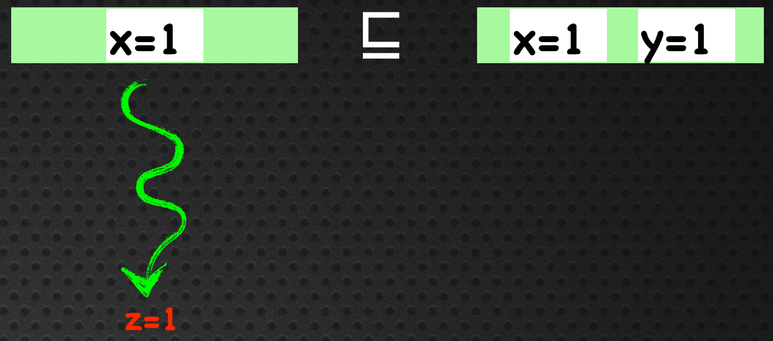
potentially bad behaviors

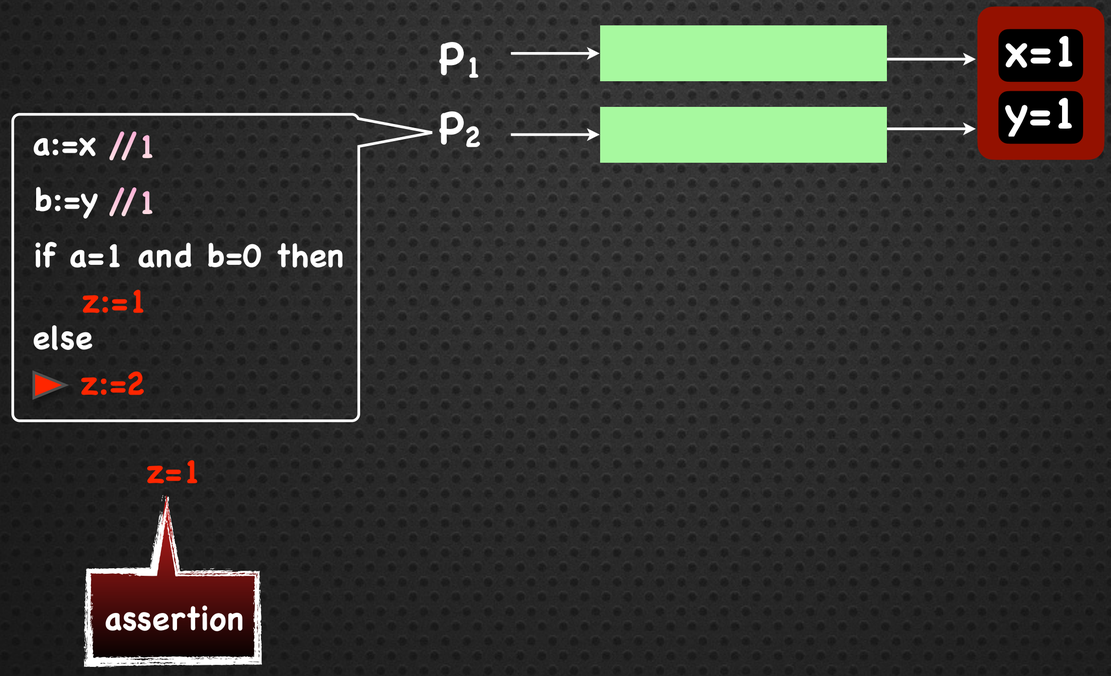
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)





potentially bad behaviors

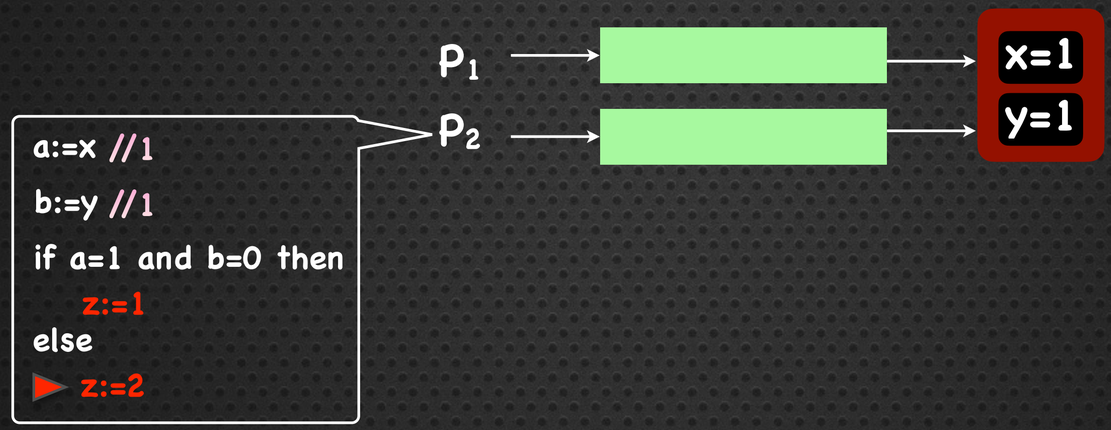
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)







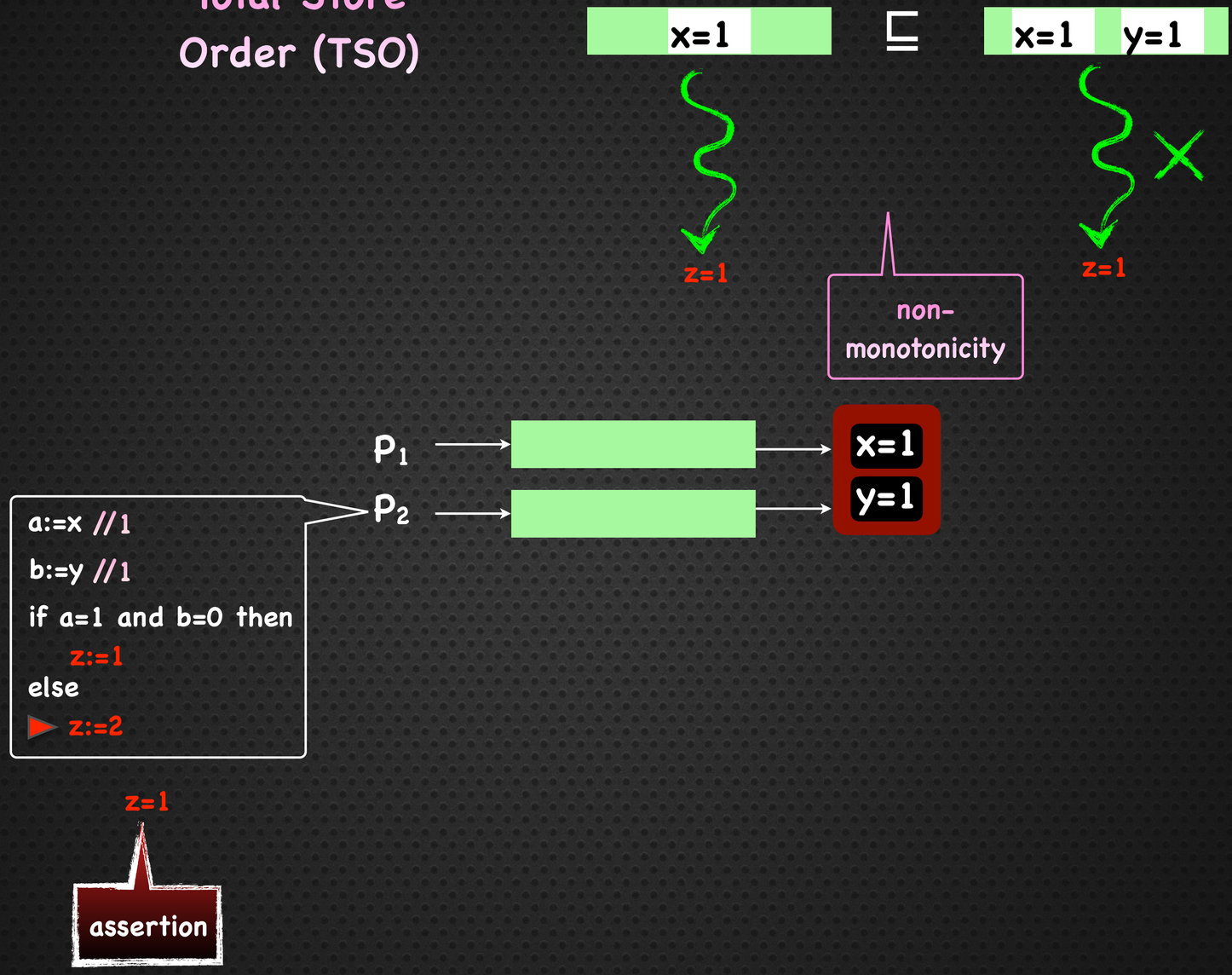
potentially bad behaviors

extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)



WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

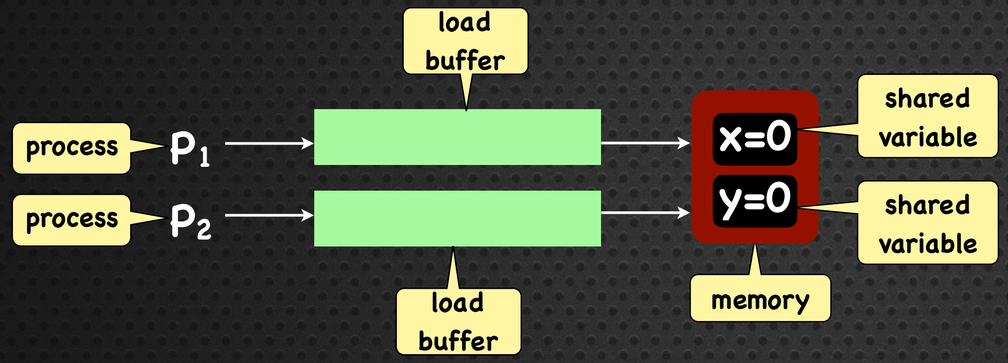
Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

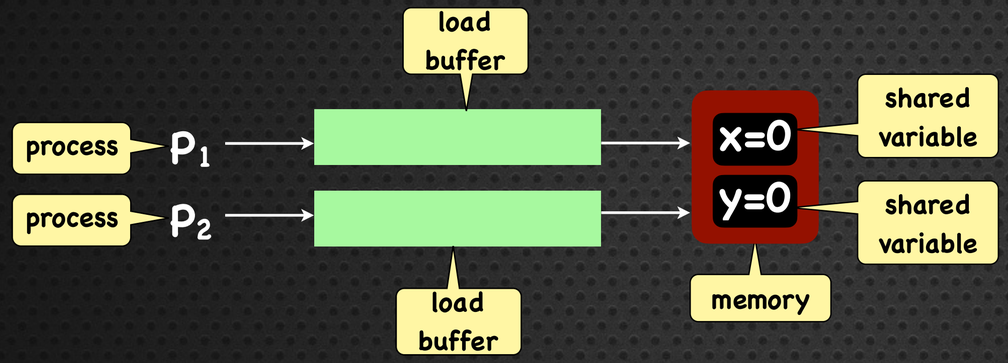


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

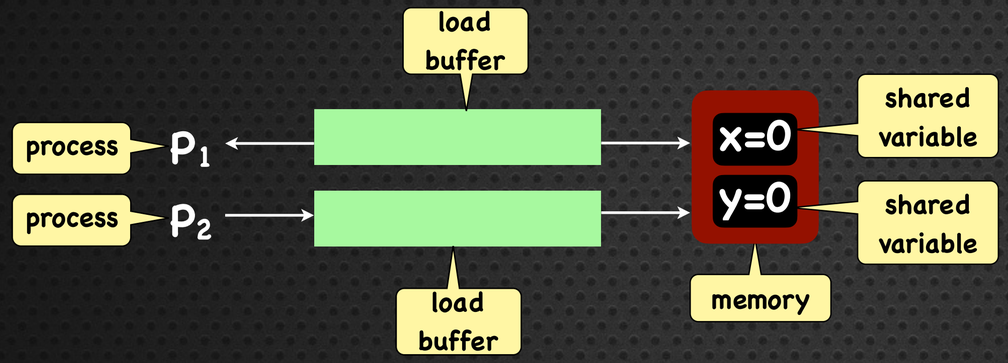


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

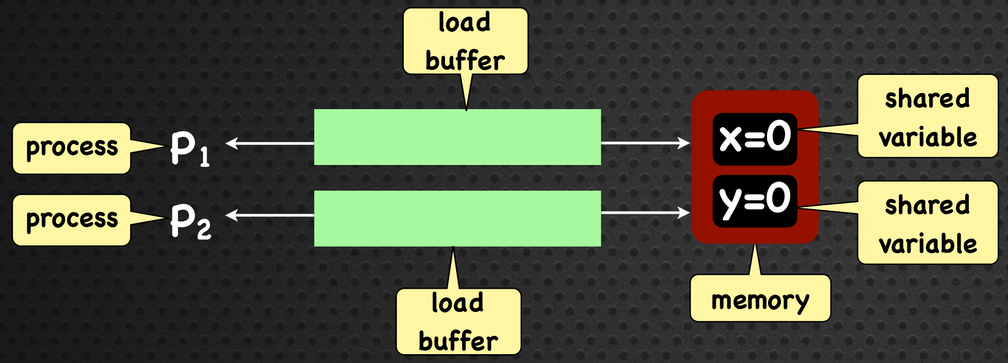


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

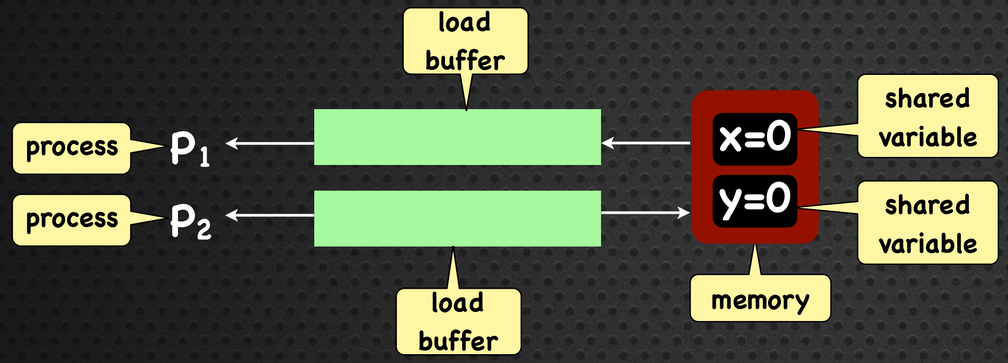


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

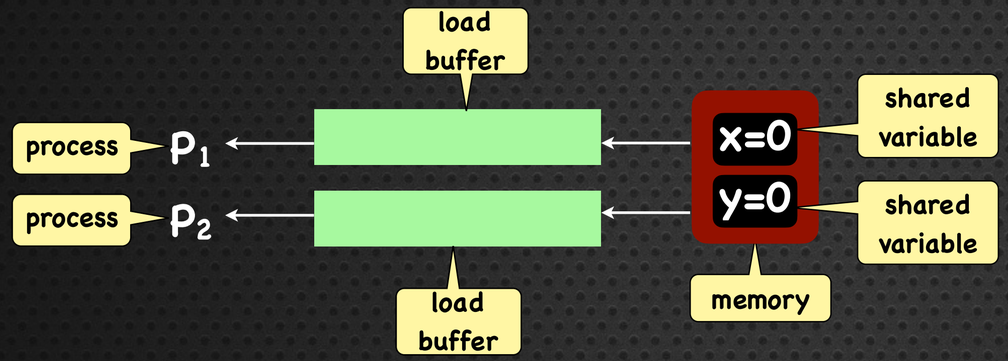


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

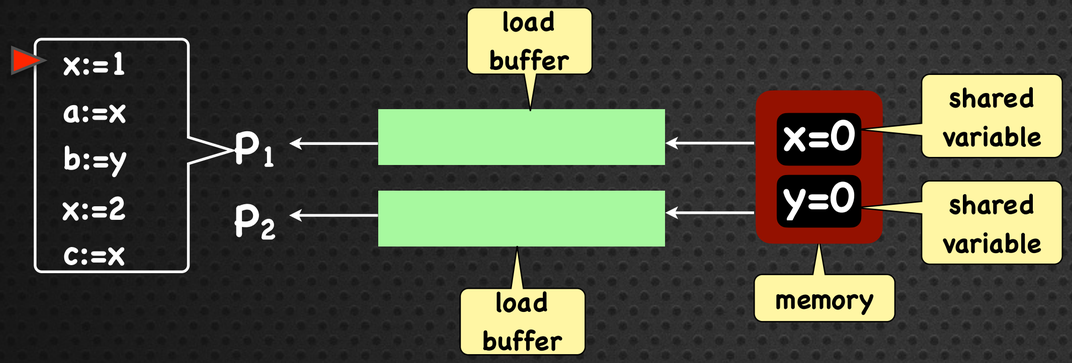


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

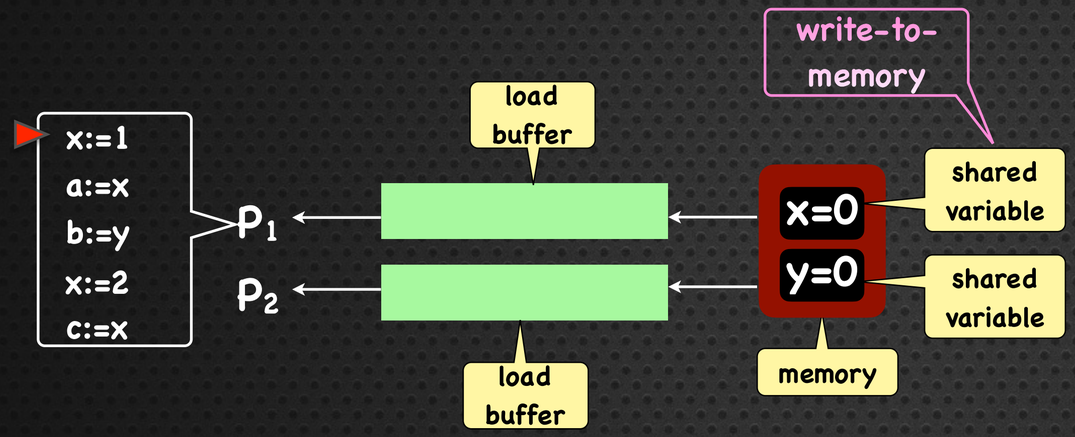


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

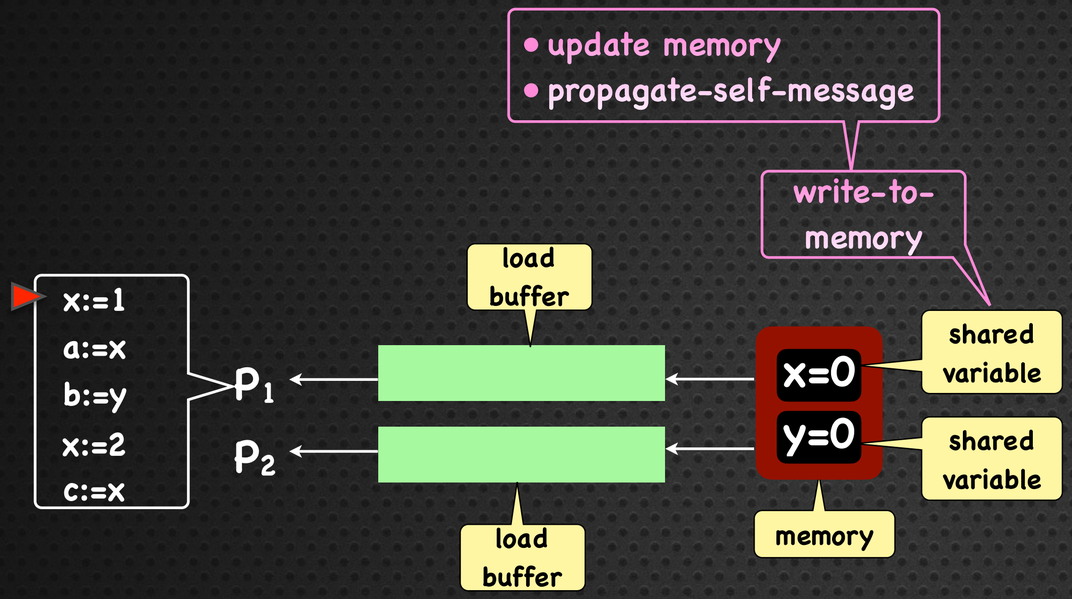


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

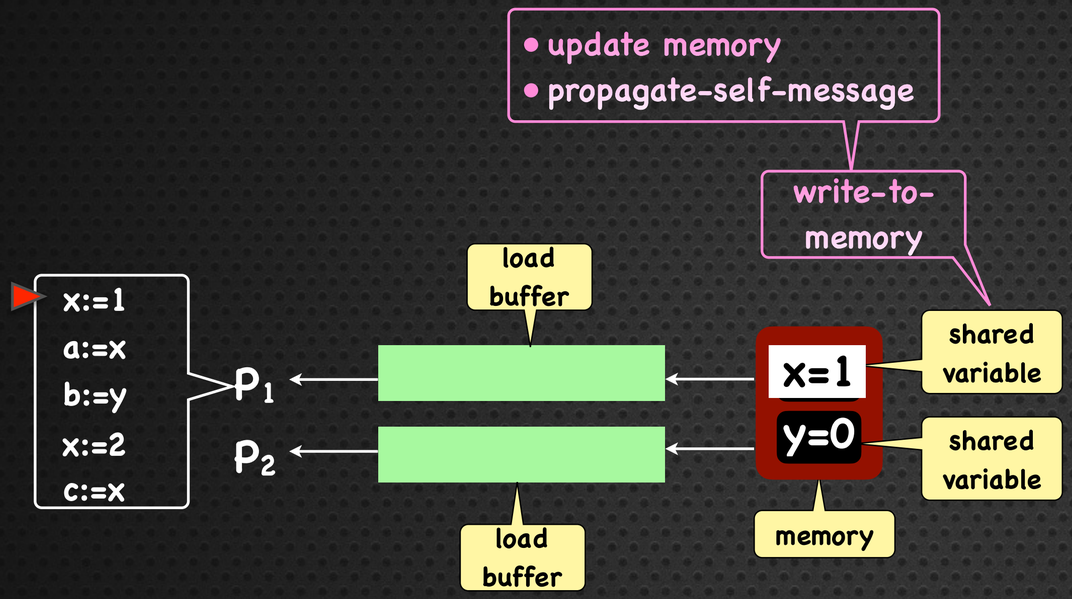


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

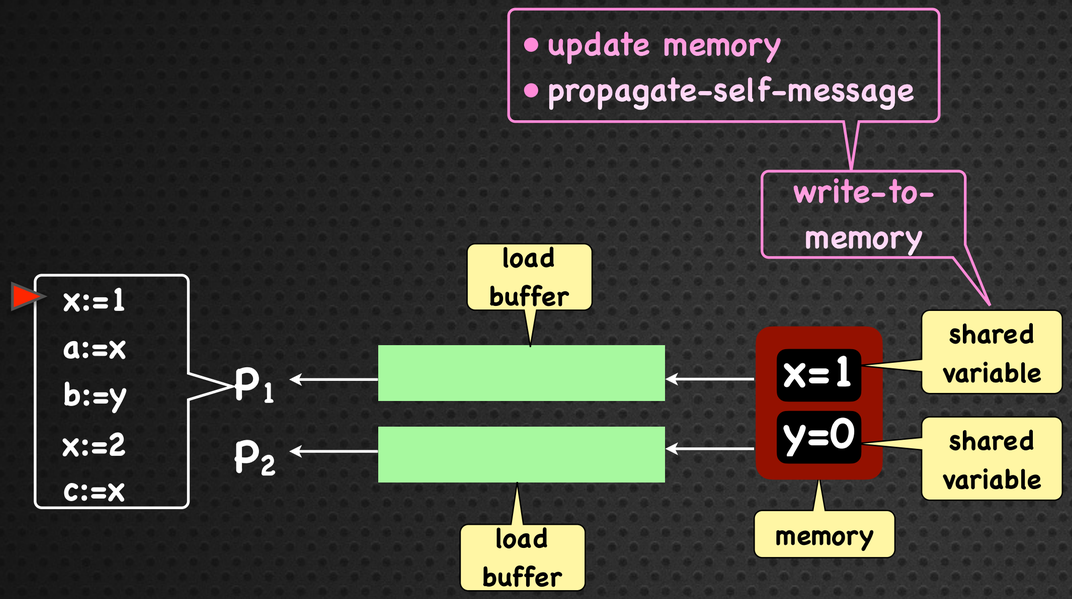


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

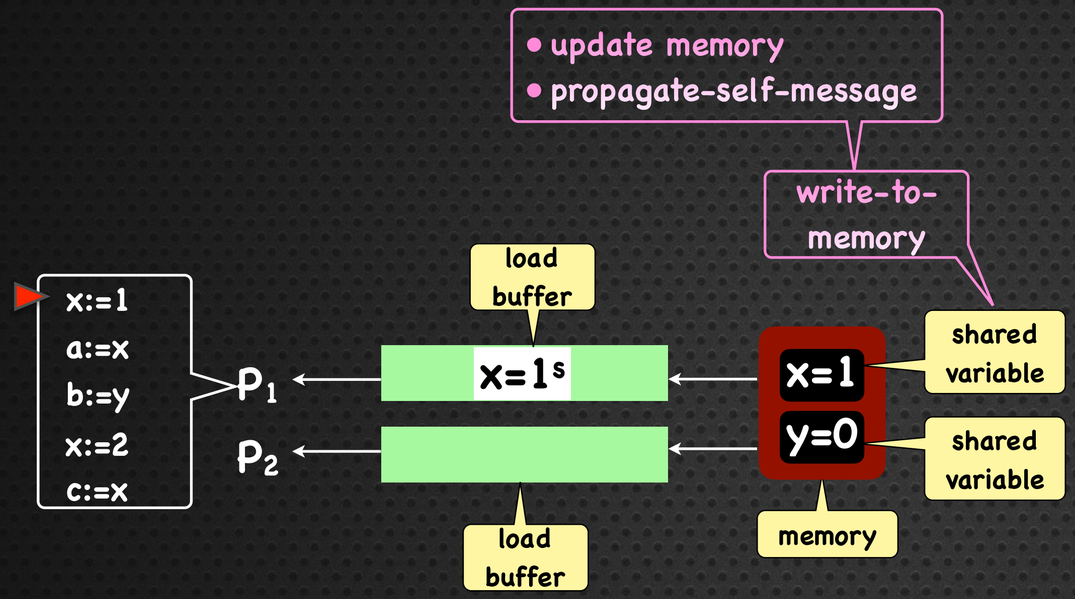


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

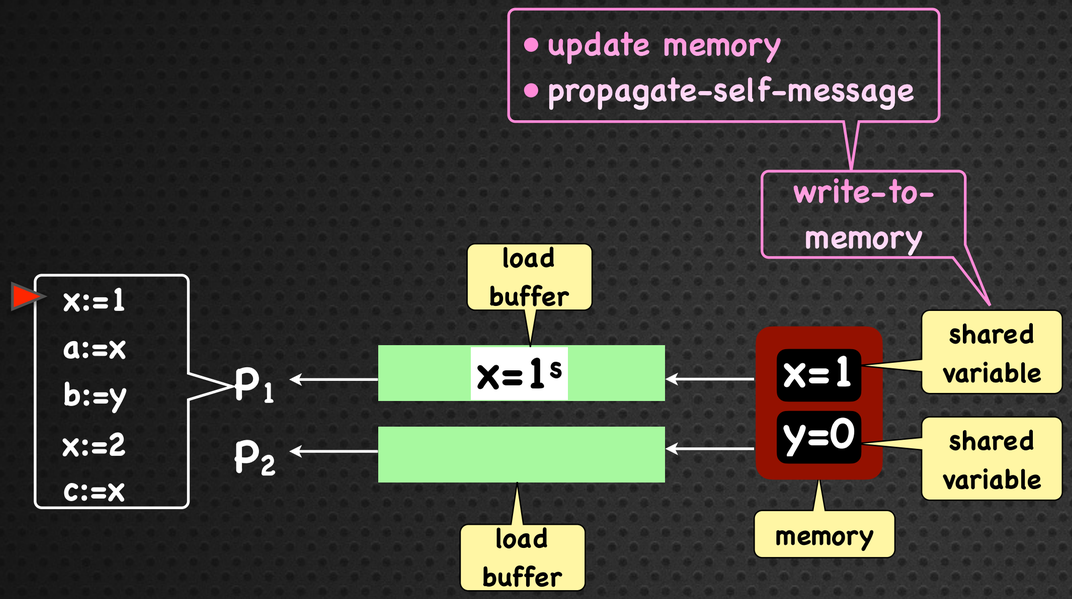


WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



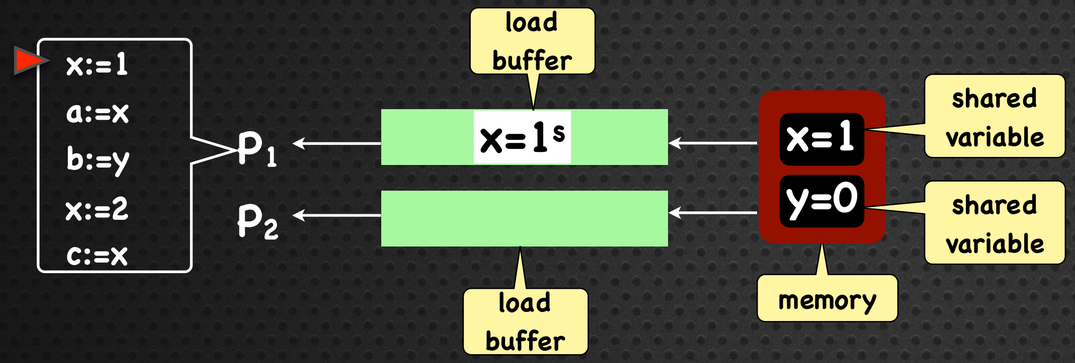
- write-to-memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



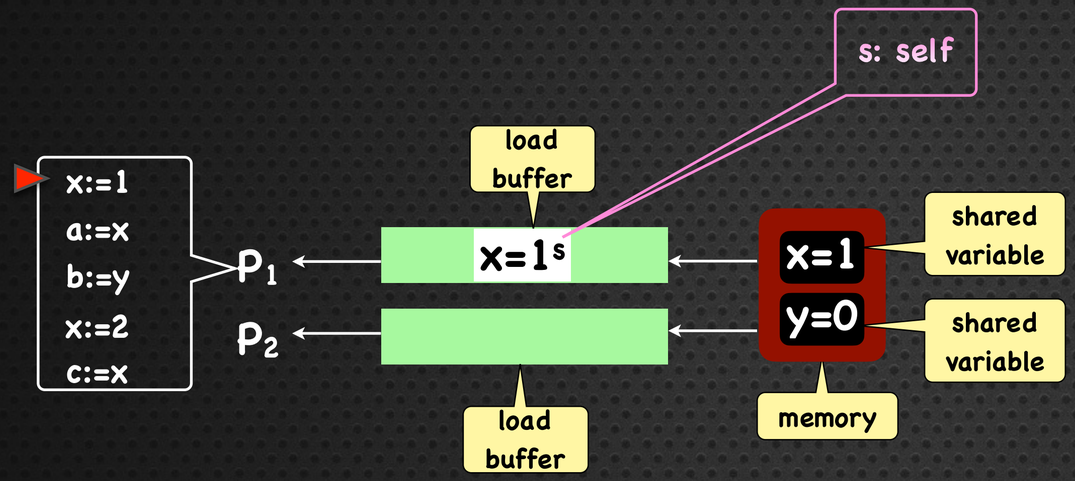
- write-to-memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



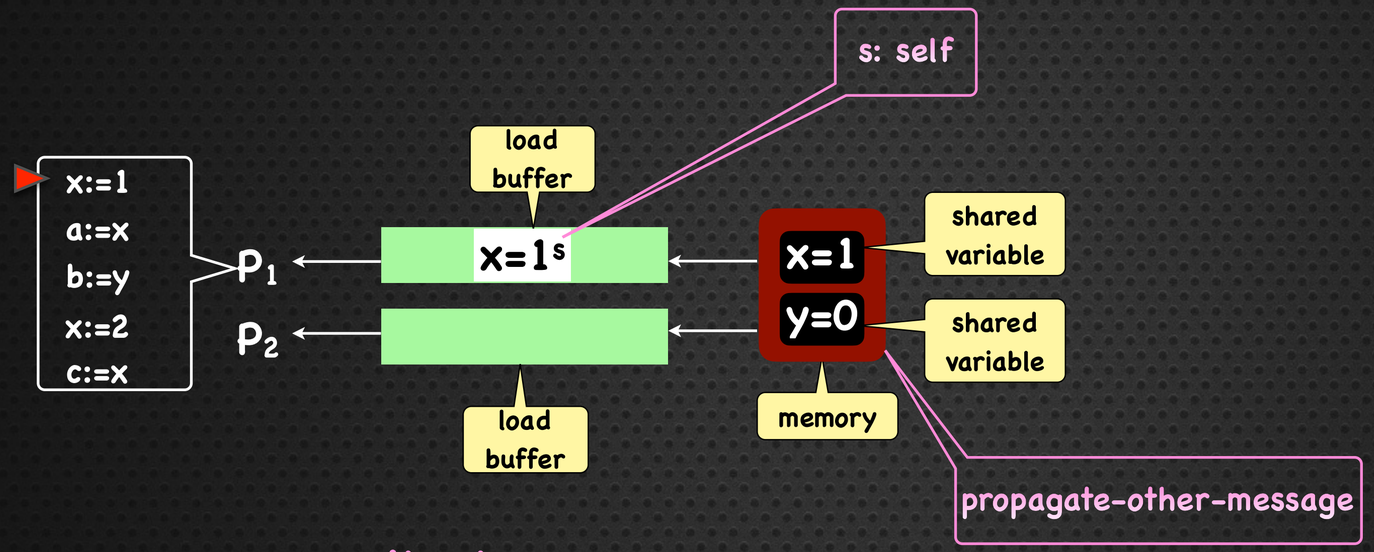
- write-to-memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



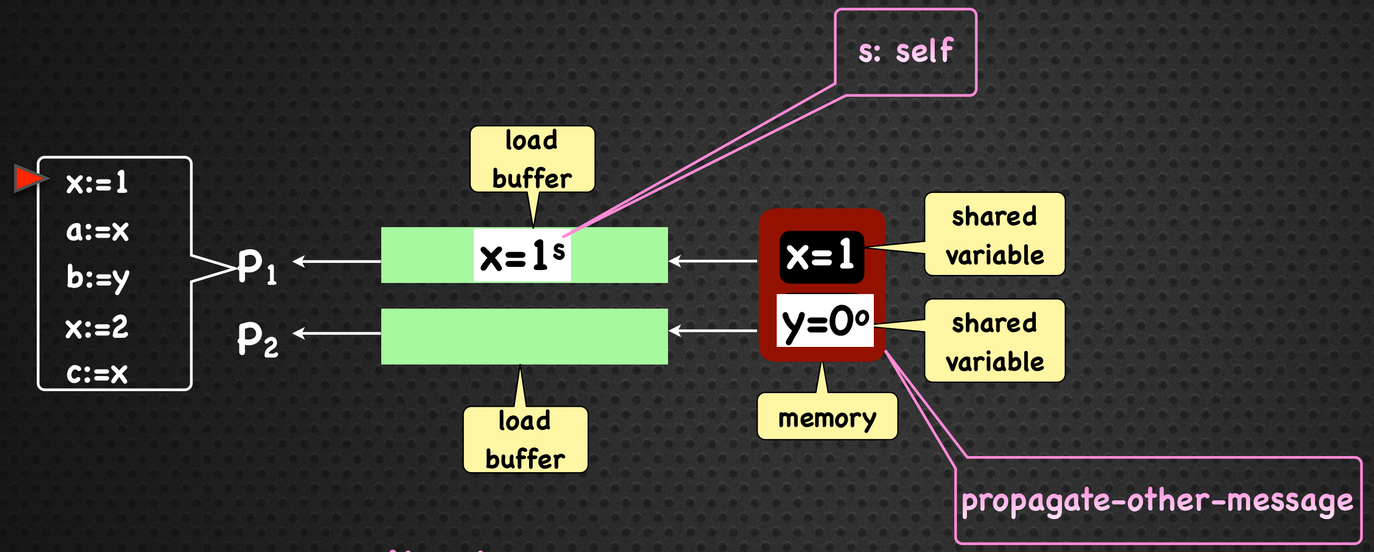
- write-to-memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

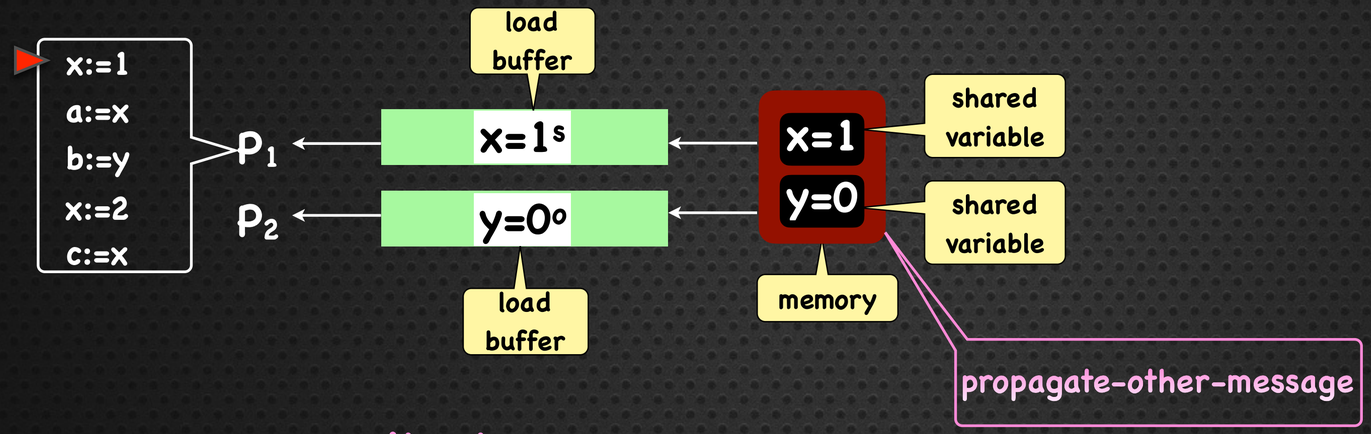


- write-to-memory

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



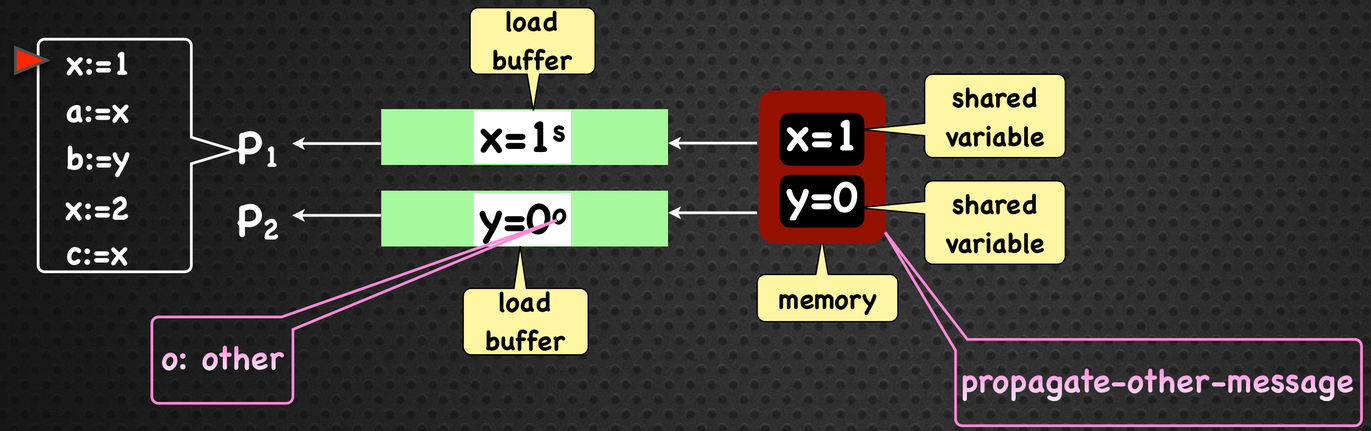
- write-to-memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



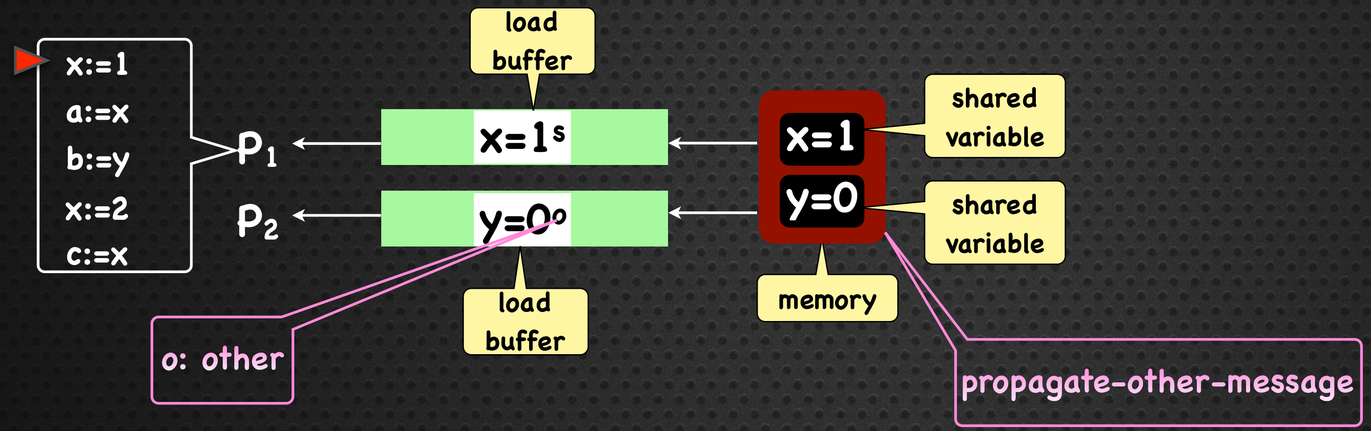
- write-to-memory

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



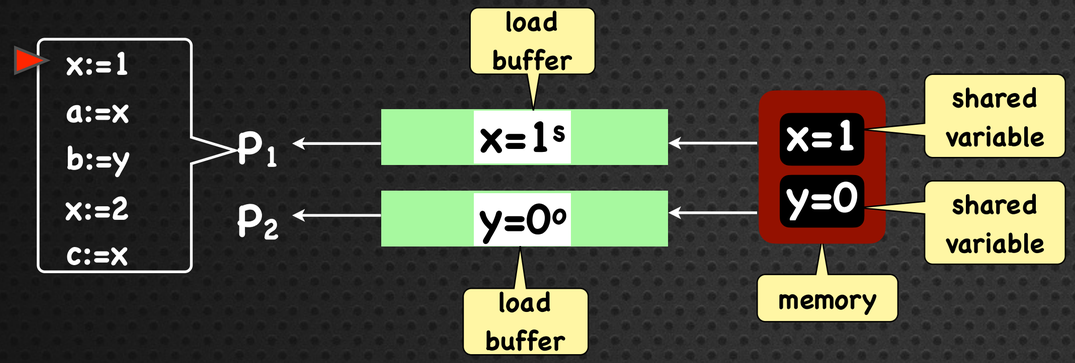
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



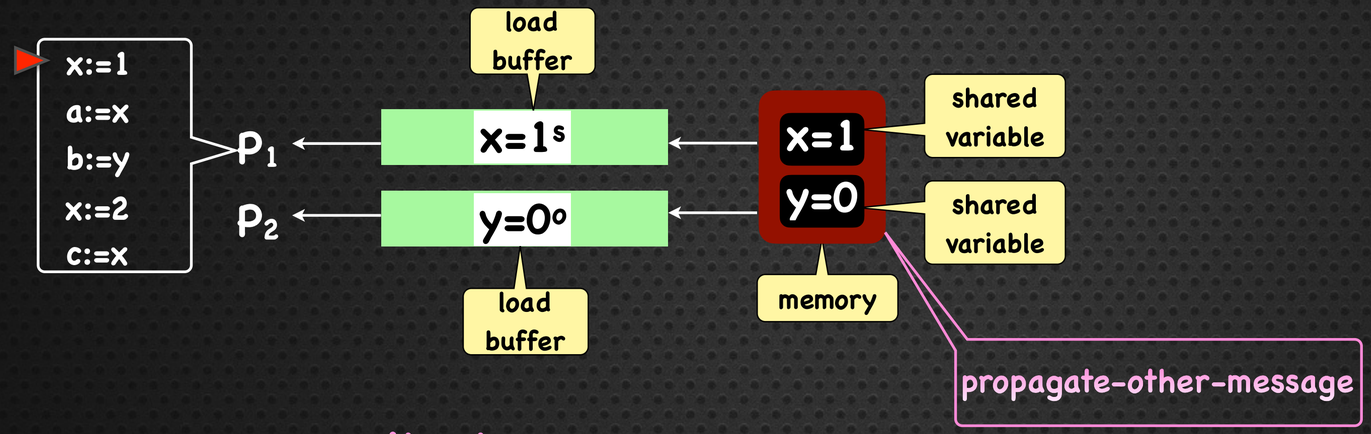
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



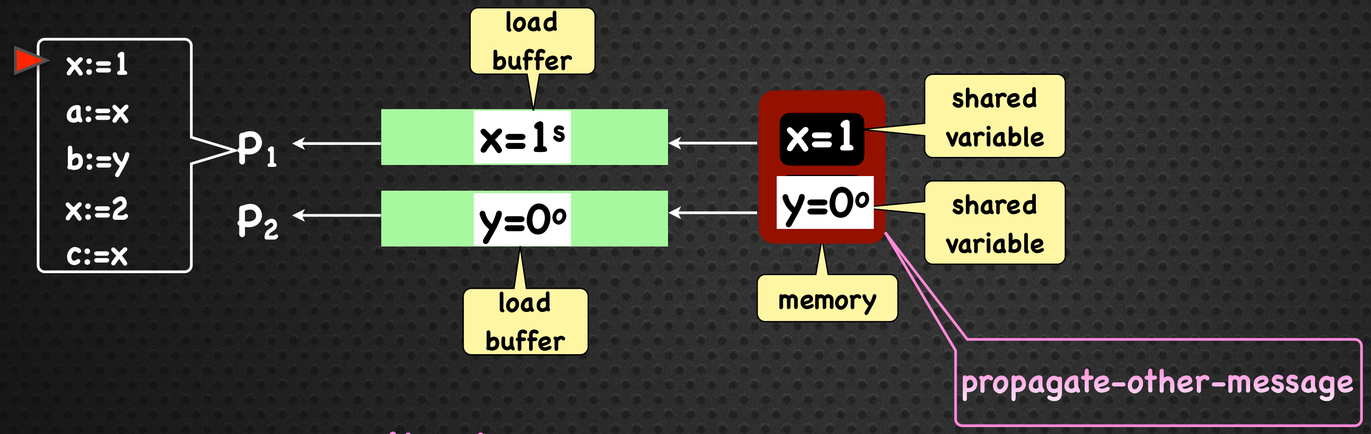
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



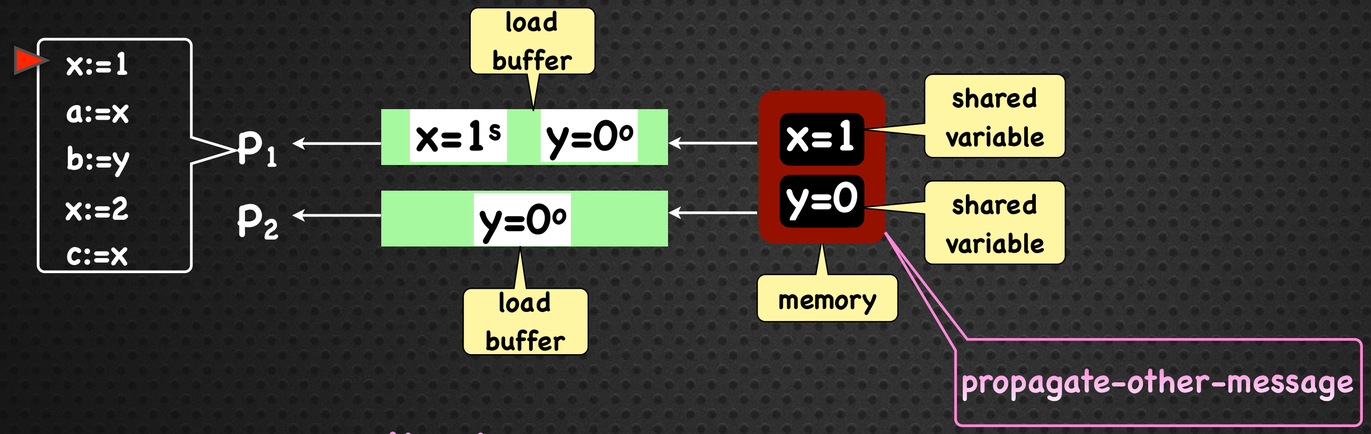
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



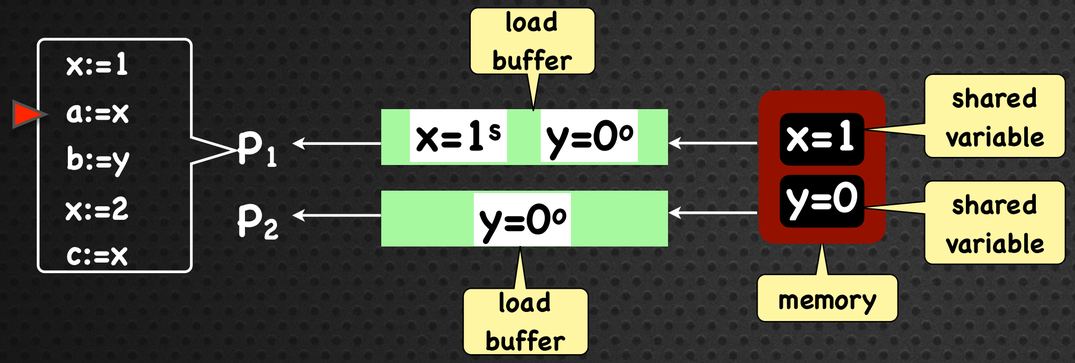
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



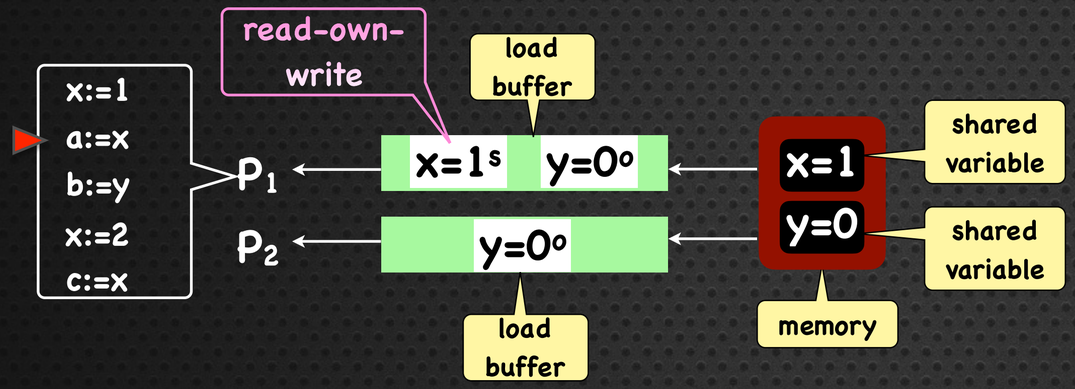
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



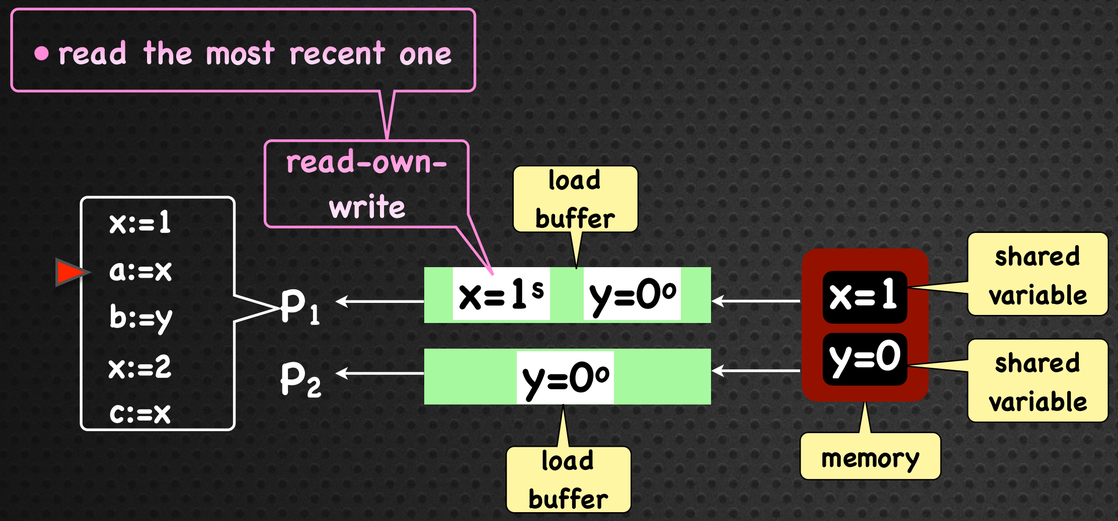
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



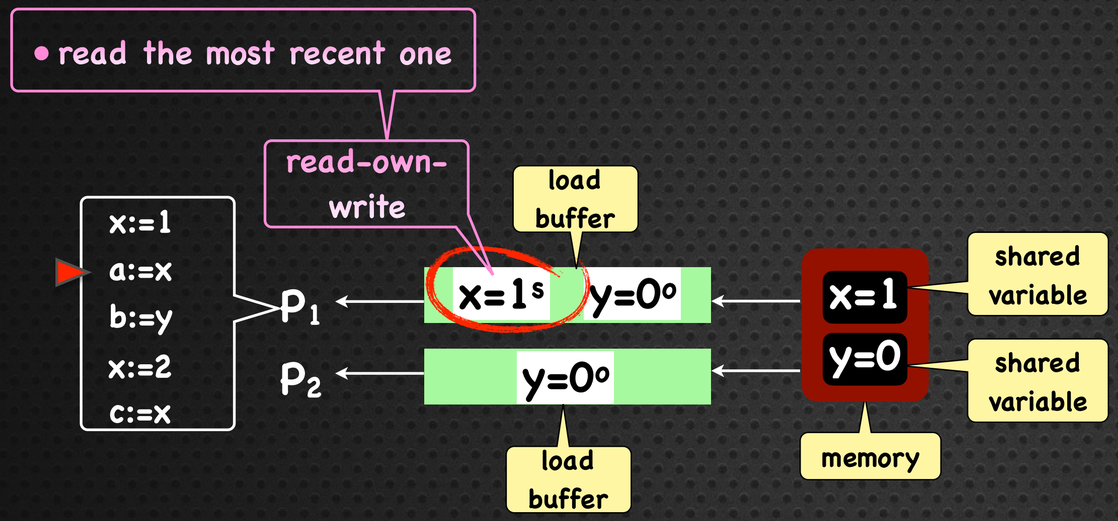
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



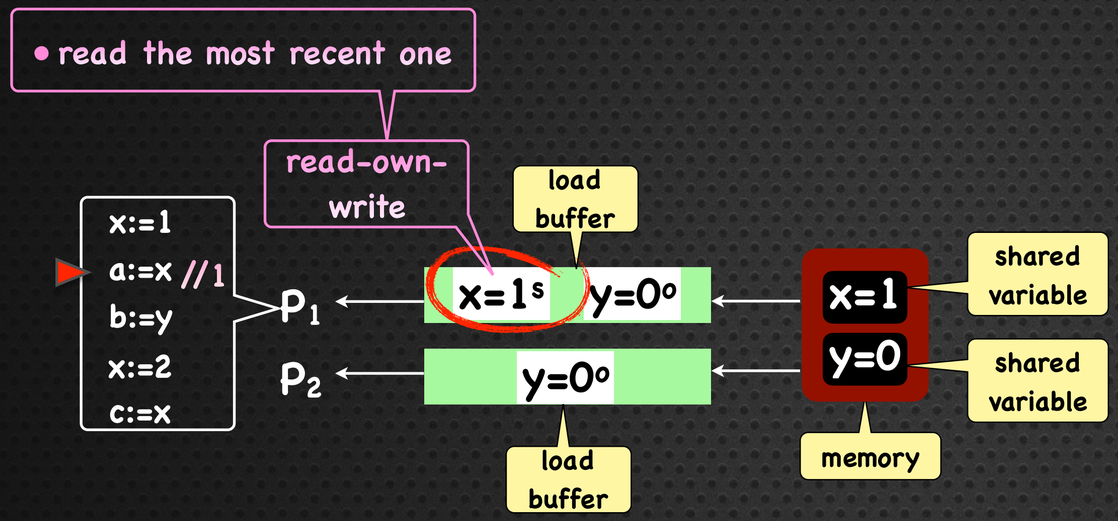
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



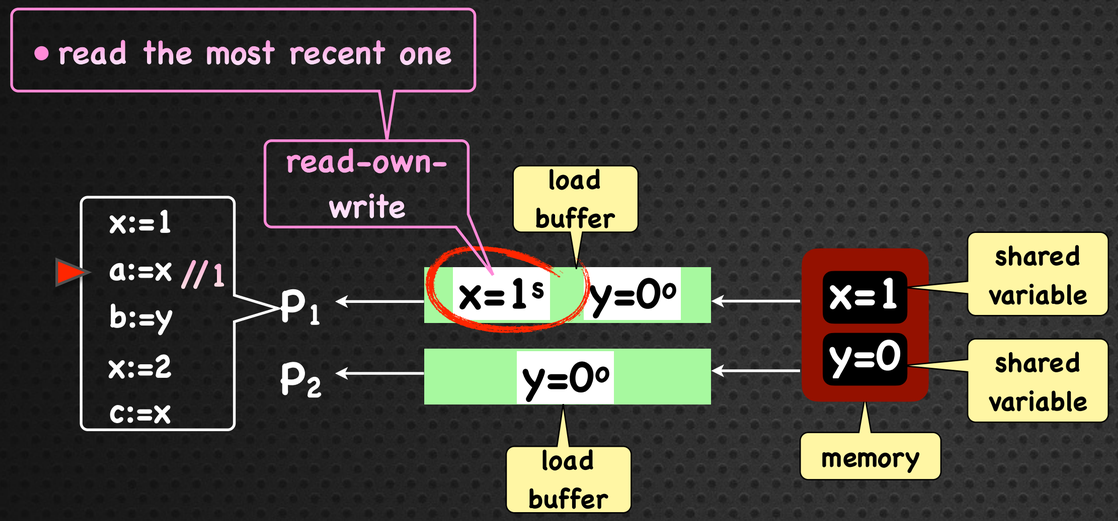
- write-to-memory
- propagate-other-message

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



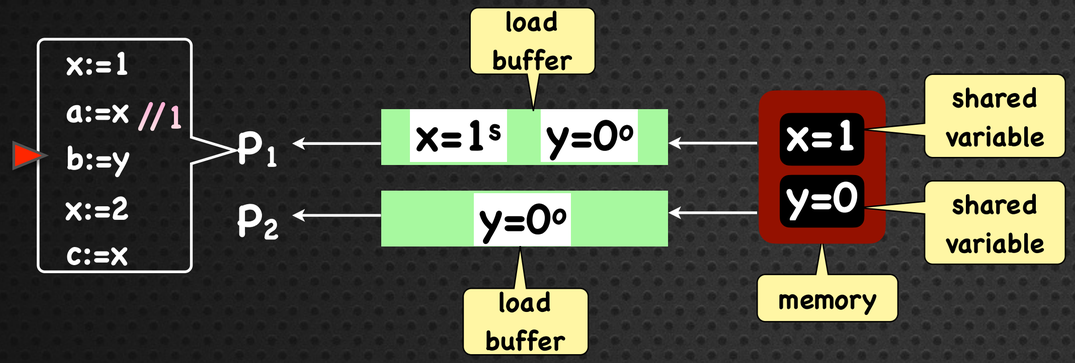
- write-to-memory
- propagate-other-message
- read-own-write

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



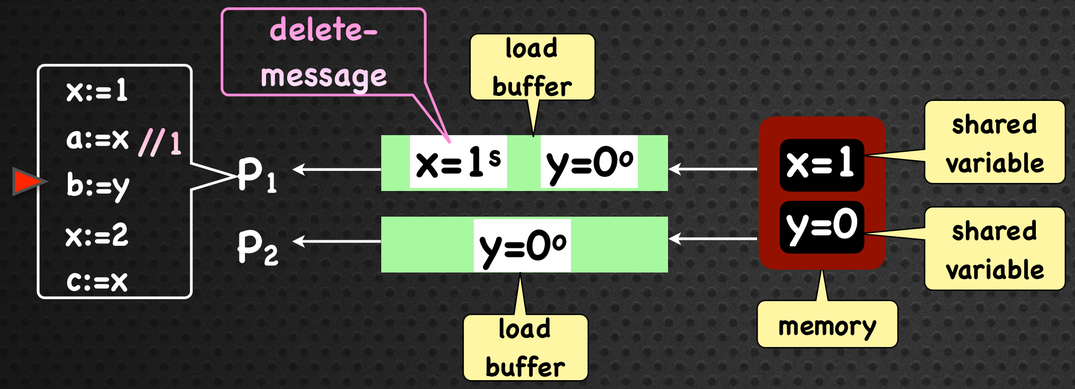
- write-to-memory
- propagate-other-message
- read-own-write

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



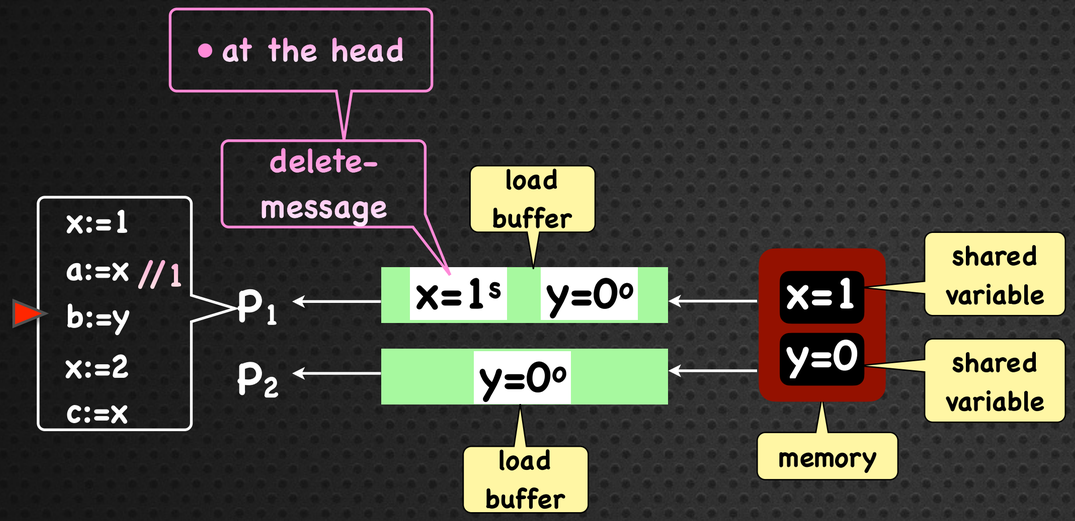
- write-to-memory
- propagate-other-message
- read-own-write

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



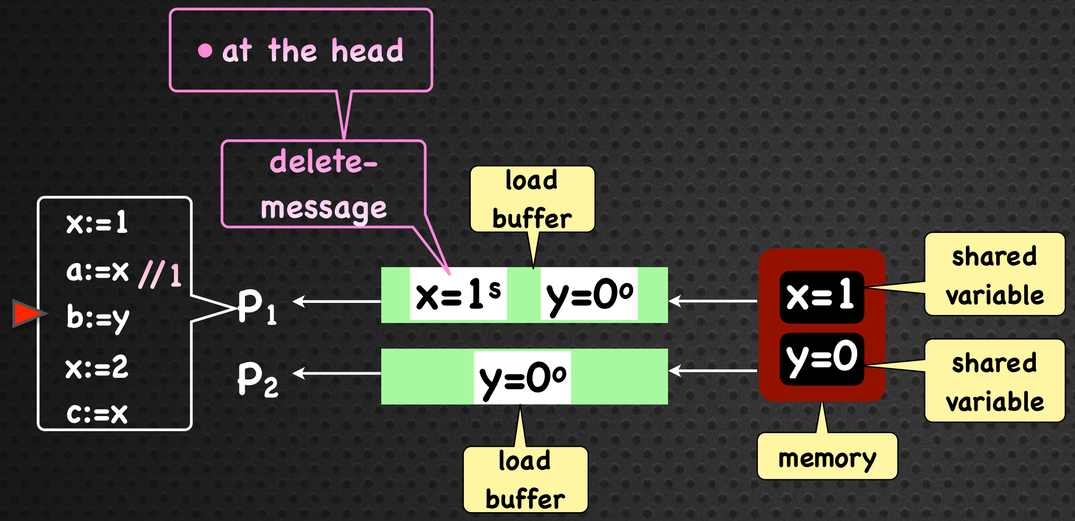
- write-to-memory
- propagate-other-message
- read-own-write

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



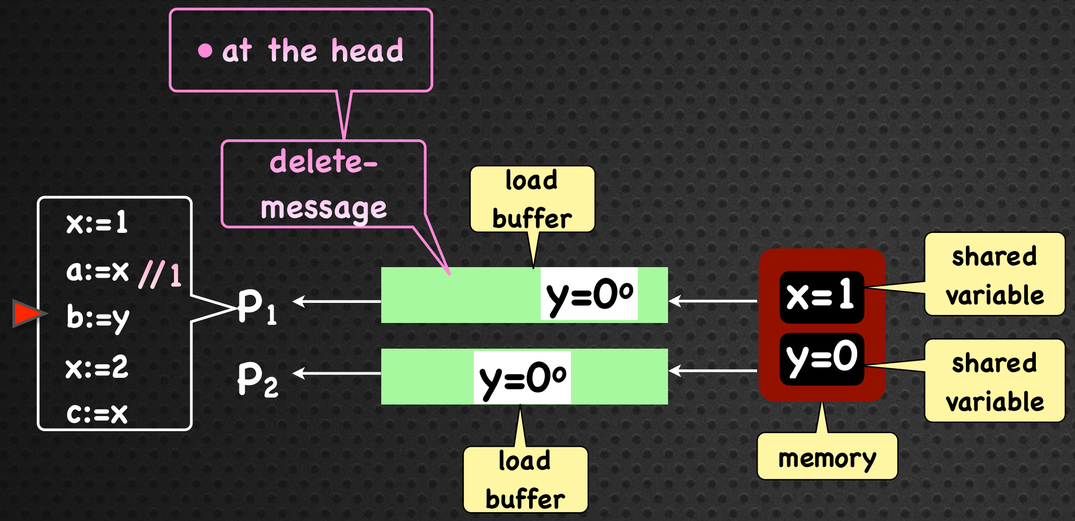
- write-to-memory
- propagate-other-message
- read-own-write
- delete

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



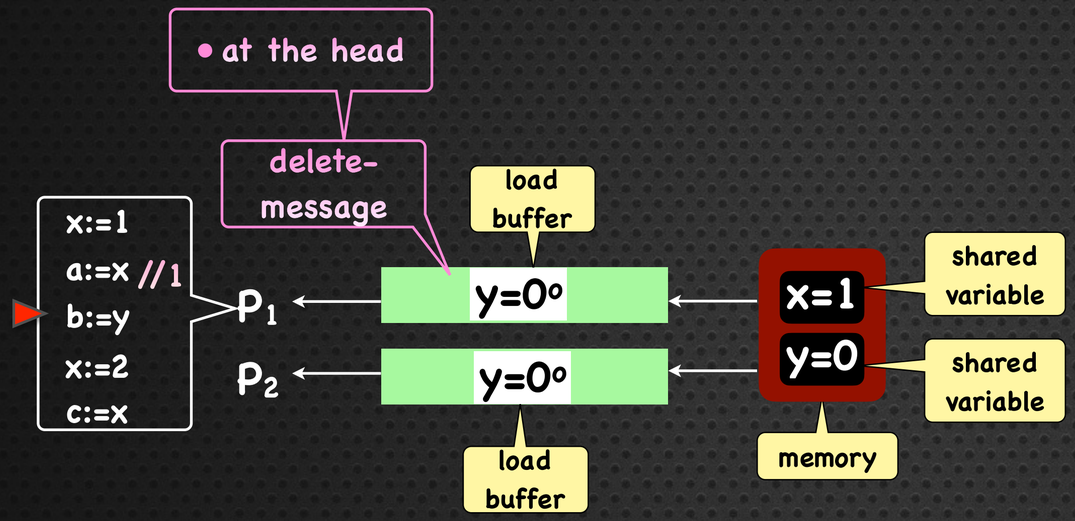
- write-to-memory
- propagate-other-message
- read-own-write
- delete

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



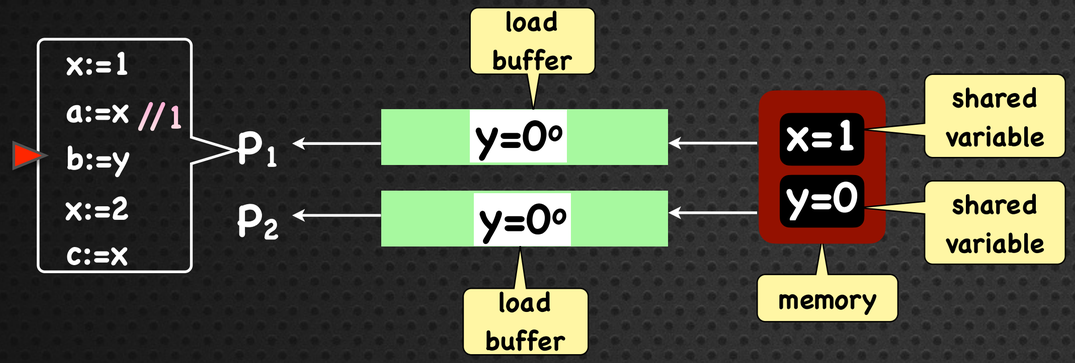
- write-to-memory
- propagate-other-message
- read-own-write
- delete

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



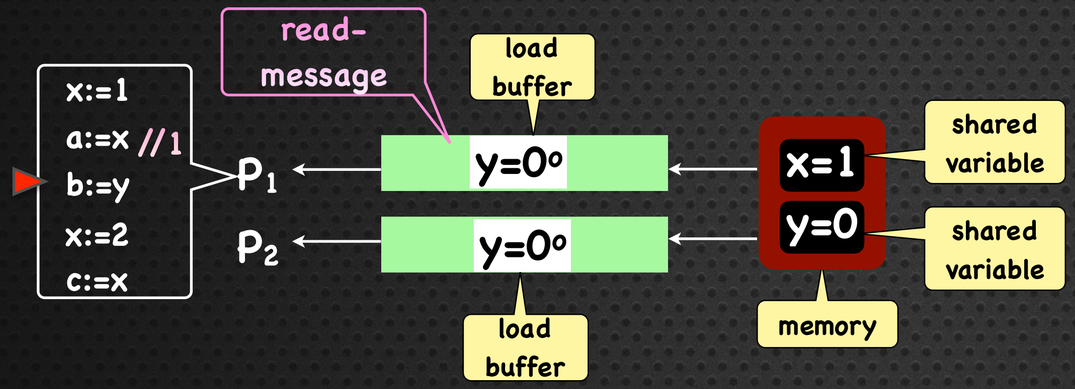
- write-to-memory
- propagate-other-message
- read-own-write
- delete

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



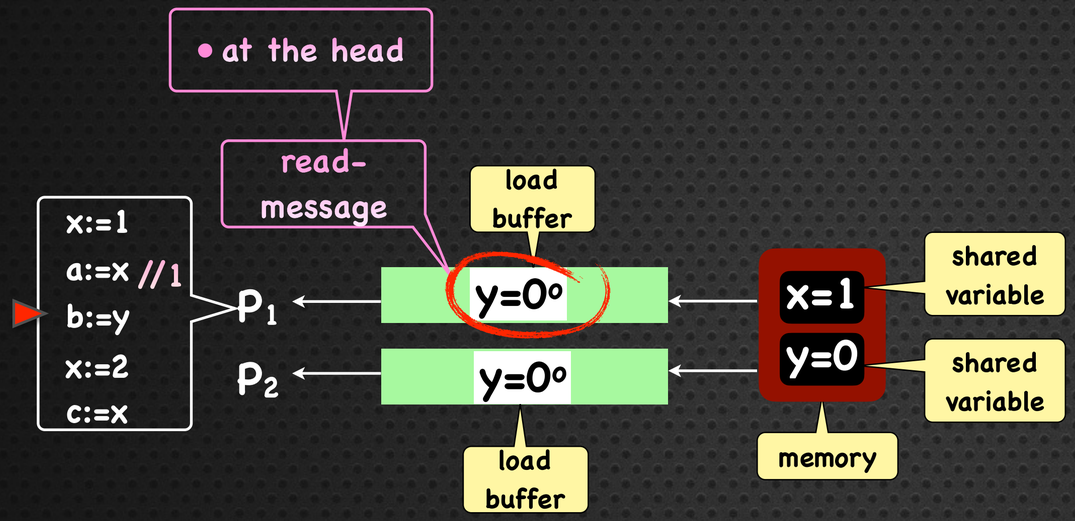
- write-to-memory
- propagate-other-message
- read-own-write
- delete

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



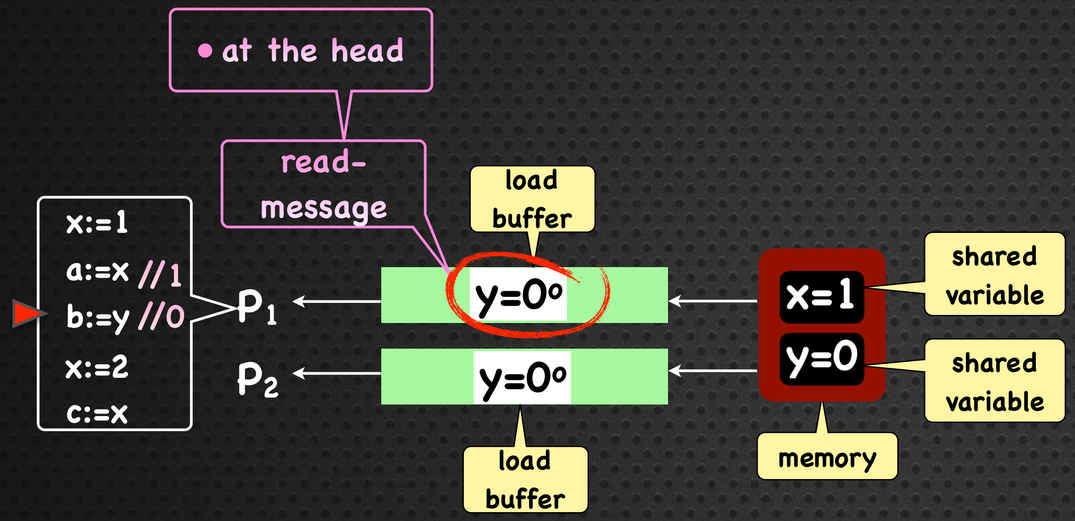
- write-to-memory
- propagate-other-message
- read-own-write
- delete

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



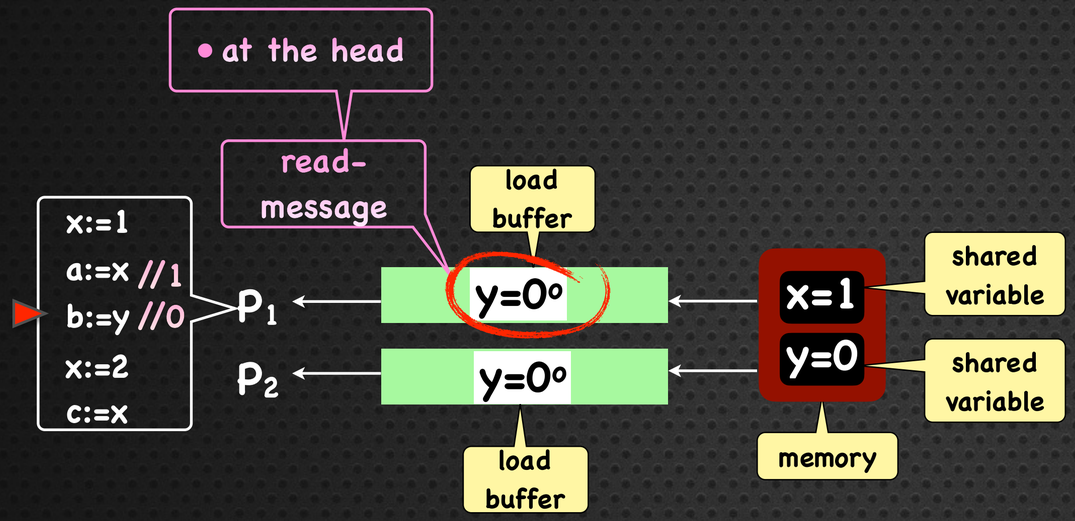
- write-to-memory
- propagate-other-message
- read-own-write
- delete

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



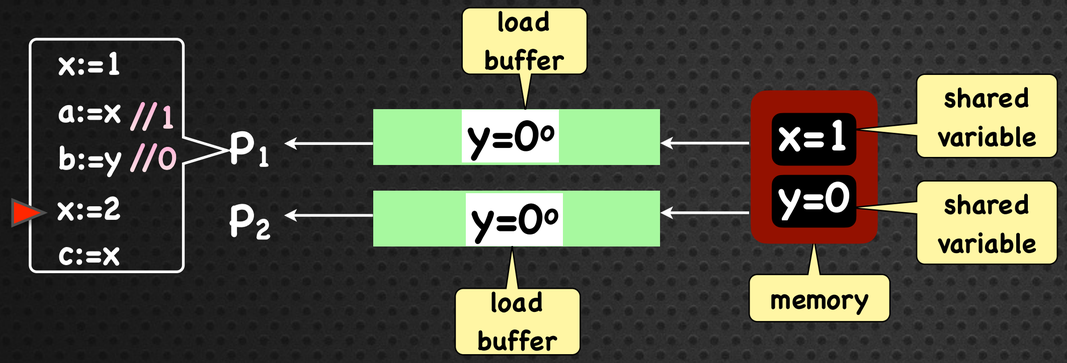
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



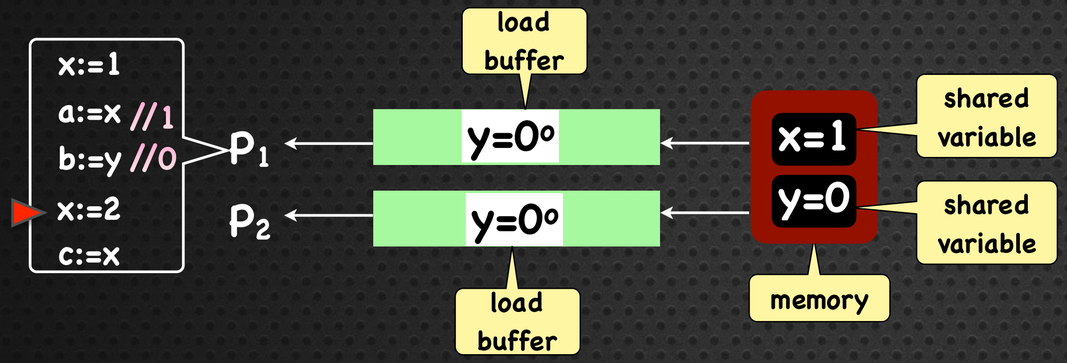
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



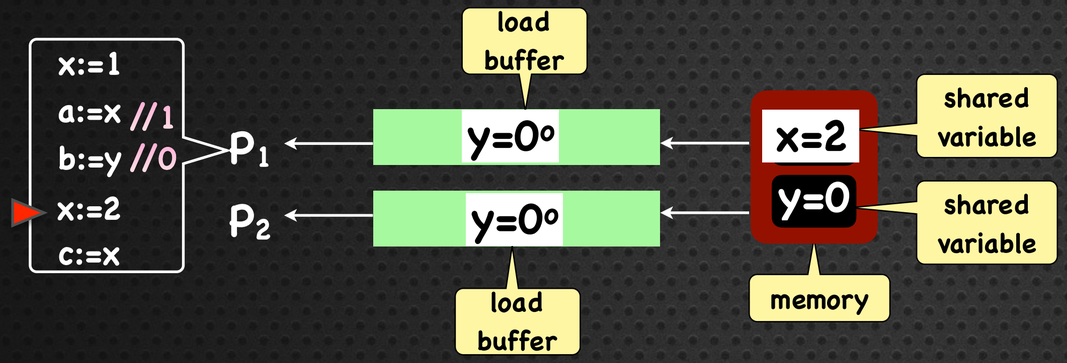
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



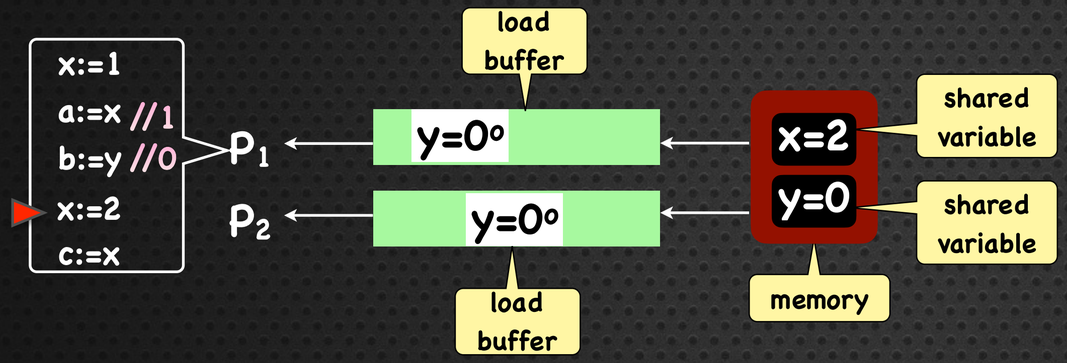
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



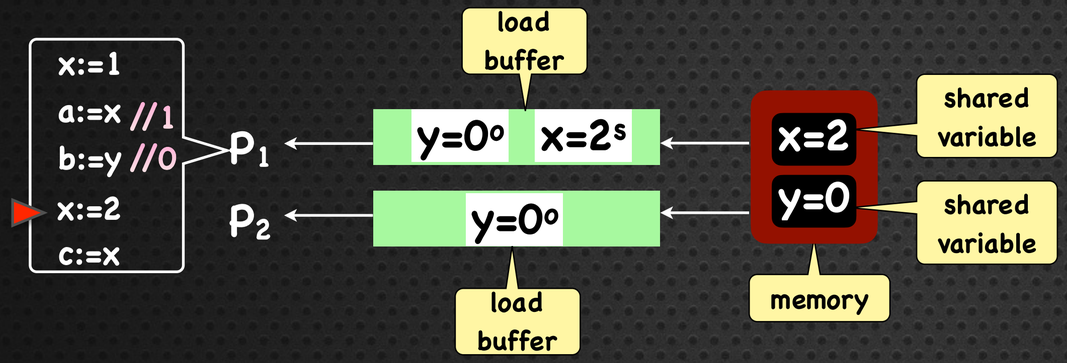
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



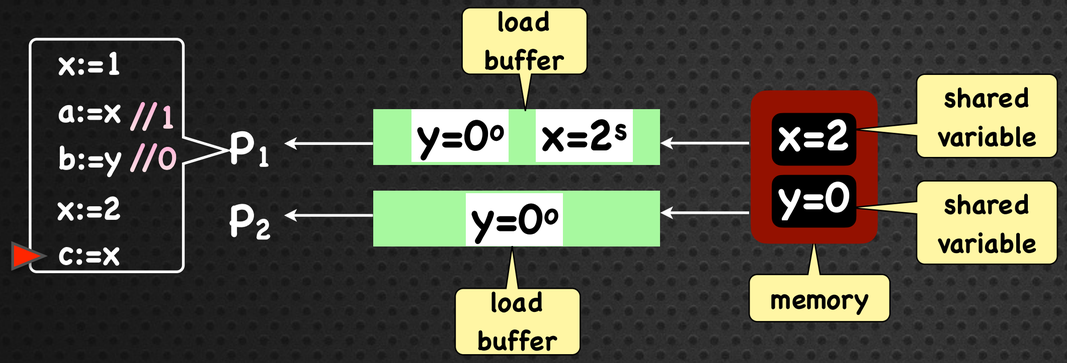
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



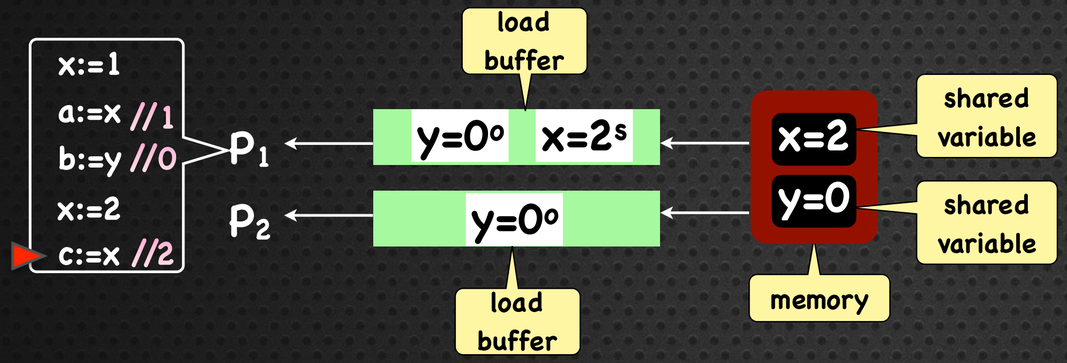
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



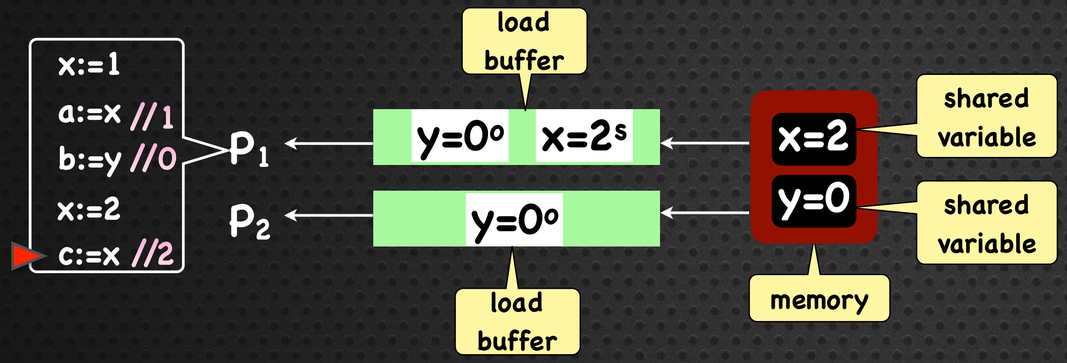
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



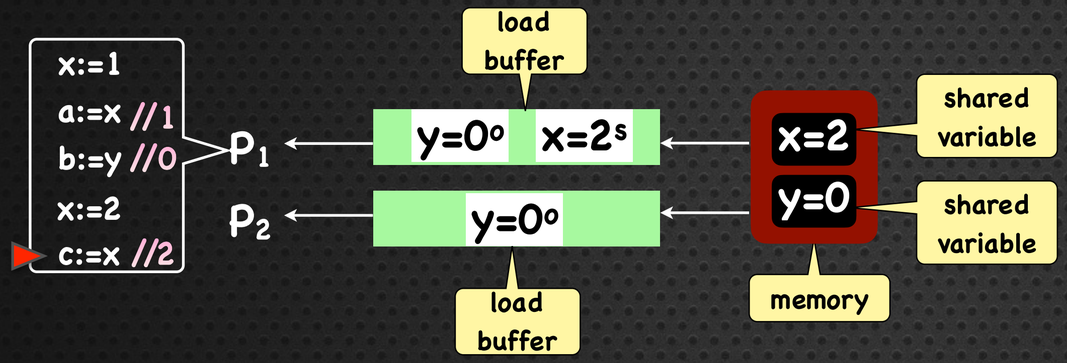
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



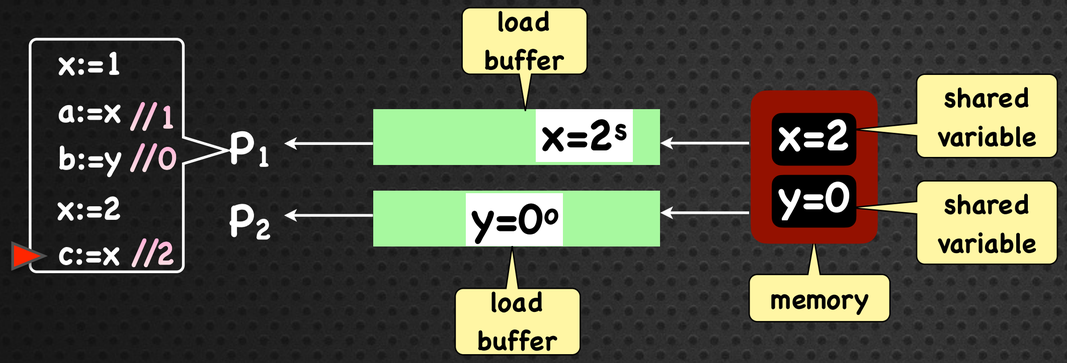
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



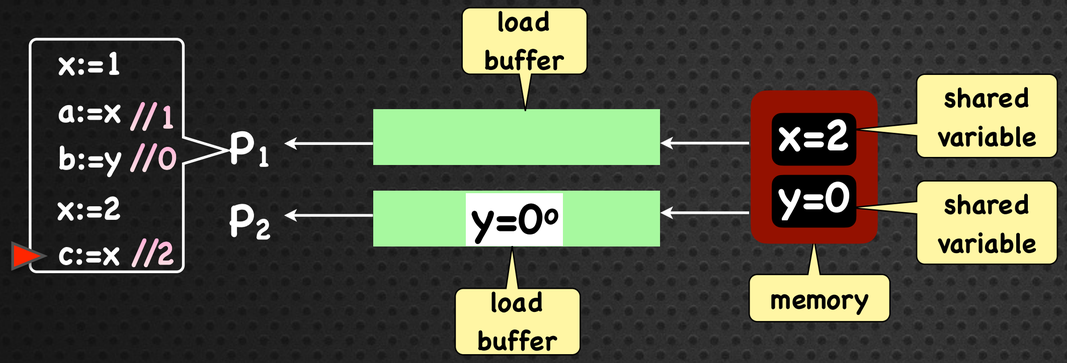
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



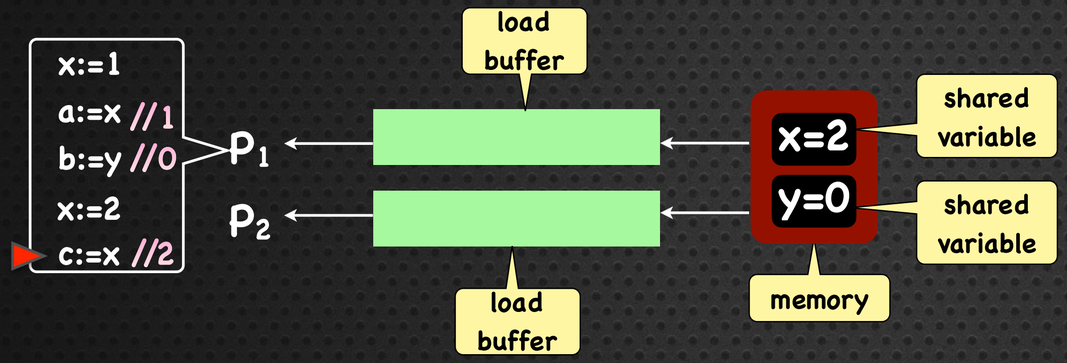
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



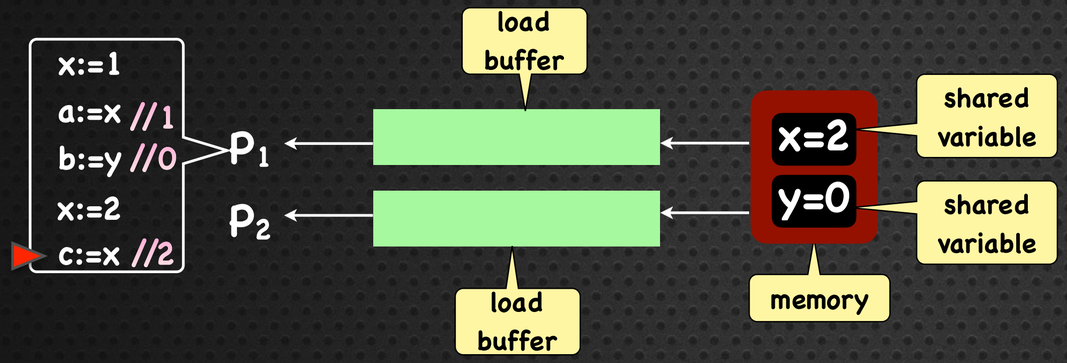
- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

- write-to-buffer
- read-own-write
- read-from-memory
- update

Load-Buffer Semantics

- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics (Examples)

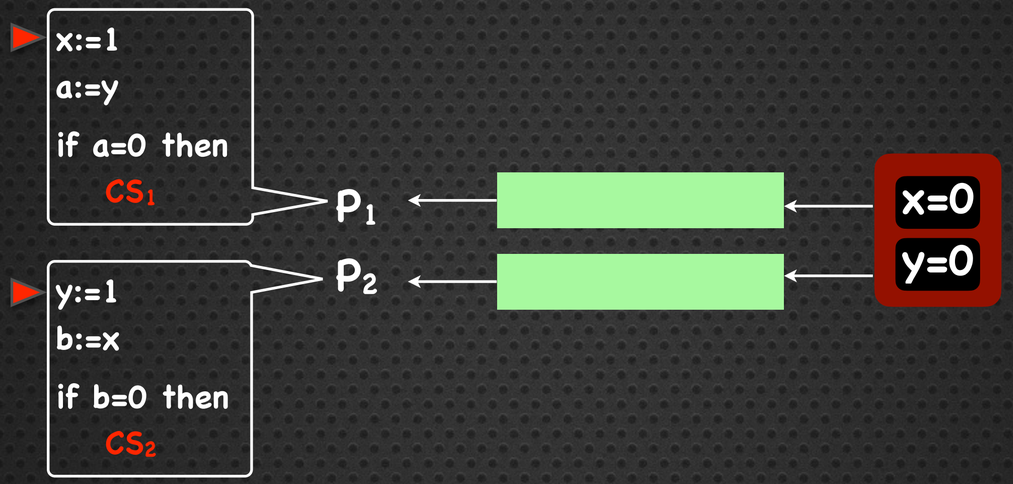
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



not (CS_1 and CS)

assertion

potentially bad behaviors

extra behaviors

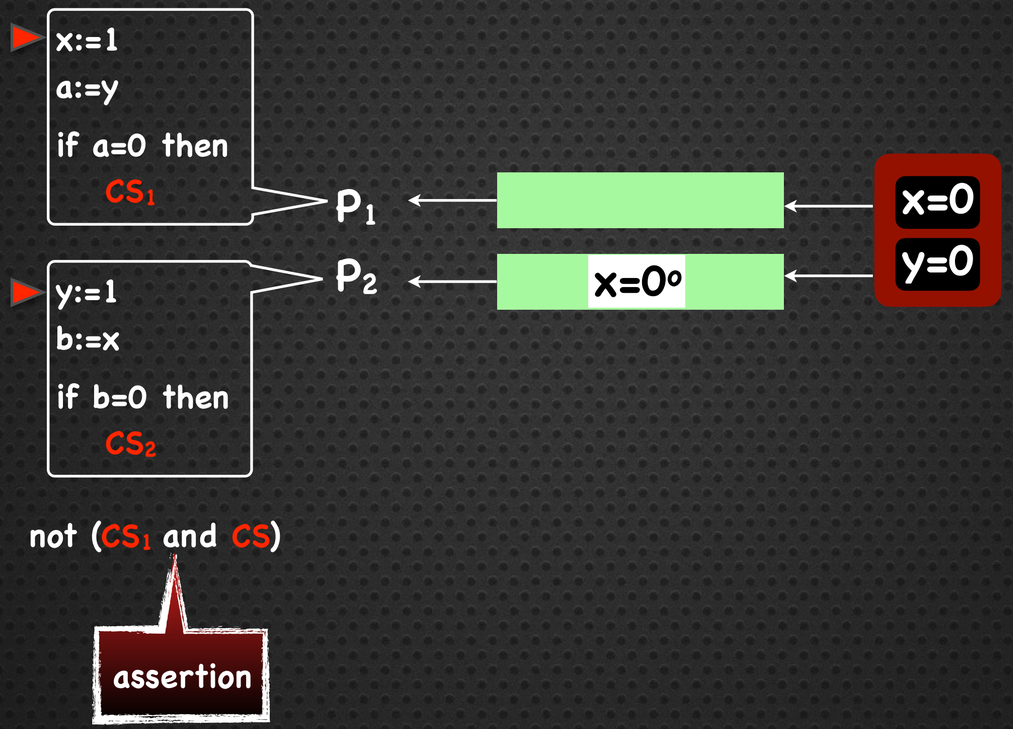
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

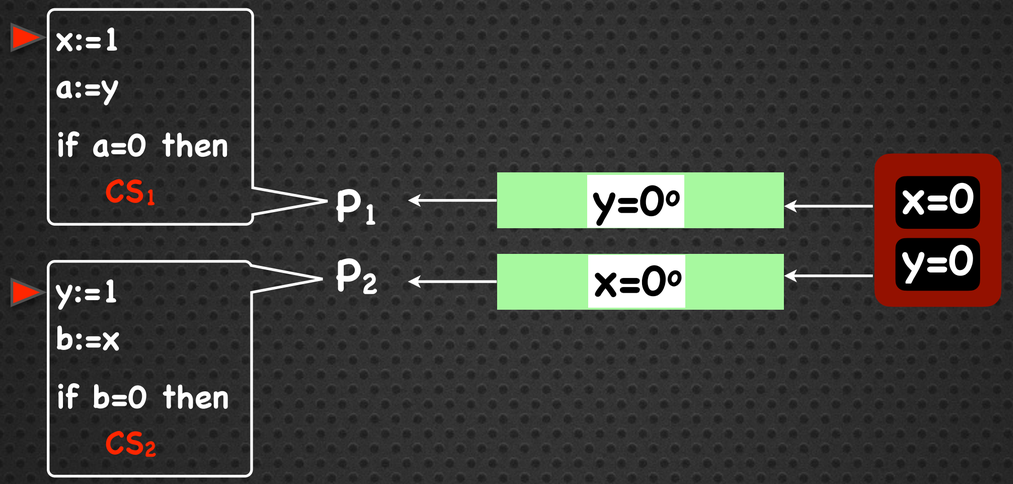
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



not (CS_1 and CS)

assertion

potentially bad behaviors

extra behaviors

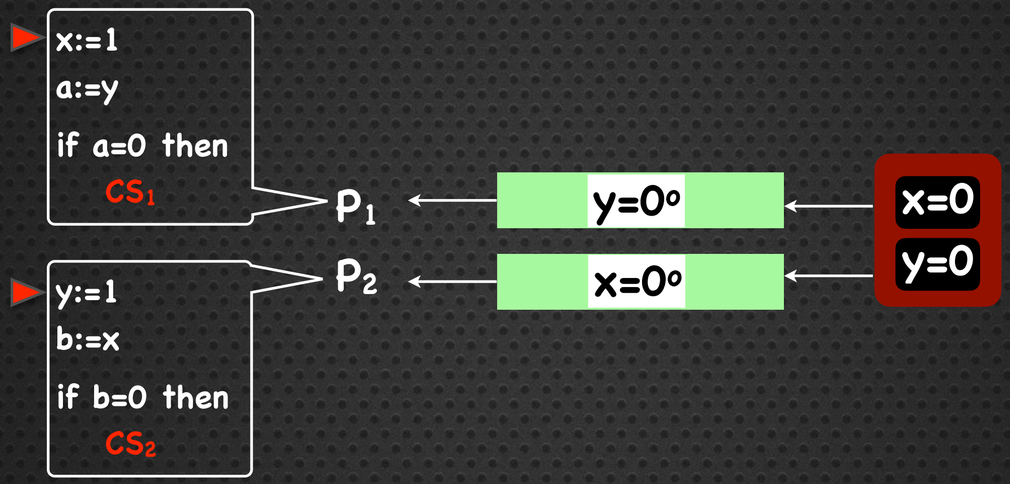
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



not (CS_1 and CS)

assertion

potentially bad behaviors

extra behaviors

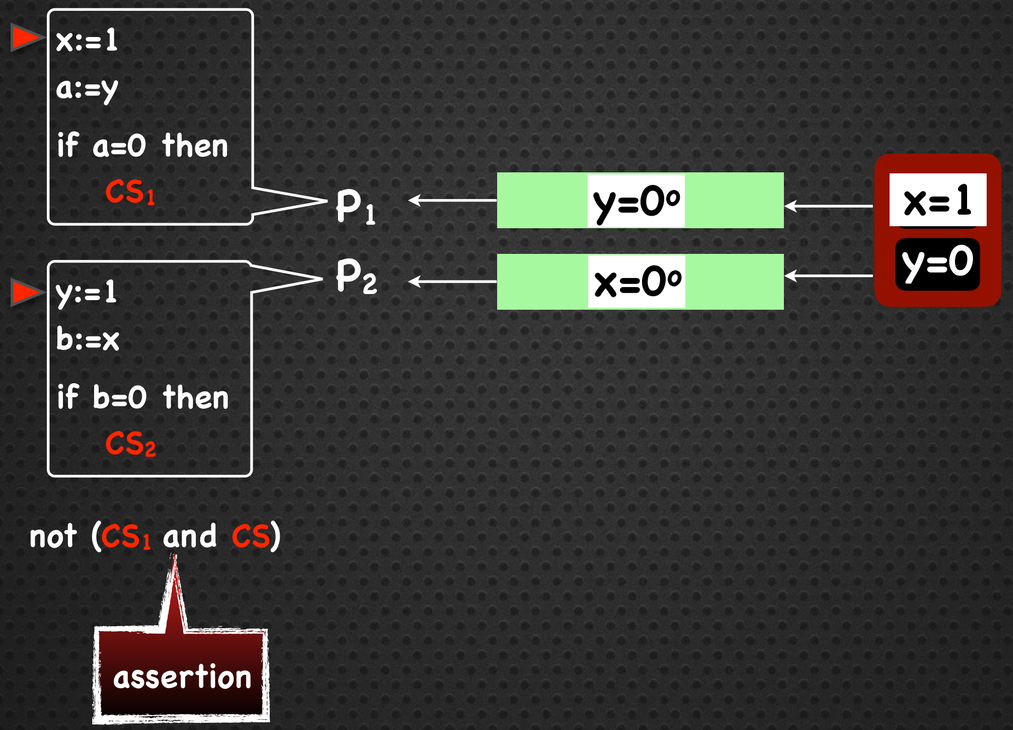
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

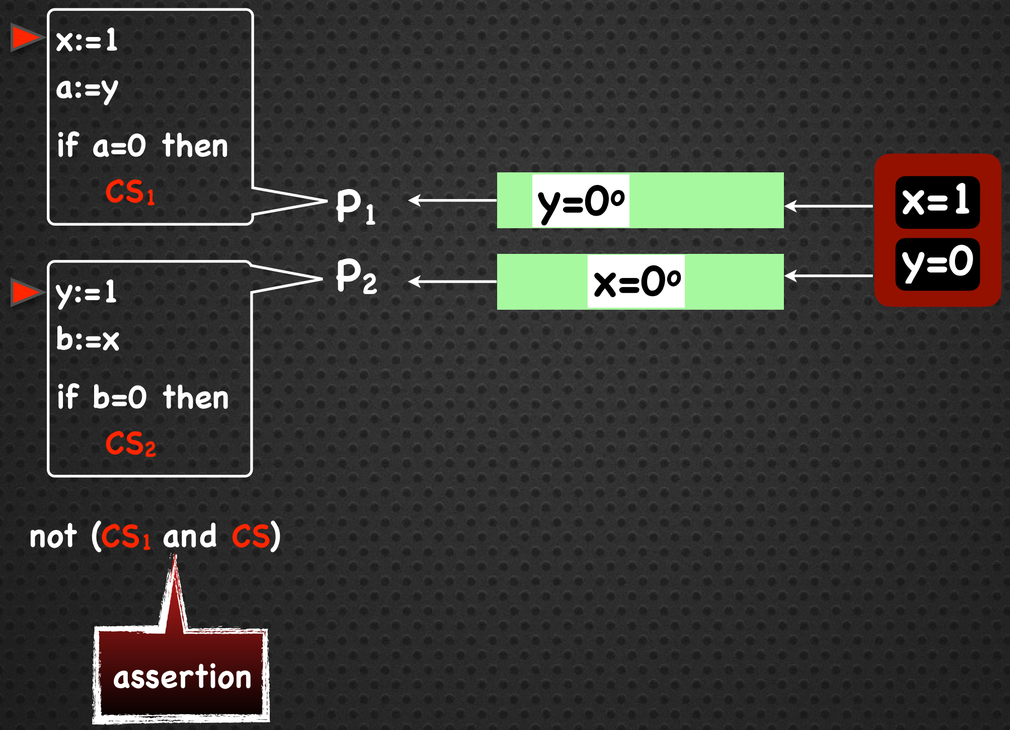
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

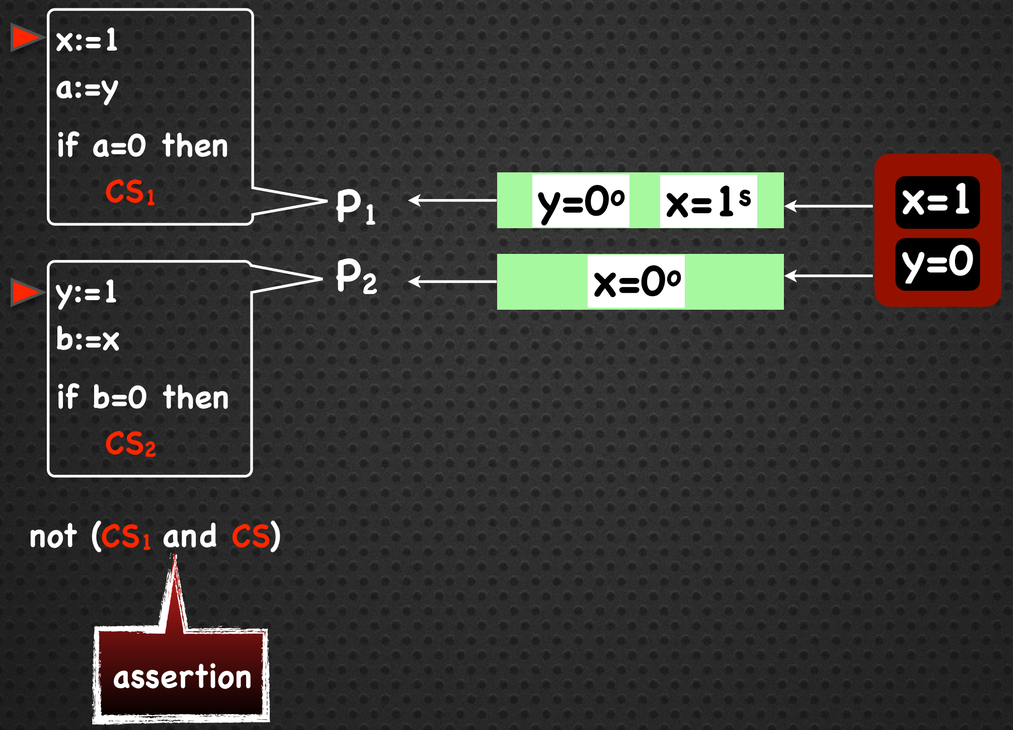
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

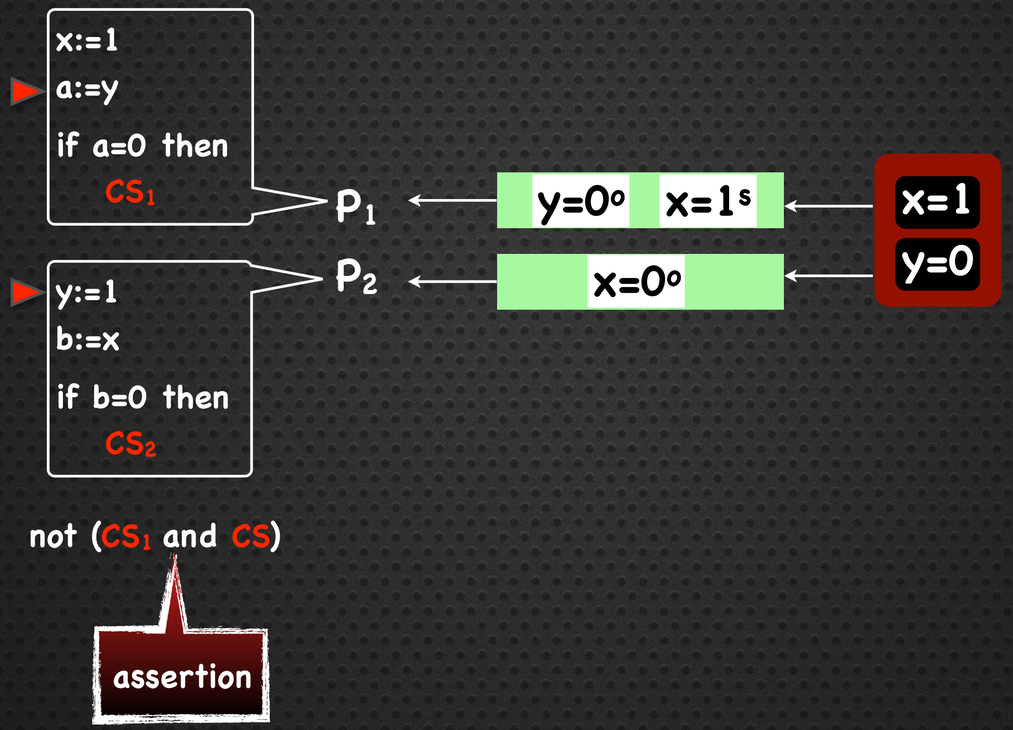
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

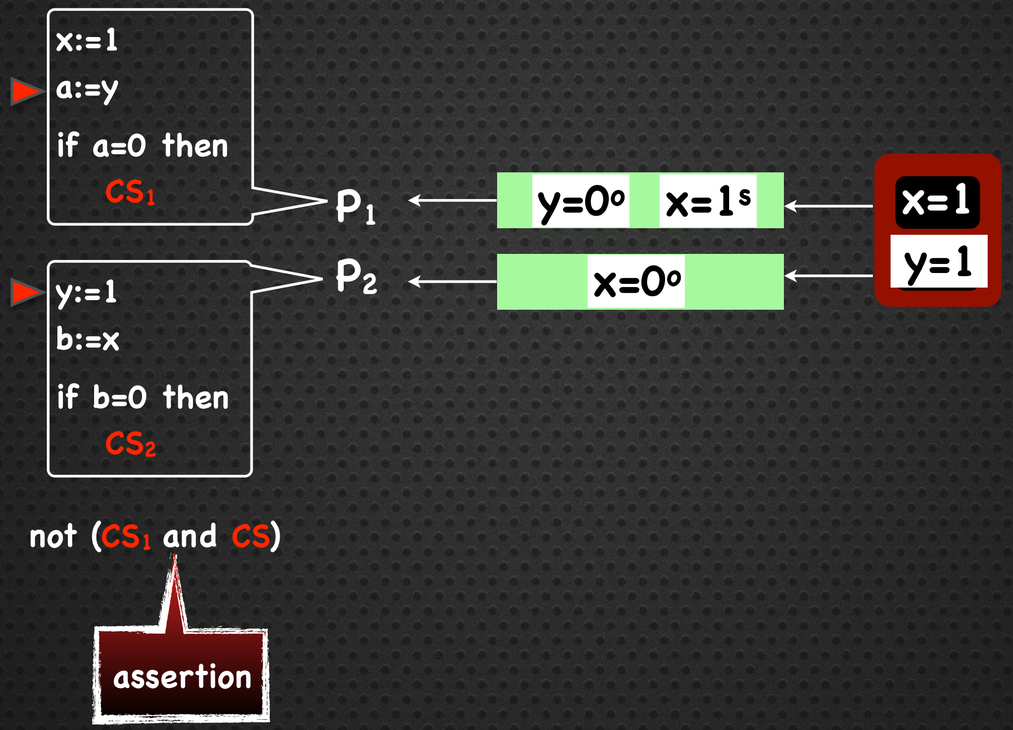
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

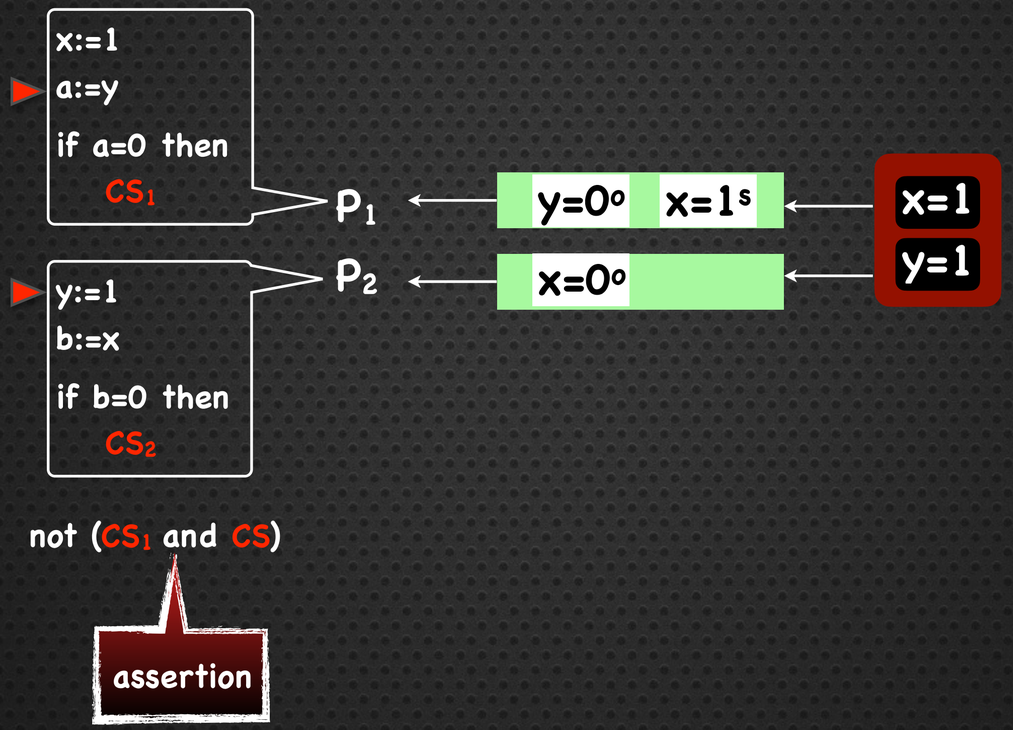
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

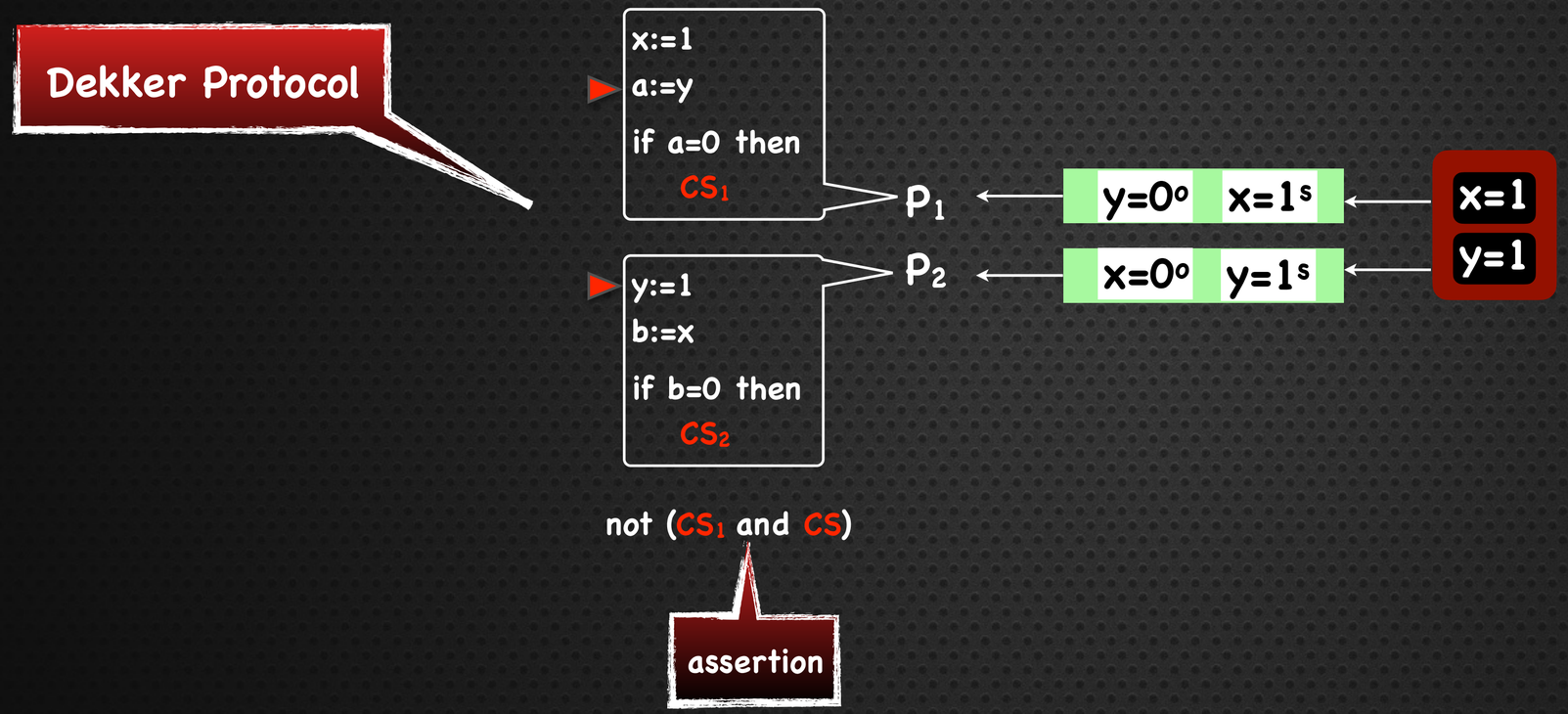
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



potentially bad behaviors

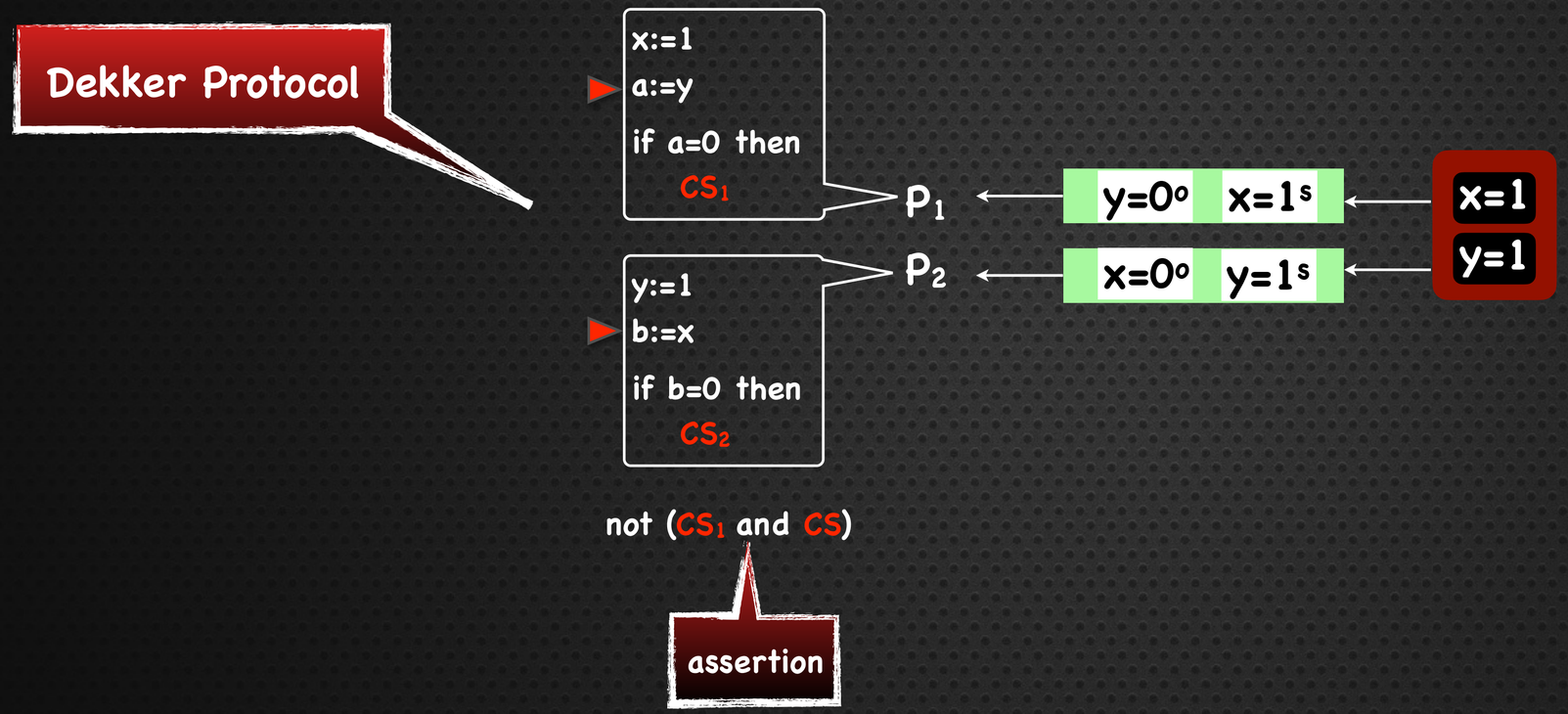
extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics



potentially bad behaviors

extra behaviors

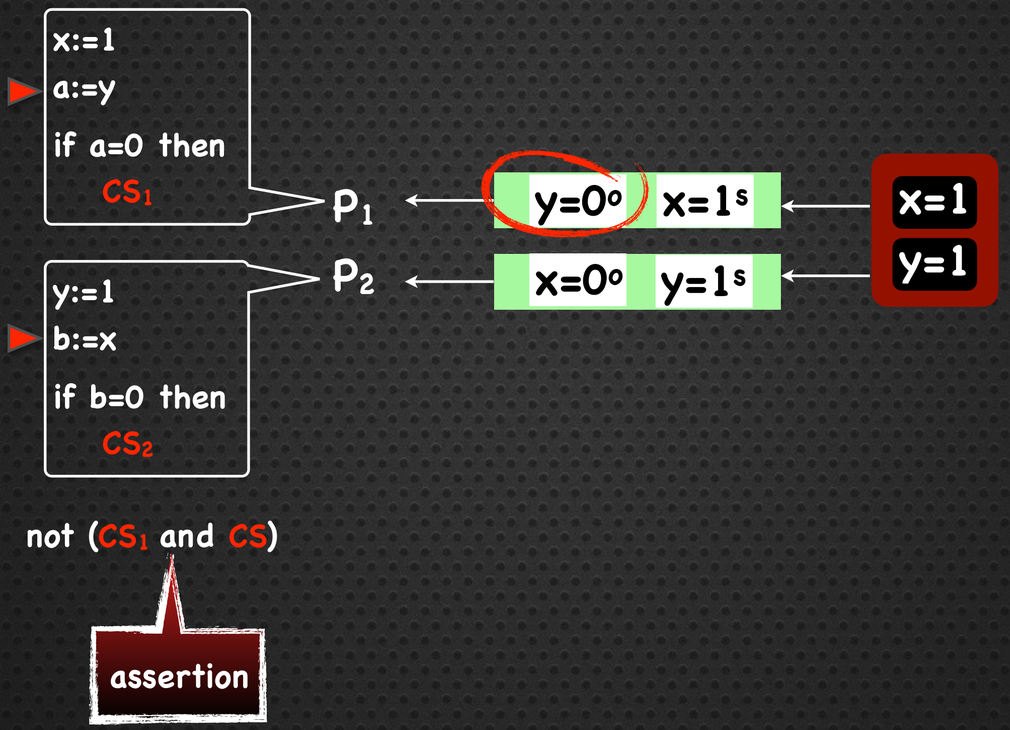
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

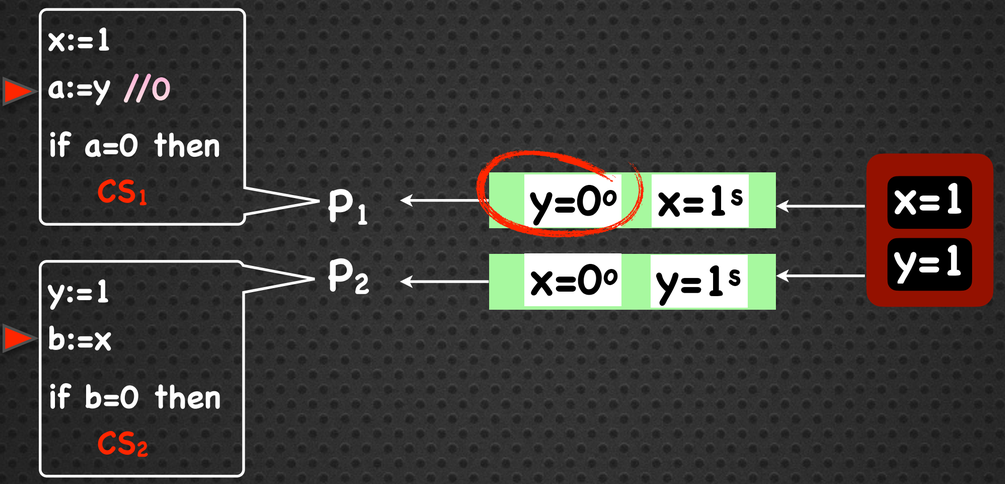
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

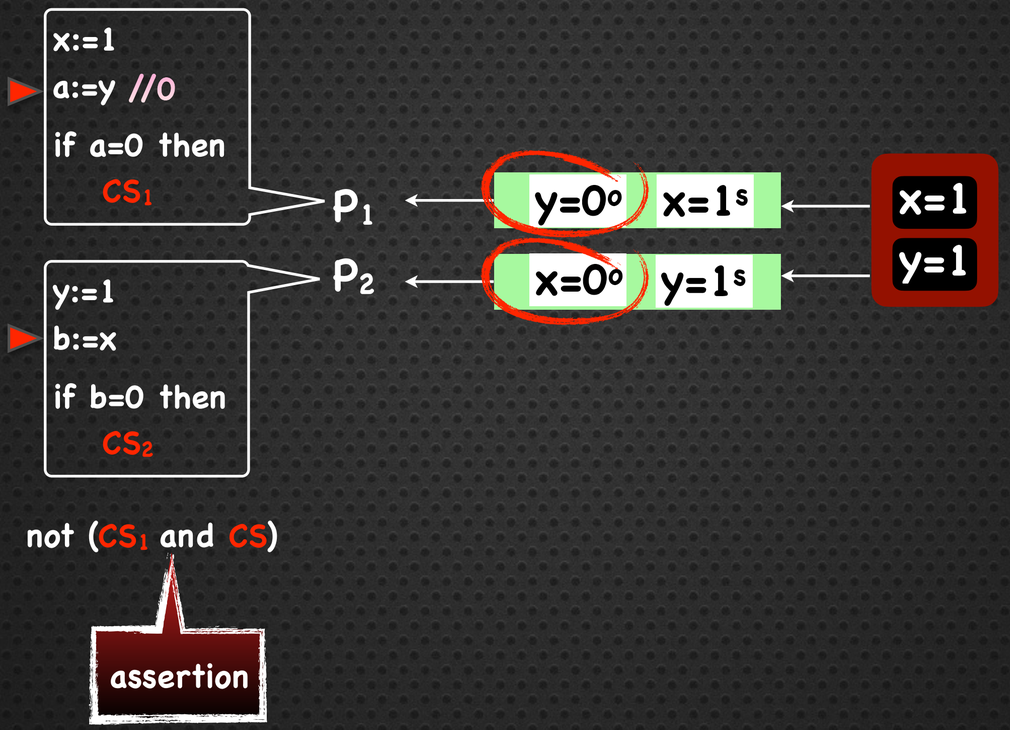
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

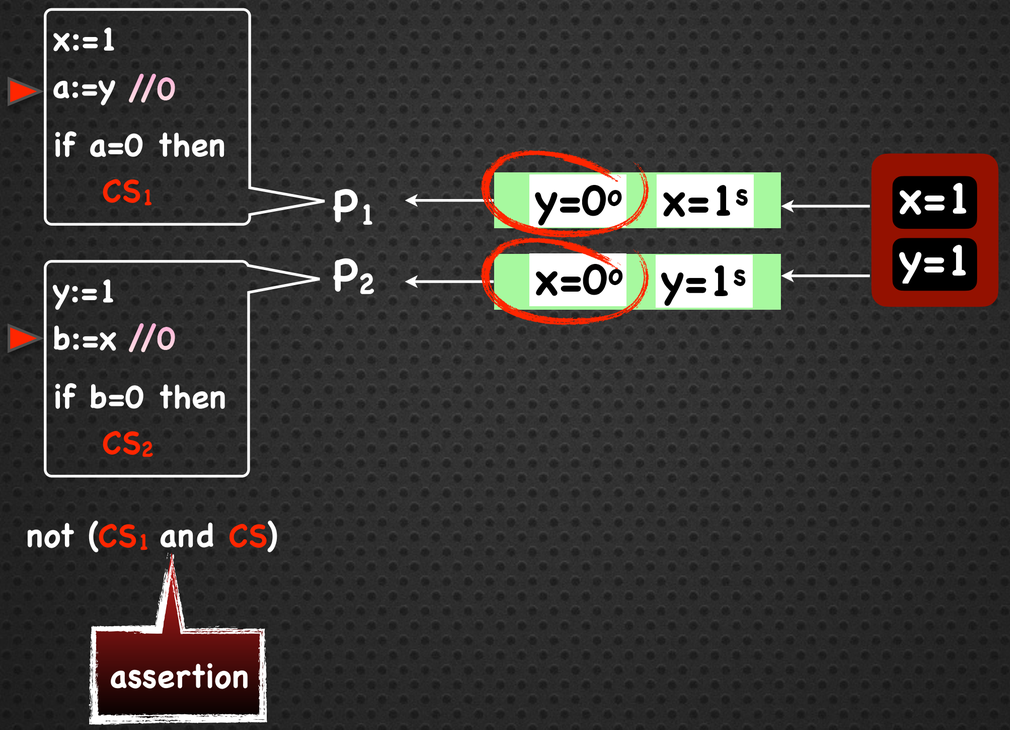
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

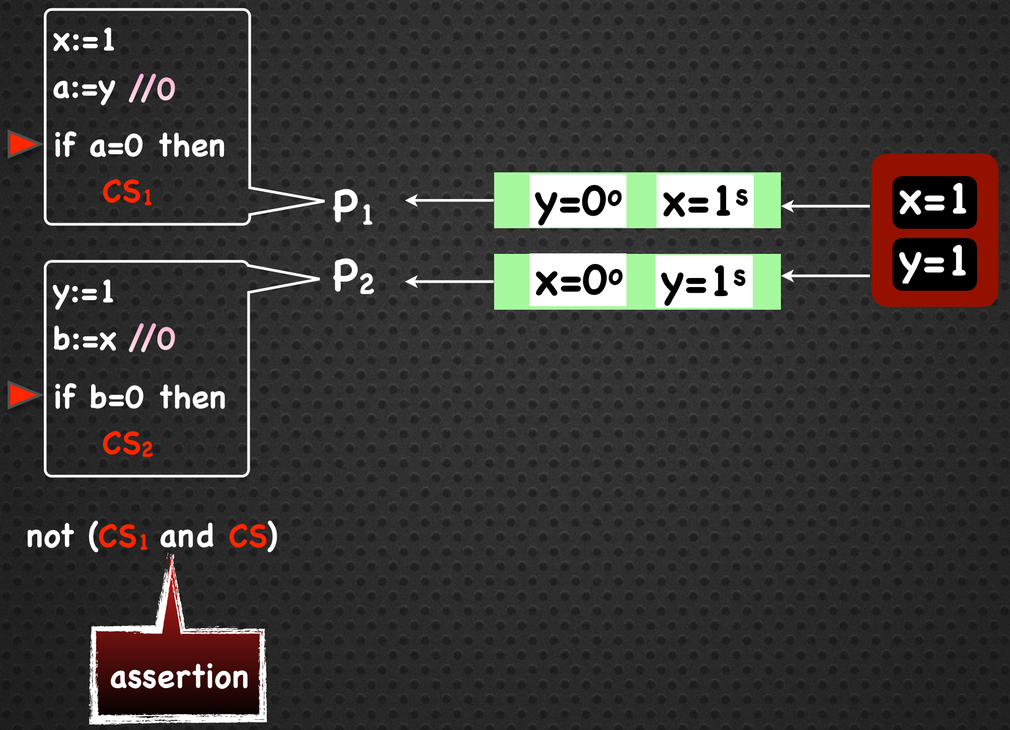
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

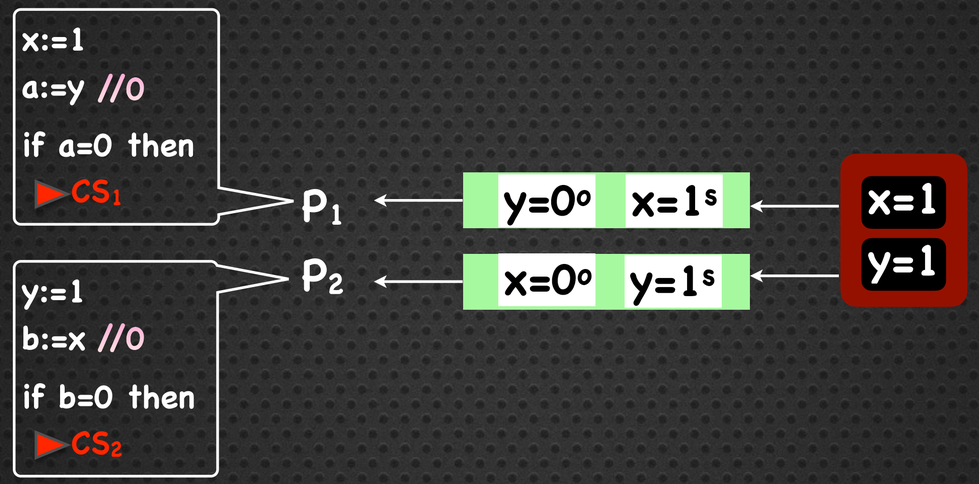
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

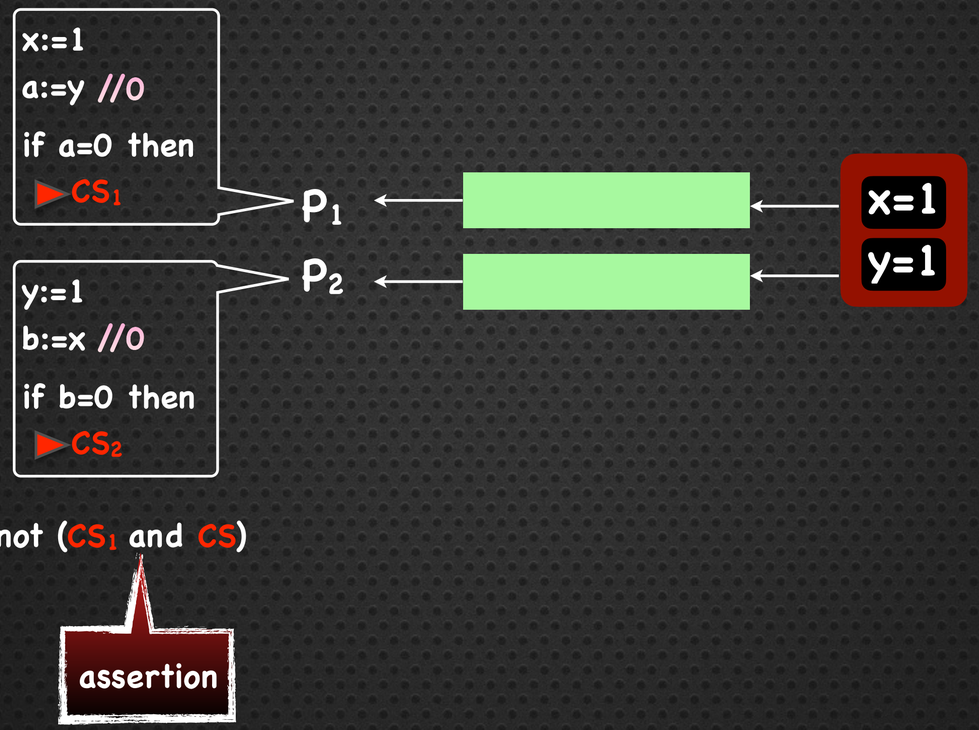
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Dekker Protocol



potentially bad behaviors

extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics (Examples)

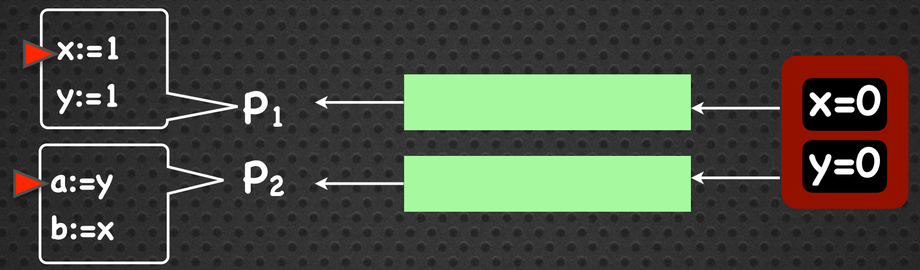
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

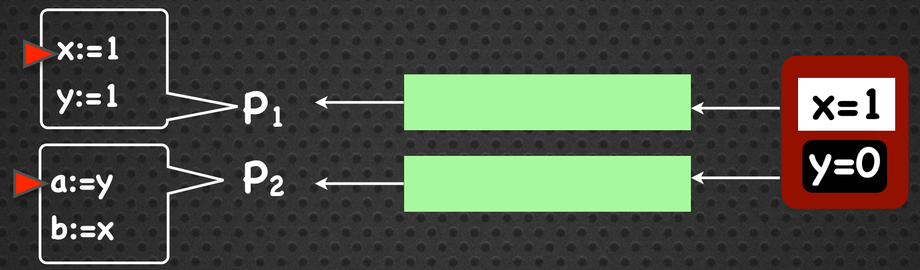
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

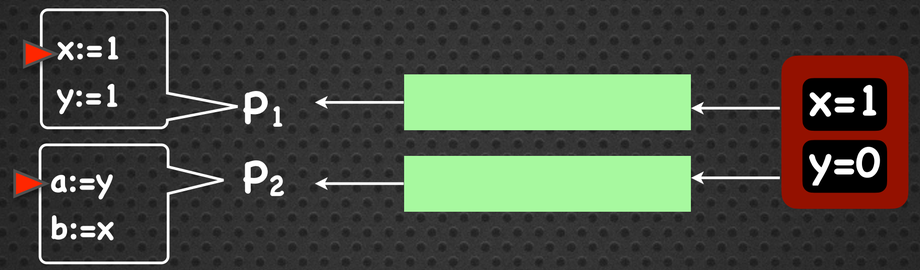
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

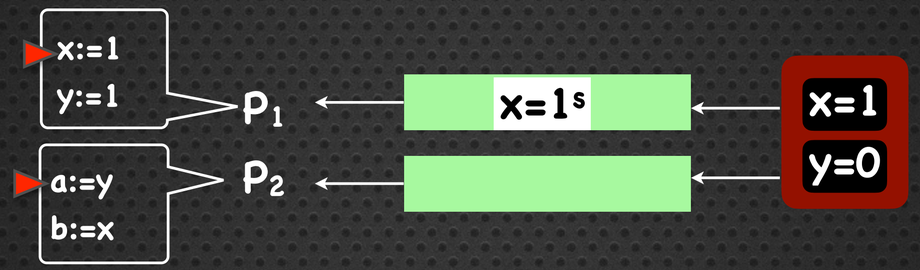
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

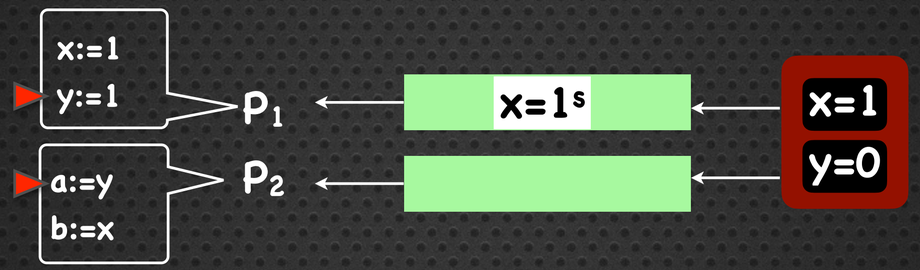
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

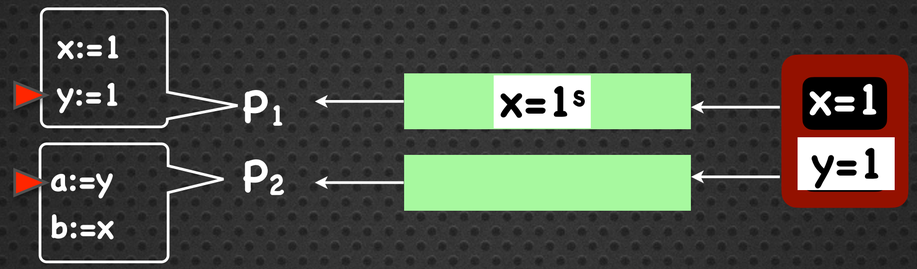
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

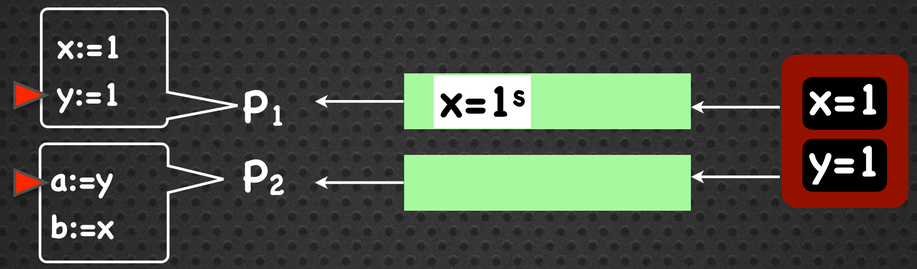
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

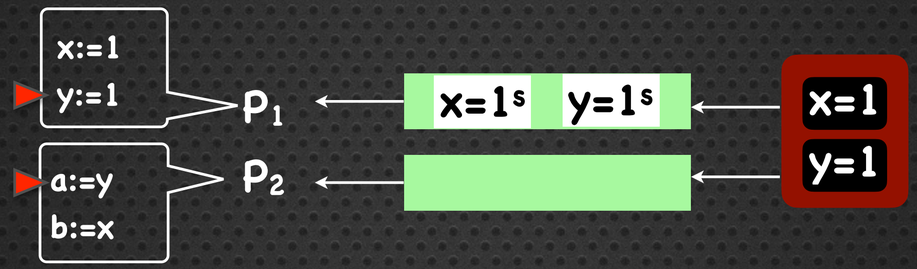
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

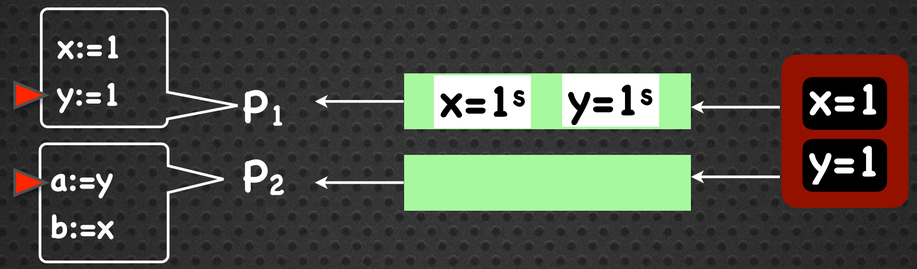
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

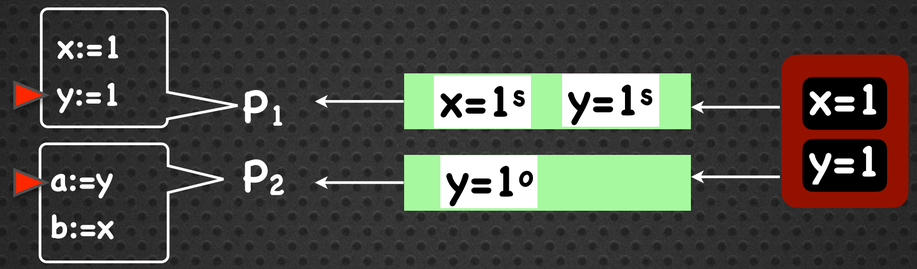
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

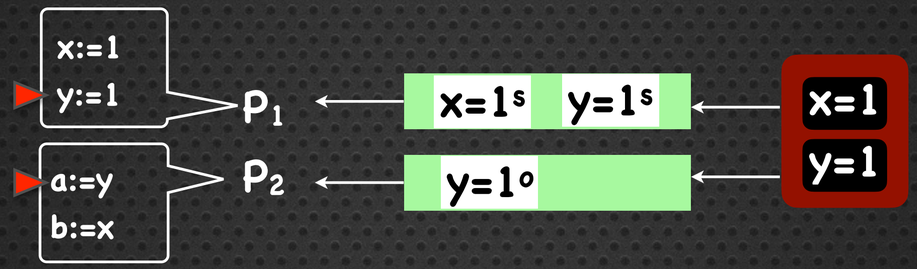
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

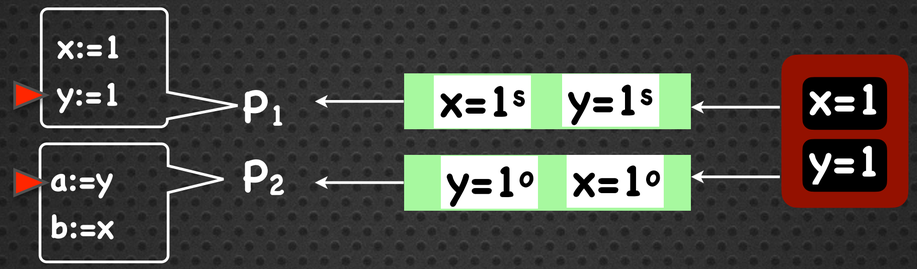
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

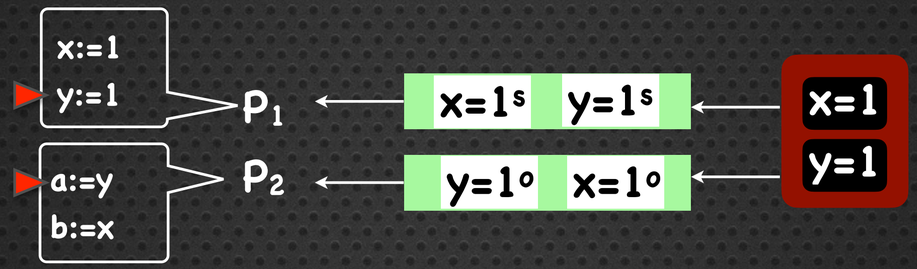
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

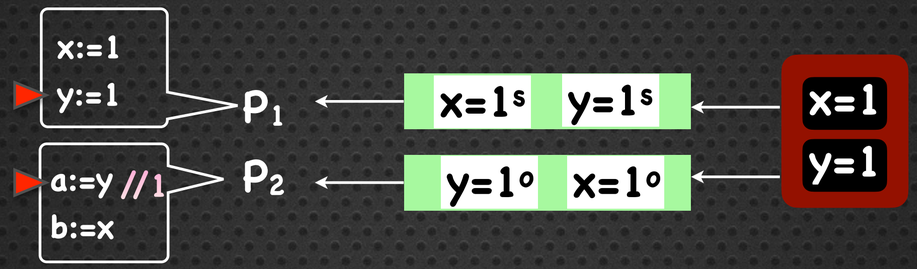
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

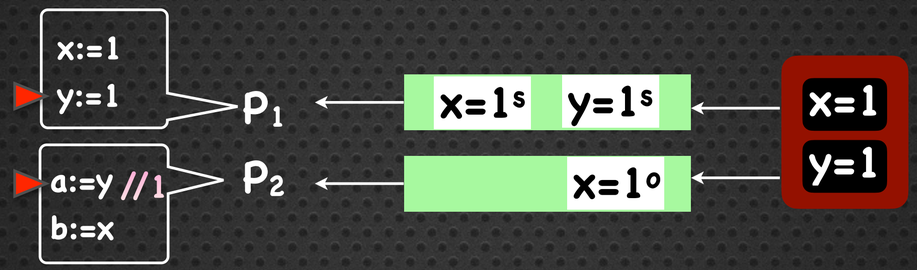
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

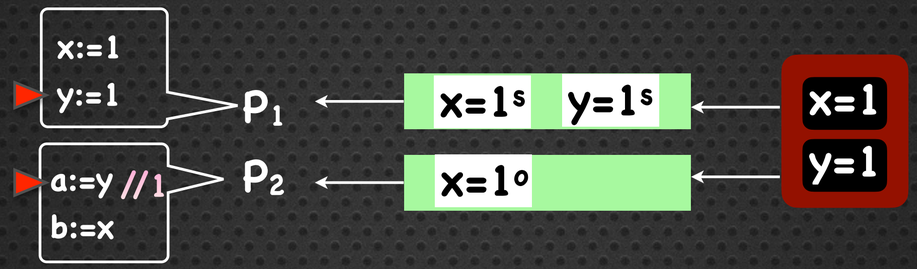
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

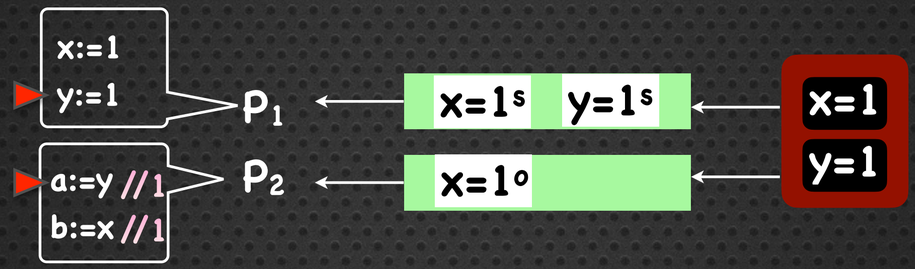
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

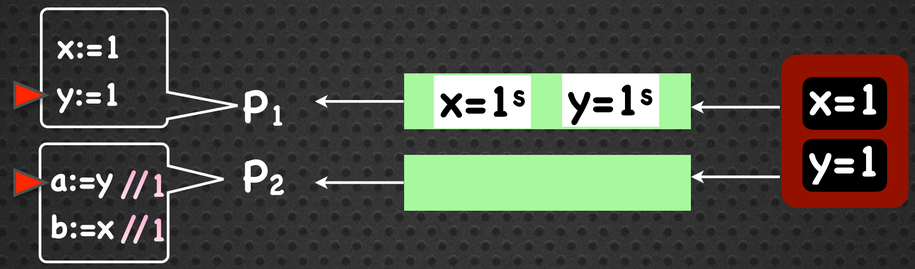
WALLPAPERSWIDE.COM

Shared Variable Concurrency

Total Store Order (TSO)

Load-Buffer Semantics

Message Passing



a=1 and b=0

potentially bad behaviors

extra behaviors

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics (Verification)

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics (Verification)
  - Equivalence with Store-Buffer Semantics
  - Monotonicity and WQO

WALLPAPERSWIDE.COM

Shared Variable Concurrency

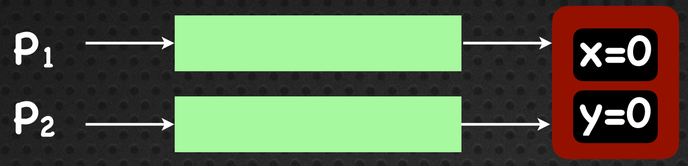
Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity

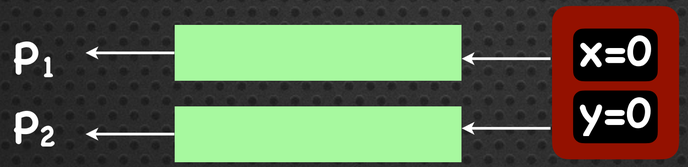
- Load-Buffer Semantics (Verification)
  - Equivalence with Store-Buffer Semantics
  - Monotonicity and WQO

WALLPAPERSWIDE.COM

Store-Buffer Semantics

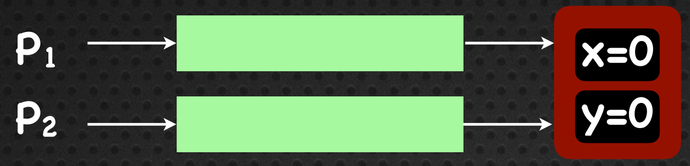


Load-Buffer Semantics



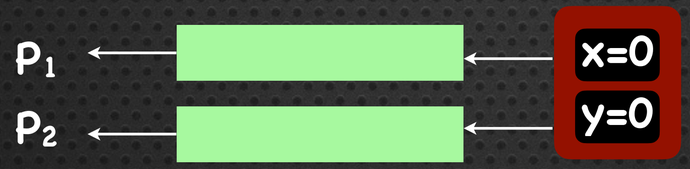
WALLPAPERSWIDE.COM

Store-Buffer Semantics



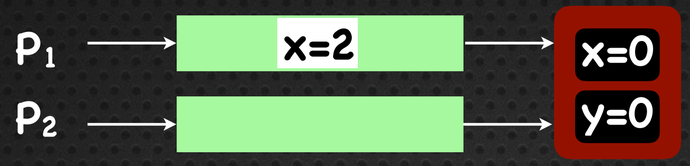
P1: w(x,2)

Load-Buffer Semantics



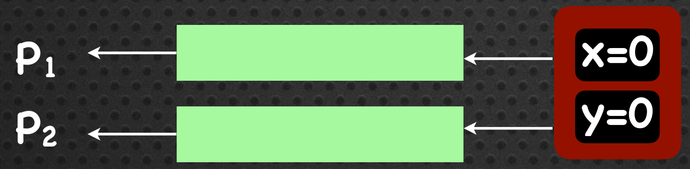
WALLPAPERSWIDE.COM

Store-Buffer Semantics



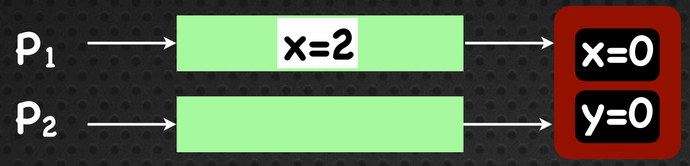
P1: w(x,2)

Load-Buffer Semantics



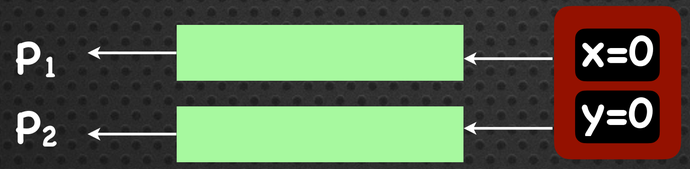
WALLPAPERSWIDE.COM

Store-Buffer Semantics



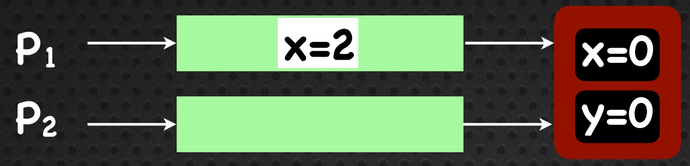
P1: w(x,2) P1: r(y,0)

Load-Buffer Semantics



WALLPAPERSWIDE.COM

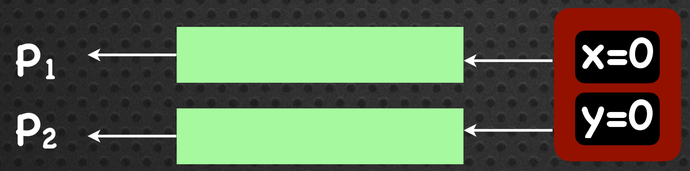
Store-Buffer Semantics



P_1: w(x,2)

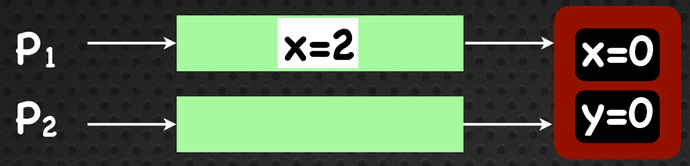
P_1: r(y,0)

Load-Buffer Semantics



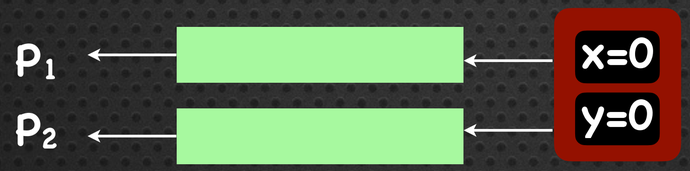
WALLPAPERSWIDE.COM

Store-Buffer Semantics



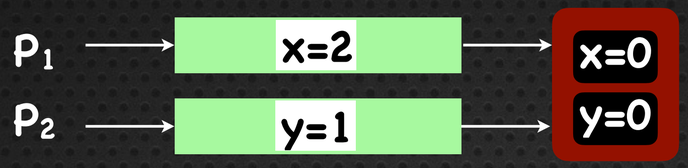
P_1: w(x,2) P_1: r(y,0) P_2: w(y,1)

Load-Buffer Semantics



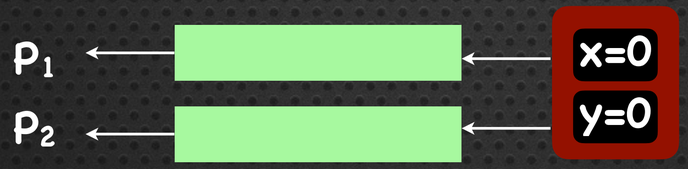
WALLPAPERSWIDE.COM

Store-Buffer Semantics



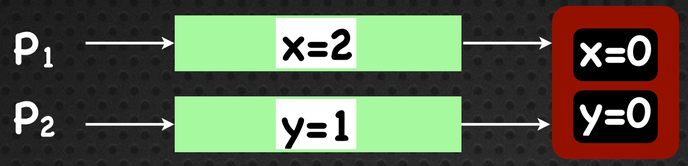
P_1: w(x,2) P_1: r(y,0) P_2: w(y,1)

Load-Buffer Semantics



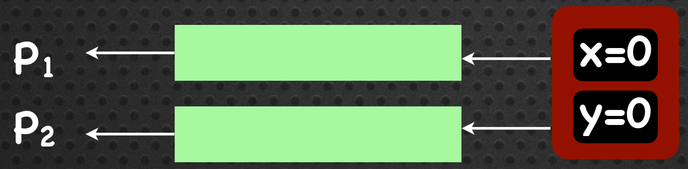
WALLPAPERSWIDE.COM

Store-Buffer Semantics



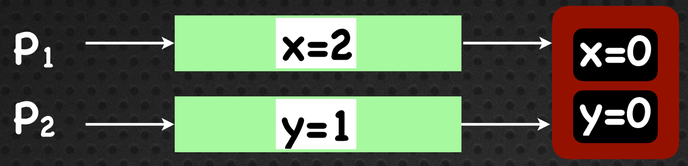
P1: w(x,2) | P1: r(y,0) | P2: w(y,1)

Load-Buffer Semantics



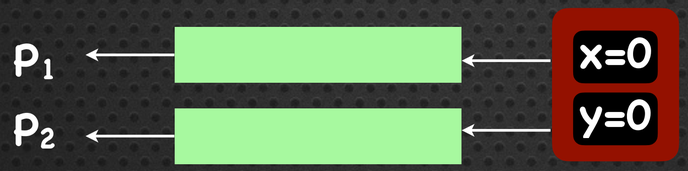
WALLPAPERSWIDE.COM

Store-Buffer Semantics



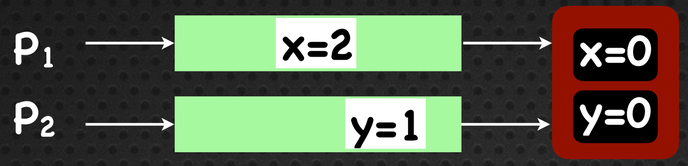
P_1: w(x,2) | P_1: r(y,0) | P_2: w(y,1) | P_2: w(x,1)

Load-Buffer Semantics



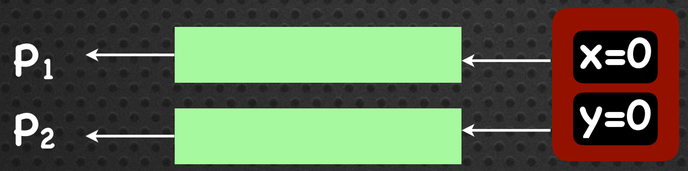
WALLPAPERSWIDE.COM

Store-Buffer Semantics



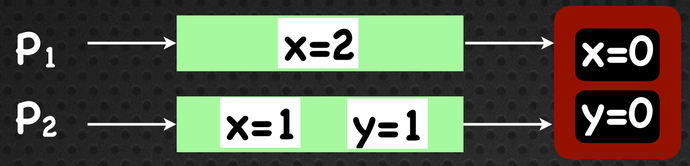
P_1: w(x,2) | P_1: r(y,0) | P_2: w(y,1) | P_2: w(x,1)

Load-Buffer Semantics



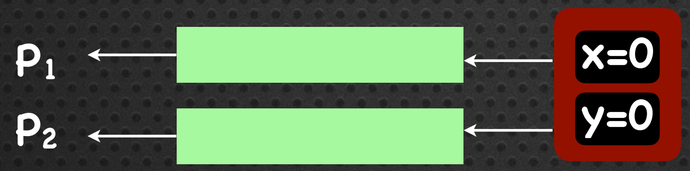
WALLPAPERSWIDE.COM

Store-Buffer Semantics



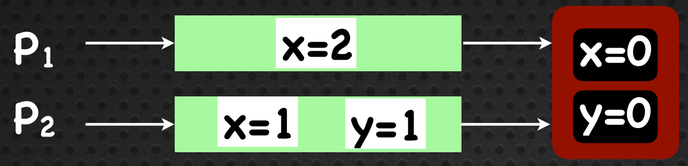
P_1: w(x,2) | P_1: r(y,0) | P_2: w(y,1) | P_2: w(x,1)

Load-Buffer Semantics



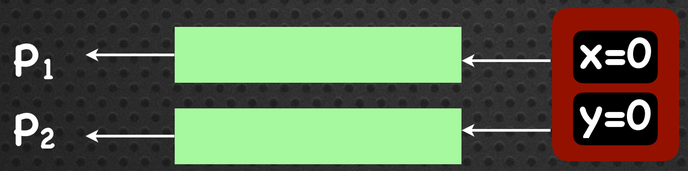
WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



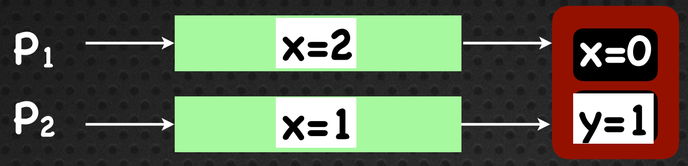
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1)

Load-
Buffer
Semantics



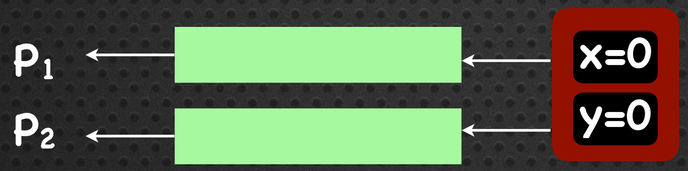
WALLPAPERSWIDE.COM

Store-Buffer Semantics



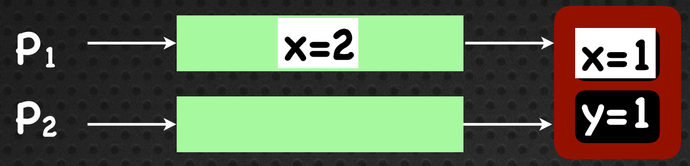
P_1: w(x,2) | P_1: r(y,0) | P_2: w(y,1) | P_2: w(x,1)

Load-Buffer Semantics



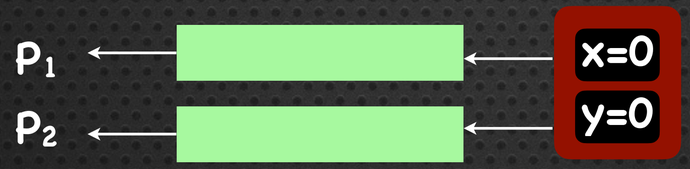
WALLPAPERSWIDE.COM

Store-Buffer Semantics



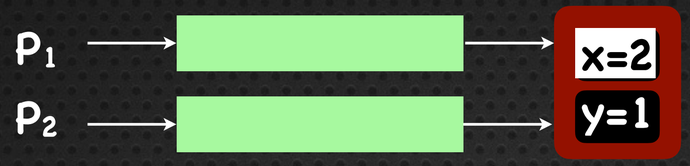
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1)

Load-Buffer Semantics



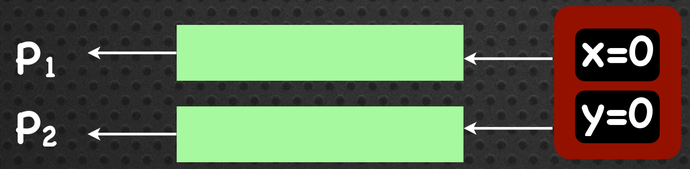
WALLPAPERSWIDE.COM

Store-Buffer Semantics



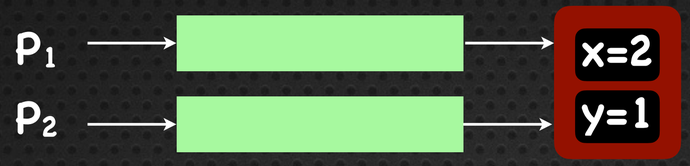
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1)

Load-Buffer Semantics



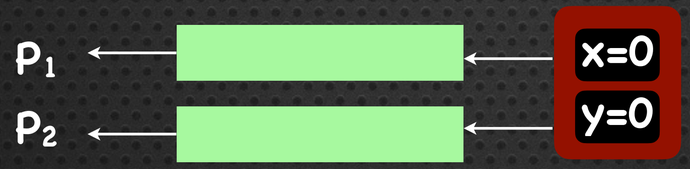
WALLPAPERSWIDE.COM

Store-Buffer Semantics



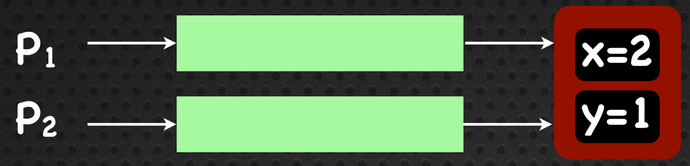
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1)

Load-Buffer Semantics



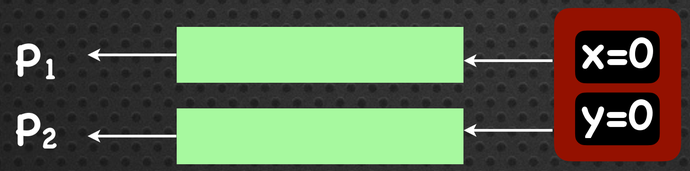
WALLPAPERSWIDE.COM

Store-Buffer Semantics



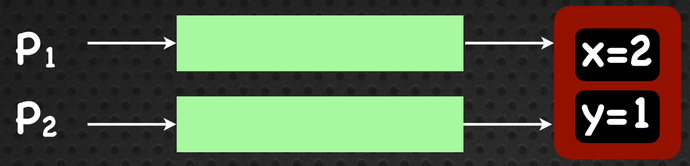
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Load-Buffer Semantics



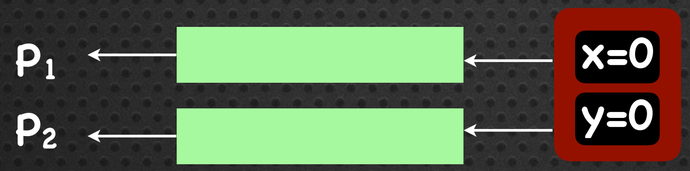
WALLPAPERSWIDE.COM

Store-Buffer Semantics



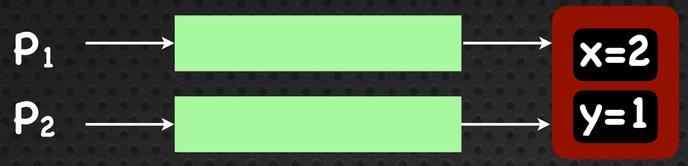
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Load-Buffer Semantics



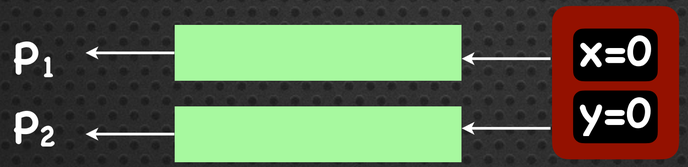
WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



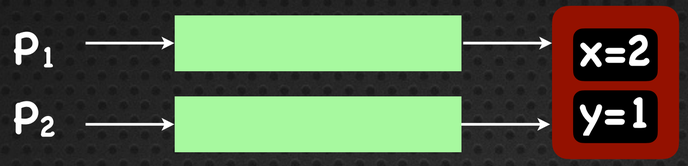
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Load-
Buffer
Semantics



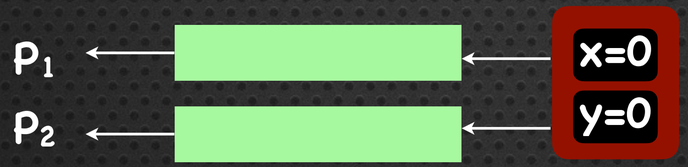
WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



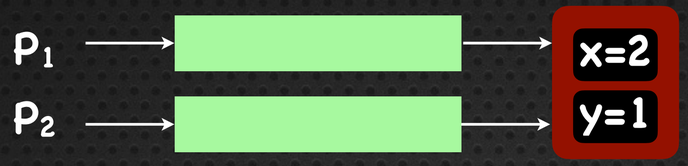
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Load-
Buffer
Semantics



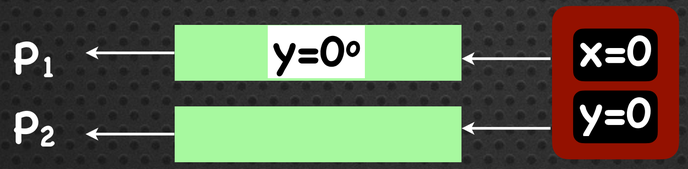
WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



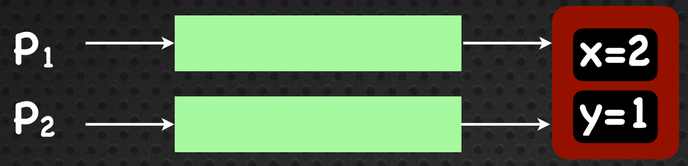
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Load-
Buffer
Semantics



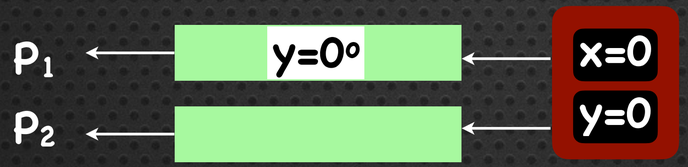
WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



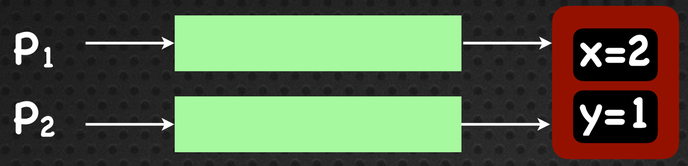
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Load-
Buffer
Semantics



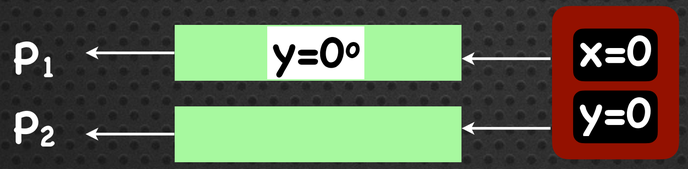
WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

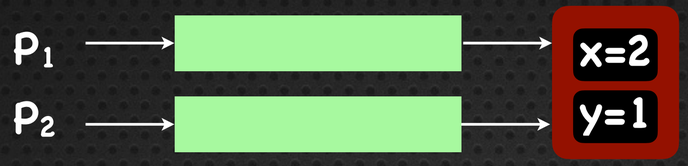
Load-
Buffer
Semantics



P2: w(y,1)

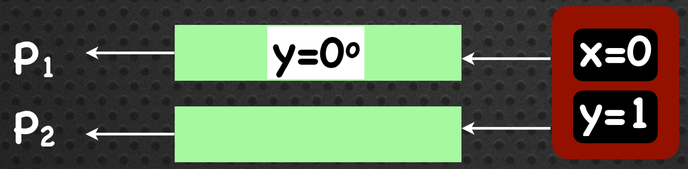
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

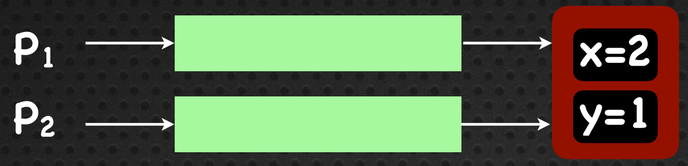
Load-Buffer Semantics



P2: w(y,1)

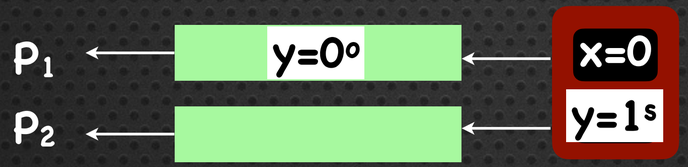
WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

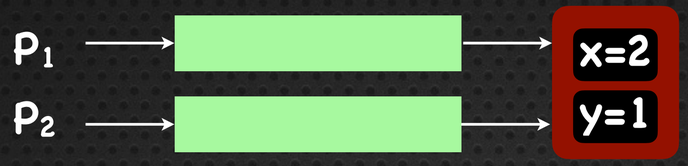
Load-
Buffer
Semantics



P2: w(y,1)

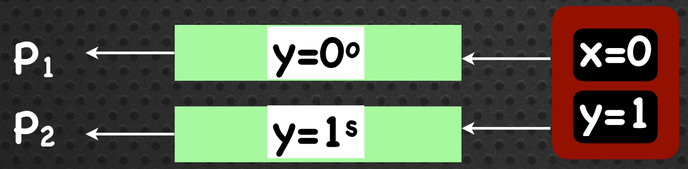
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

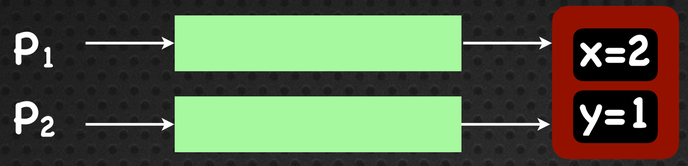
Load-Buffer Semantics



P2: w(y,1)

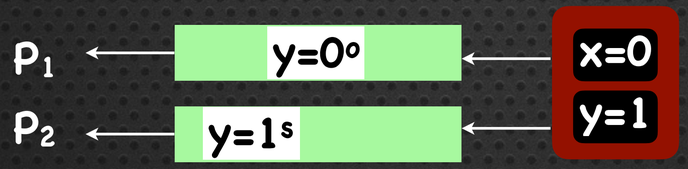
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

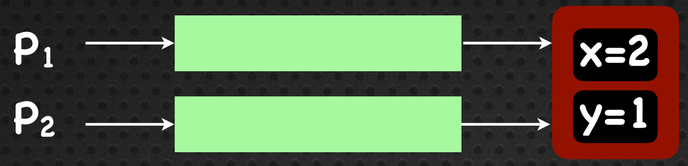
Load-Buffer Semantics



P2: w(y,1)

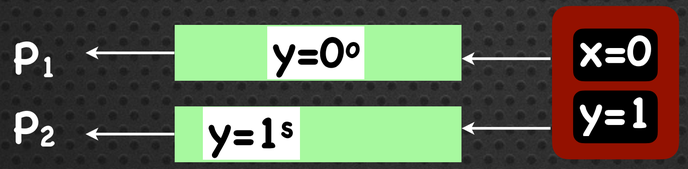
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

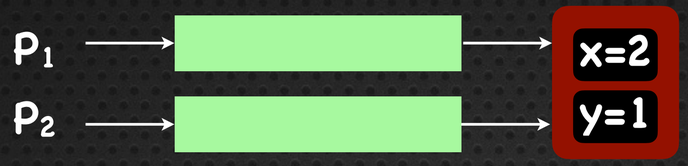
Load-Buffer Semantics



P2: w(y,1) P2: w(x,1)

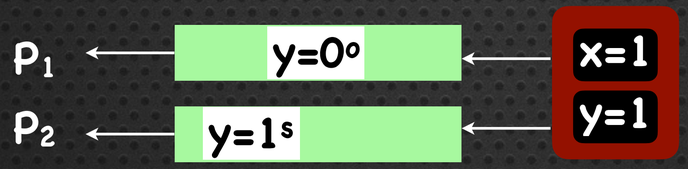
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

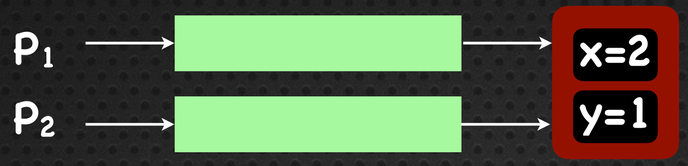
Load-Buffer Semantics



P2: w(y,1) P2: w(x,1)

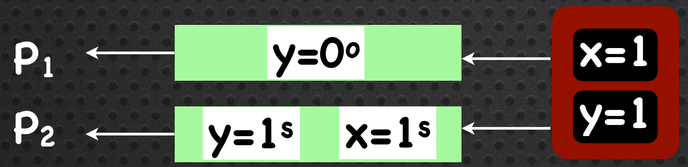
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

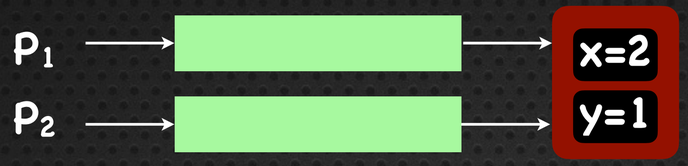
Load-Buffer Semantics



P2: w(y,1) P2: w(x,1)

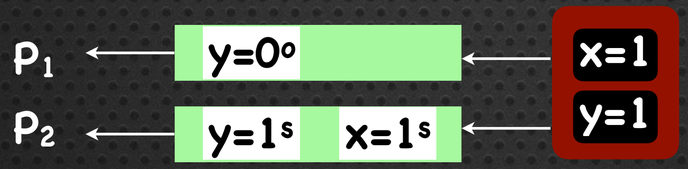
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

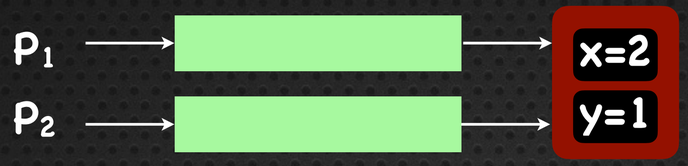
Load-Buffer Semantics



P2: w(y,1) P2: w(x,1)

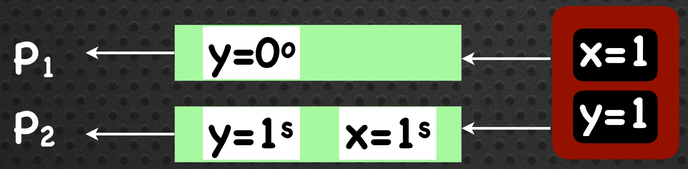
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

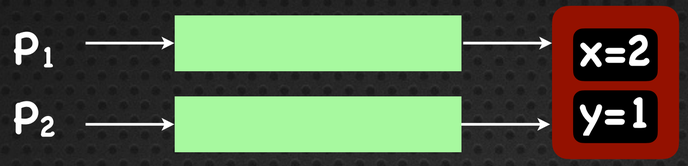
Load-Buffer Semantics



P2: w(y,1) P2: w(x,1) P1: w(x,2)

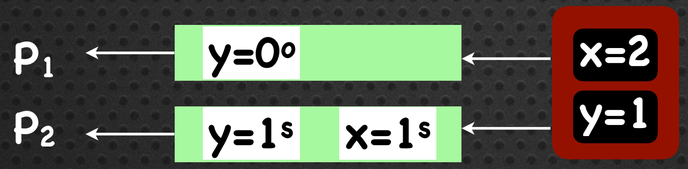
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

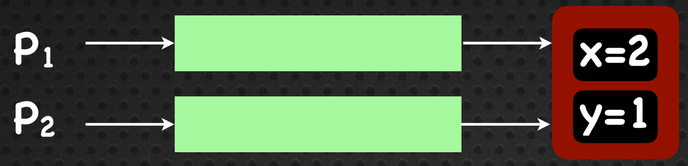
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2)

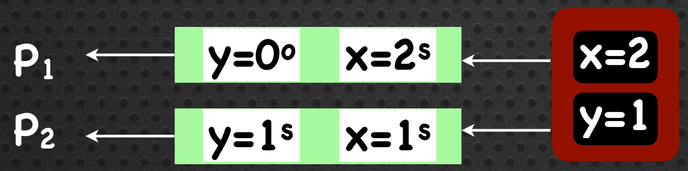
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

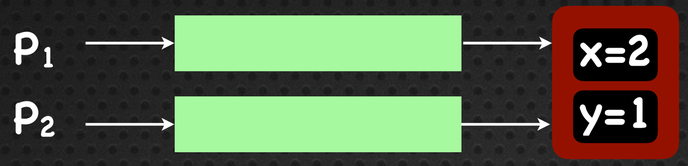
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2)

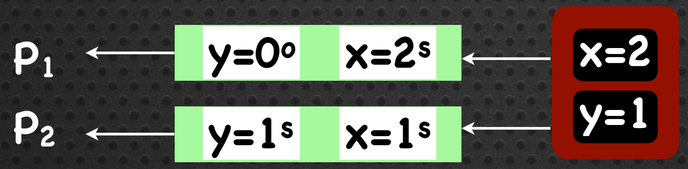
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

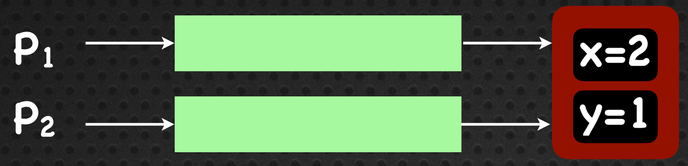
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2)

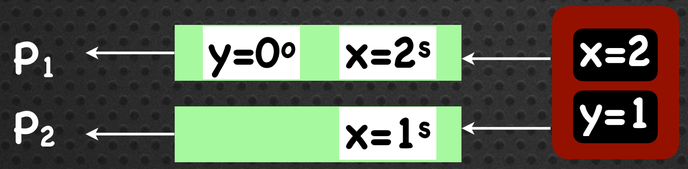
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

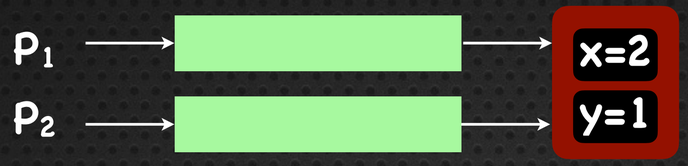
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2)

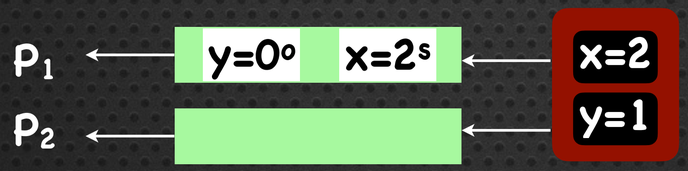
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

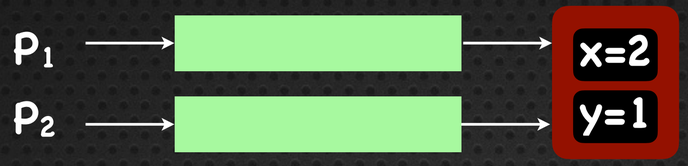
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2)

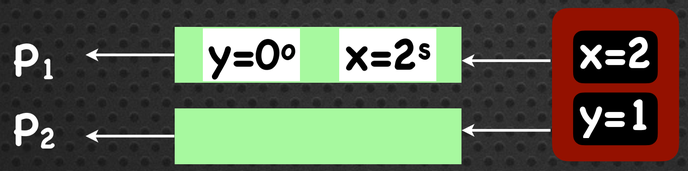
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

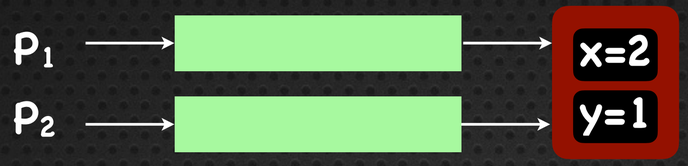
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2)

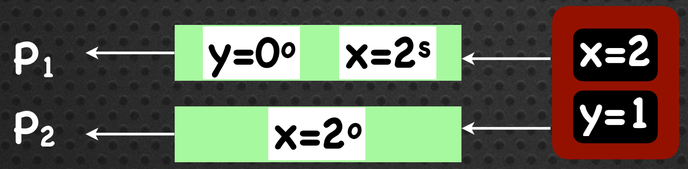
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

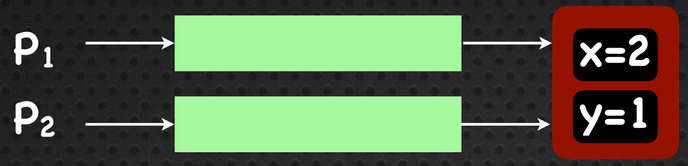
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2)

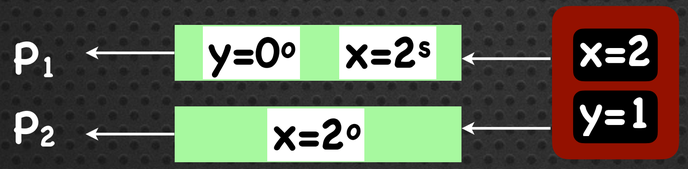
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

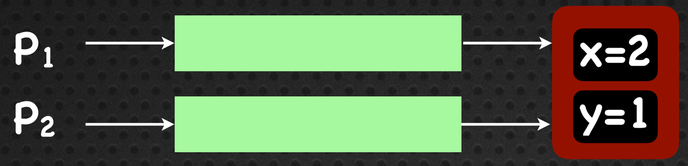
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2)

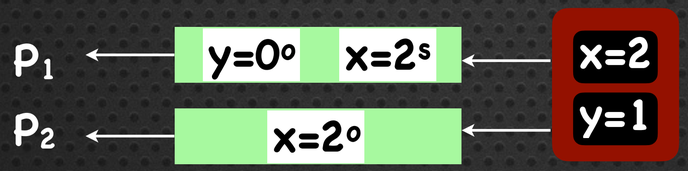
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

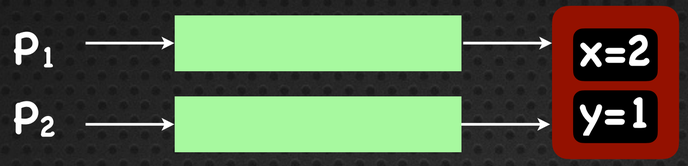
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2)

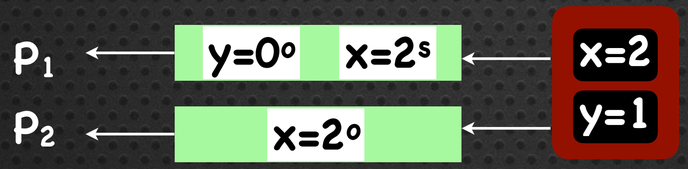
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

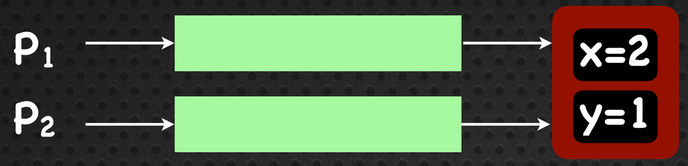
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

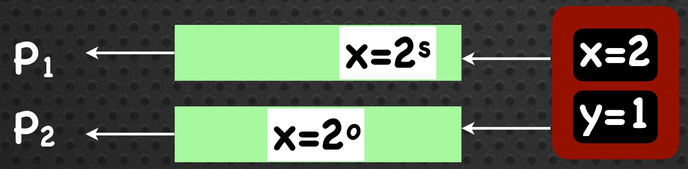
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

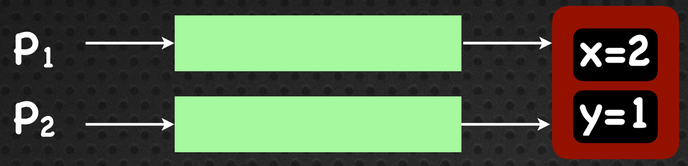
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

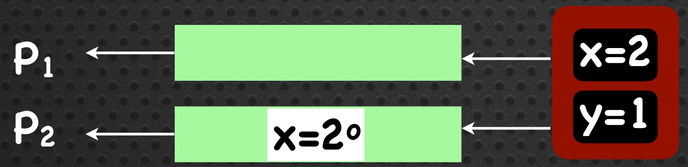
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

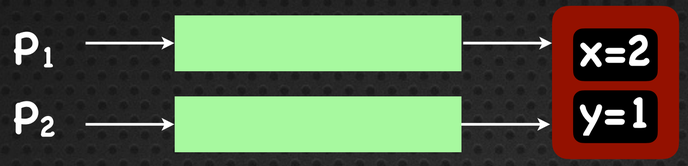
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

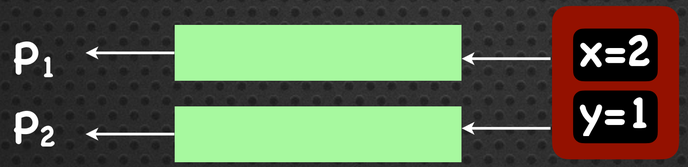
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

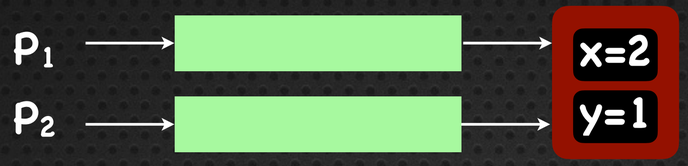
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

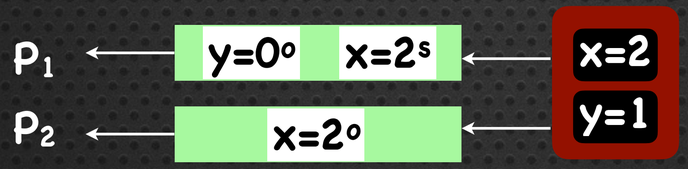
WALLPAPERSWIDE.COM

Store-Buffer Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

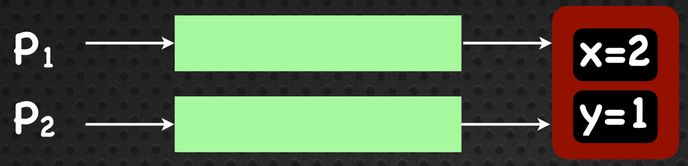
Load-Buffer Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

WALLPAPERSWIDE.COM

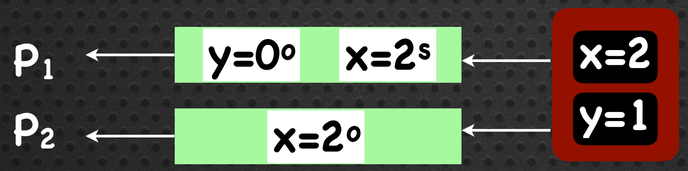
Store-
Buffer
Semantics



P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Store-
Buffer
Semantics

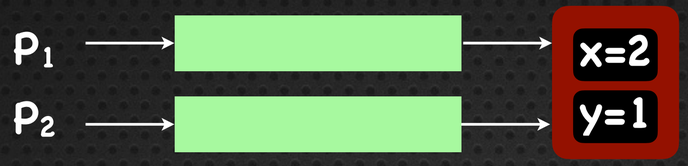
Load-
Buffer
Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

WALLPAPERSWIDE.COM

Store-
Buffer
Semantics

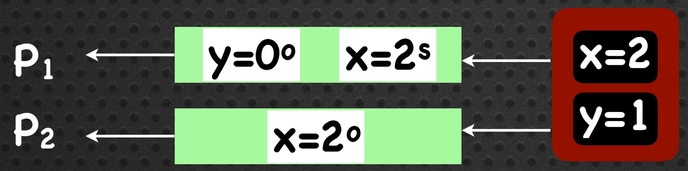


P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Store-
Buffer
Semantics

Load-
Buffer
Semantics

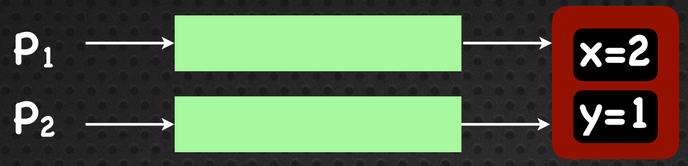
Load-
Buffer
Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

WALLPAPERSWIDE.COM

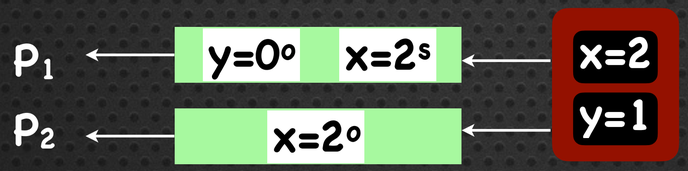
Store-Buffer Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

Store-Buffer Semantics = Load-Buffer Semantics

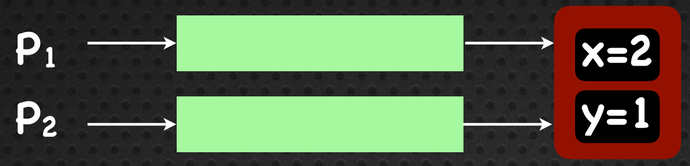
Load-Buffer Semantics



P2: w(y,1) P2: w(x,1) P1: w(x,2) P2: r(x,2) P1: r(y,0)

WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



"processes perform
identical
transition sequences"

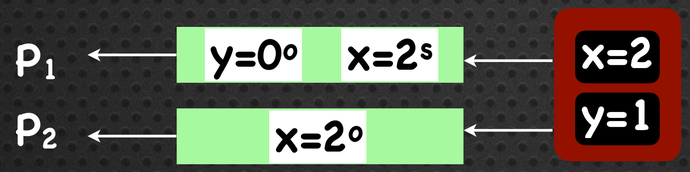
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Store-
Buffer
Semantics

≡

Load-
Buffer
Semantics

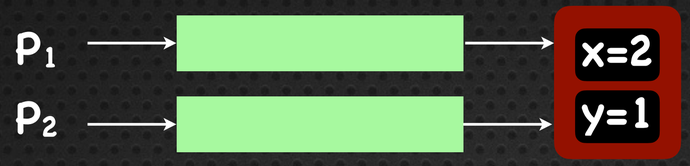
Load-
Buffer
Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

WALLPAPERSWIDE.COM

Store-
Buffer
Semantics

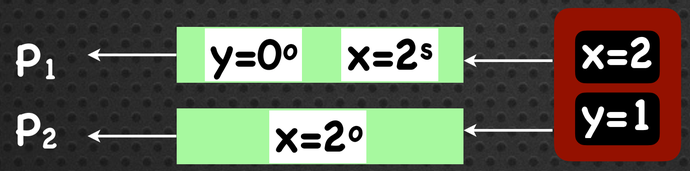


"processes perform
identical
transition sequences"

P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

Store-
Buffer
Semantics
≡
Load-
Buffer
Semantics

Load-
Buffer
Semantics



P2: w(y,1) P2: w(x,1) P1: w(x,2) P2: r(x,2) P1: r(y,0)

WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

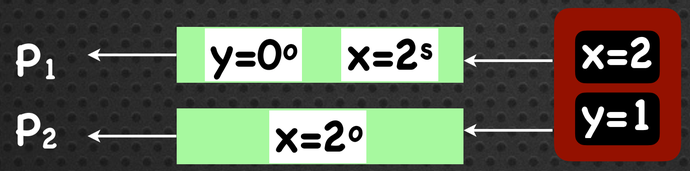
"processes perform
identical
transition sequences"

Store-
Buffer
Semantics

≡

Load-
Buffer
Semantics

Load-
Buffer
Semantics



P2: w(y,1) P2: w(x,1) P1: w(x,2) P2: r(x,2) P1: r(y,0)

WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

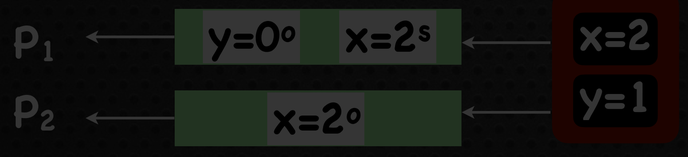
"processes perform
identical
transition sequences"

Store-
Buffer
Semantics

≡

Load-
Buffer
Semantics

Load-
Buffer
Semantics



P2: w(y,1) P2: w(x,1) P1: w(x,2) P2: r(x,2) P1: r(y,0)

WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



"processes perform
identical
transition sequences"

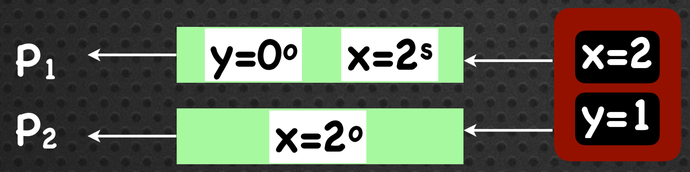
P1: w(x,2) | P1: r(y,0) | P2: w(y,1) | P2: w(x,1) | P2: r(x,2)

Store-
Buffer
Semantics

≡

Load-
Buffer
Semantics

Load-
Buffer
Semantics



P2: w(y,1) | P2: w(x,1) | P1: w(x,2) | P2: r(x,2) | P1: r(y,0)

WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

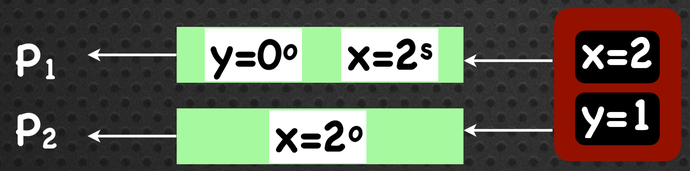
"processes perform
identical
transition sequences"

Store-
Buffer
Semantics

≡

Load-
Buffer
Semantics

Load-
Buffer
Semantics



P2: w(y,1) P2: w(x,1) P1: w(x,2) P2: r(x,2) P1: r(y,0)

WALLPAPERSWIDE.COM

Store-
Buffer
Semantics



P1: w(x,2) P1: r(y,0) P2: w(y,1) P2: w(x,1) P2: r(x,2)

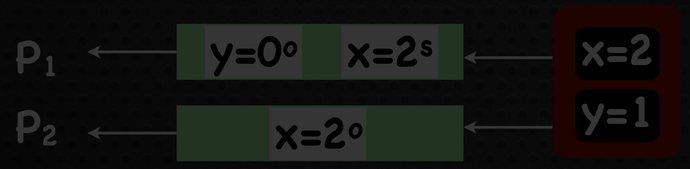
"processes perform
identical
transition sequences"

Store-
Buffer
Semantics

≡

Load-
Buffer
Semantics

Load-
Buffer
Semantics



P2: w(y,1) P2: w(x,1) P1: w(x,2) P2: r(x,2) P1: r(y,0)

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics (Verification)

WALLPAPERSWIDE.COM

Shared Variable Concurrency

Load-Buffer Semantics

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics (Verification)
  - Equivalence with Store-Buffer Semantics
  - Monotonicity and WQO

WALLPAPERSWIDE.COM

Shared Variable Concurrency

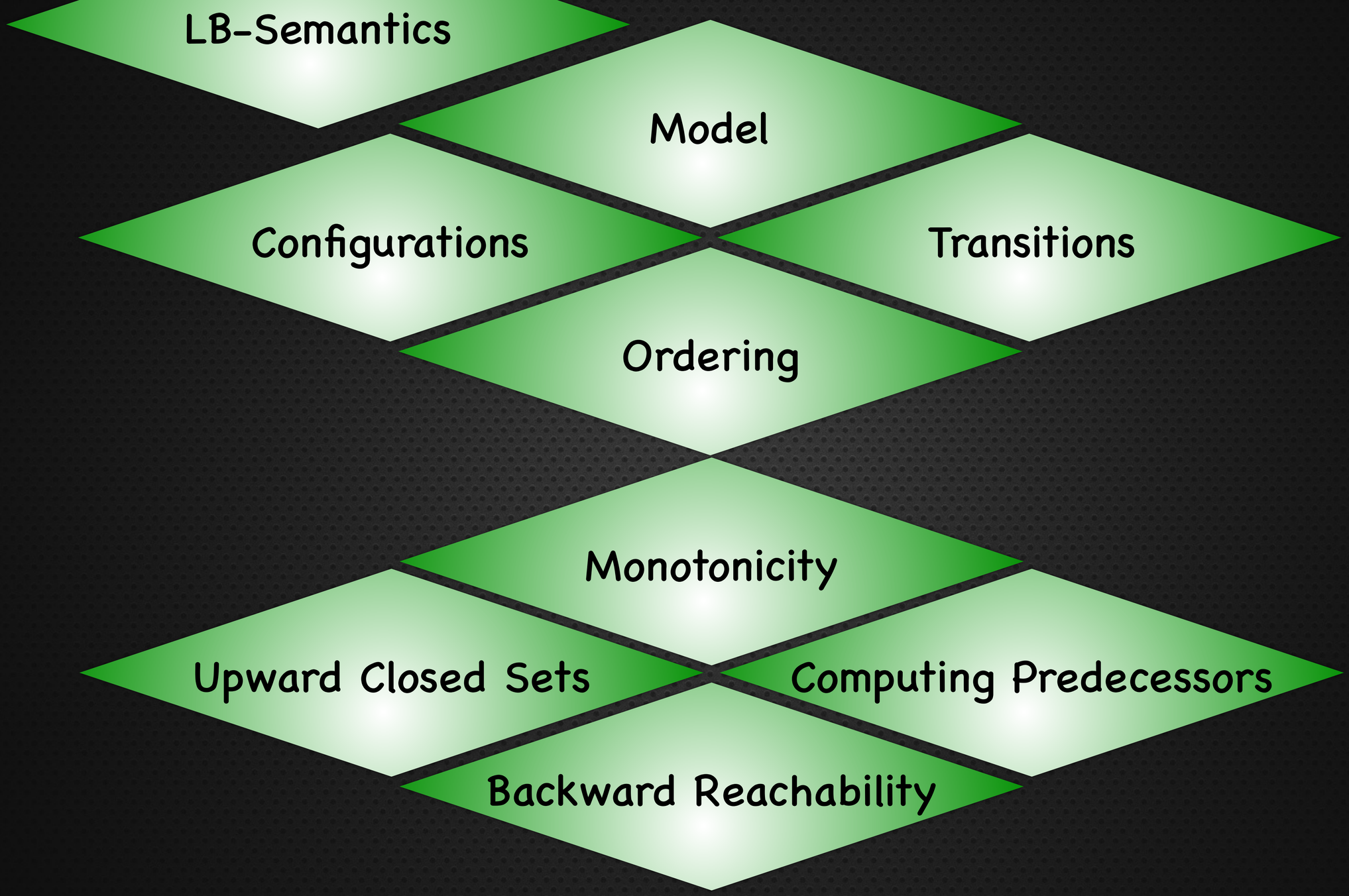
Load-Buffer Semantics

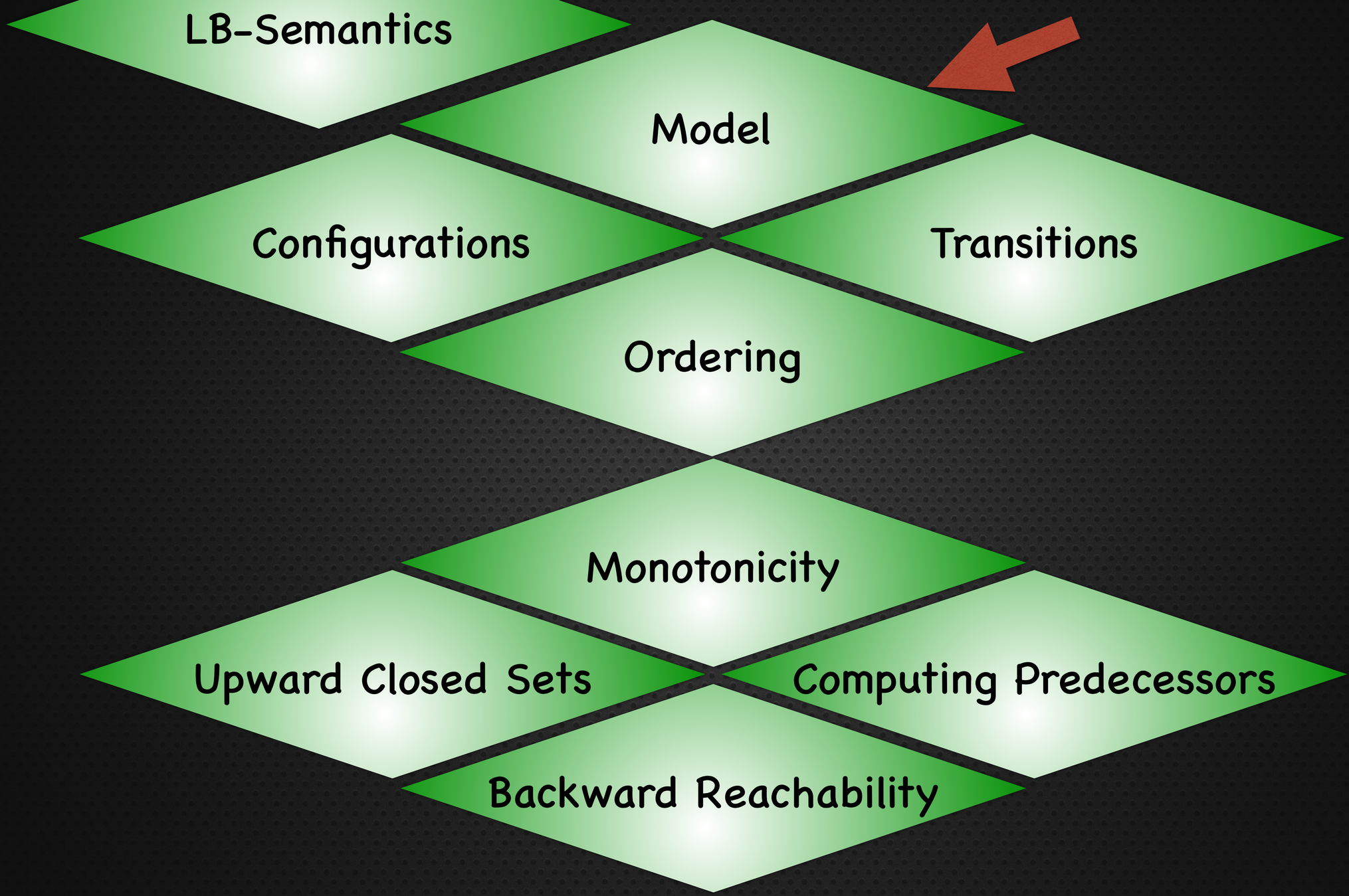
- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity

- Load-Buffer Semantics (Verification)
  - Equivalence with Store-Buffer Semantics
  - Monotonicity and WQO

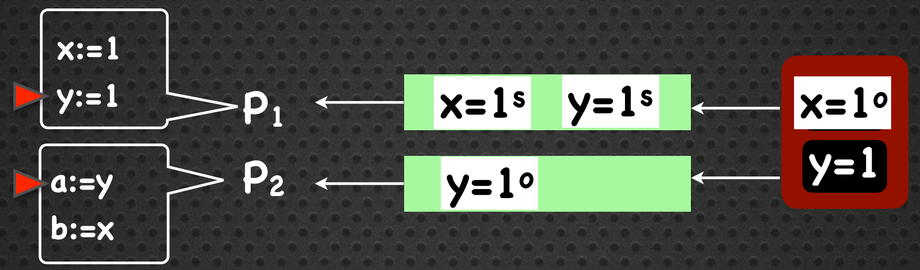


WALLPAPERSWIDE.COM


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability

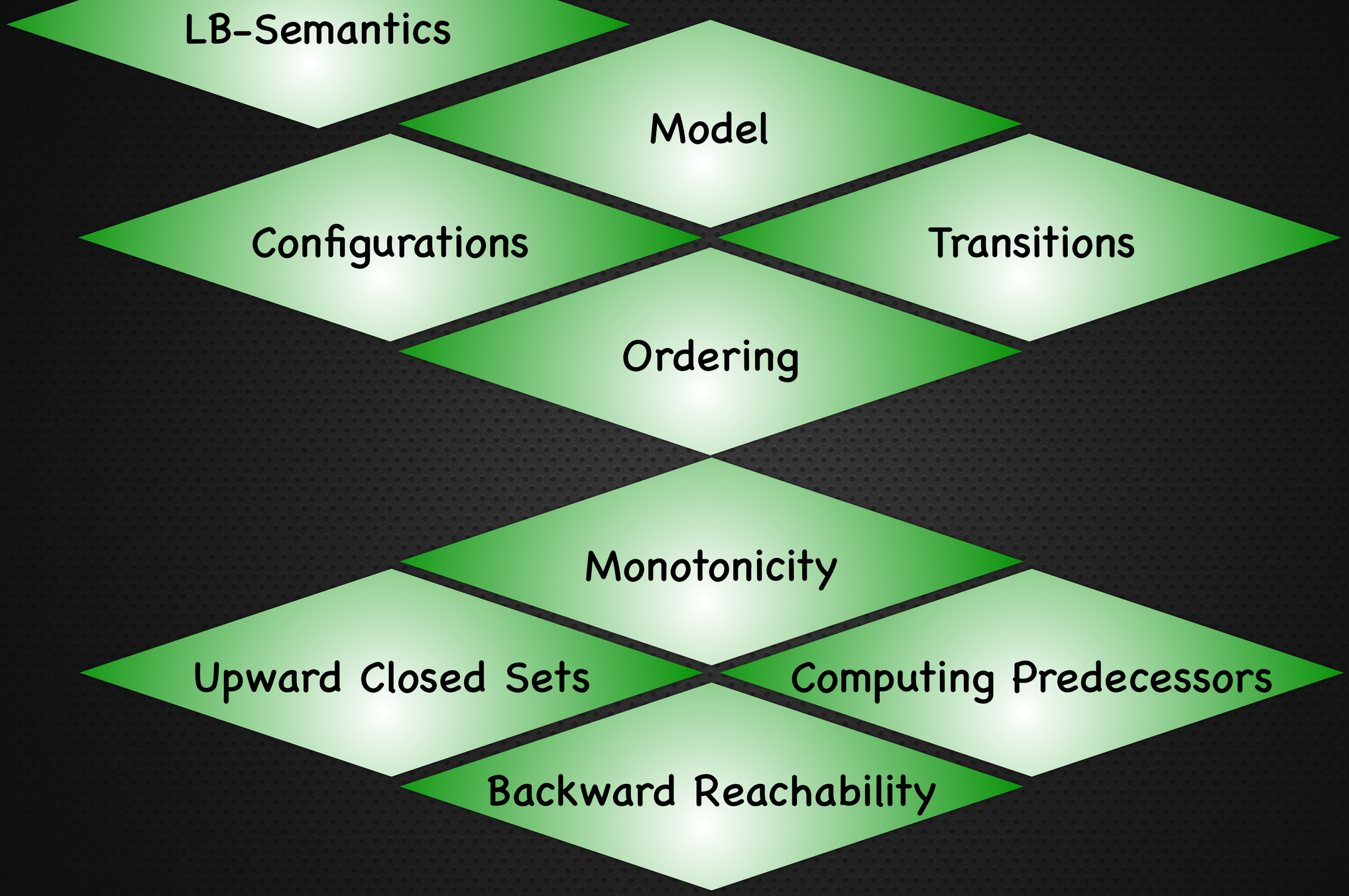

LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability

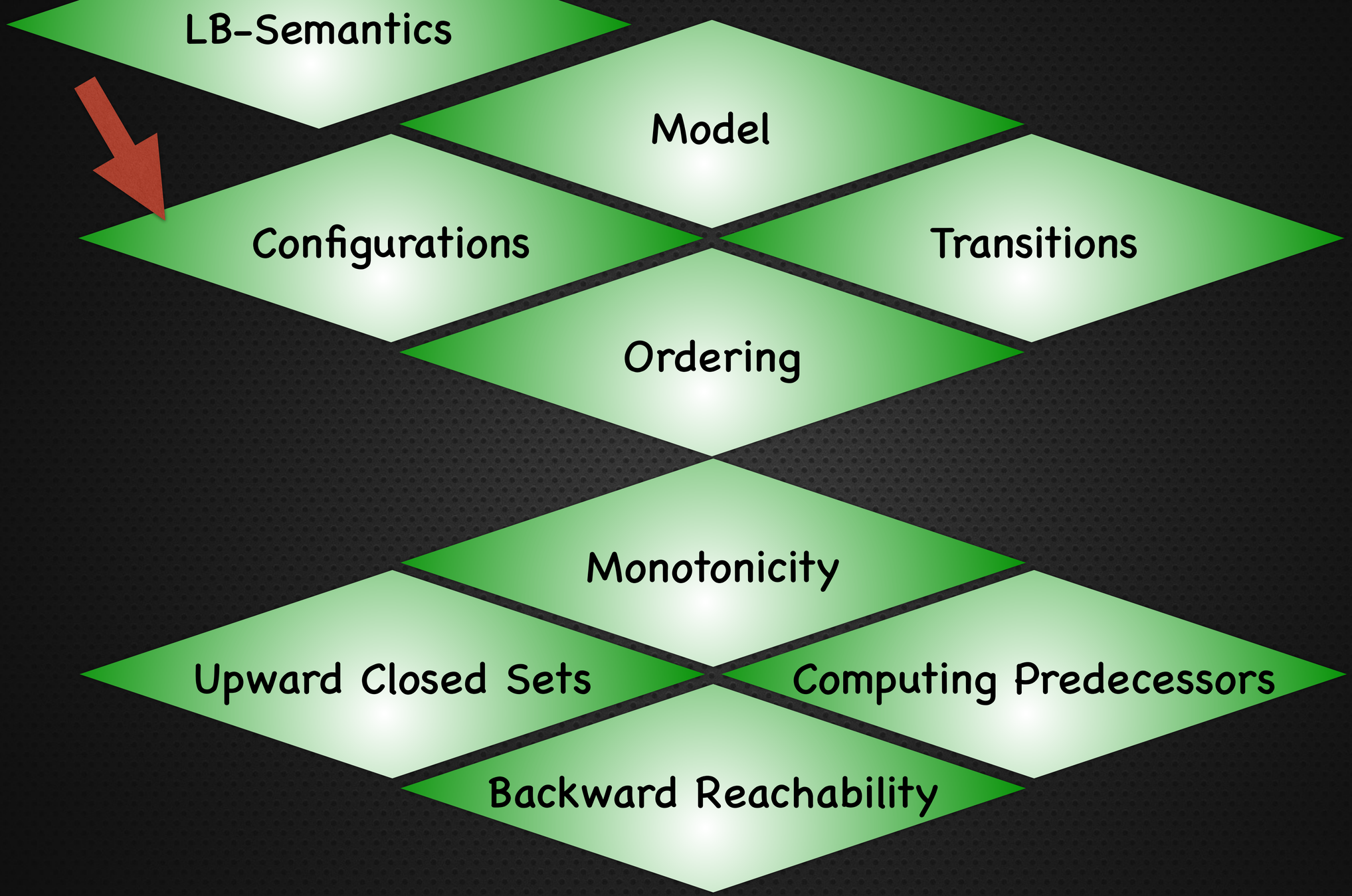
WALLPAPERSWIDE.COM



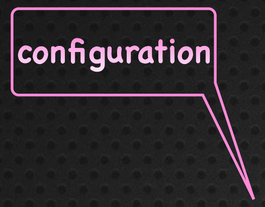
a=1 and b=0

WALLPAPERSWIDE.COM

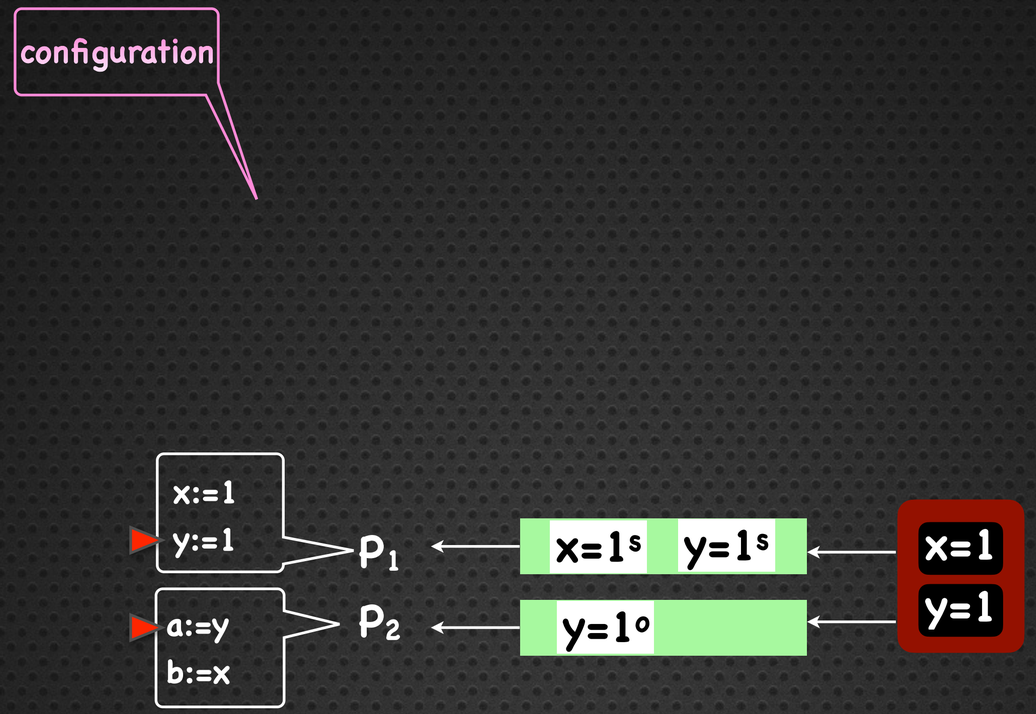

LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability

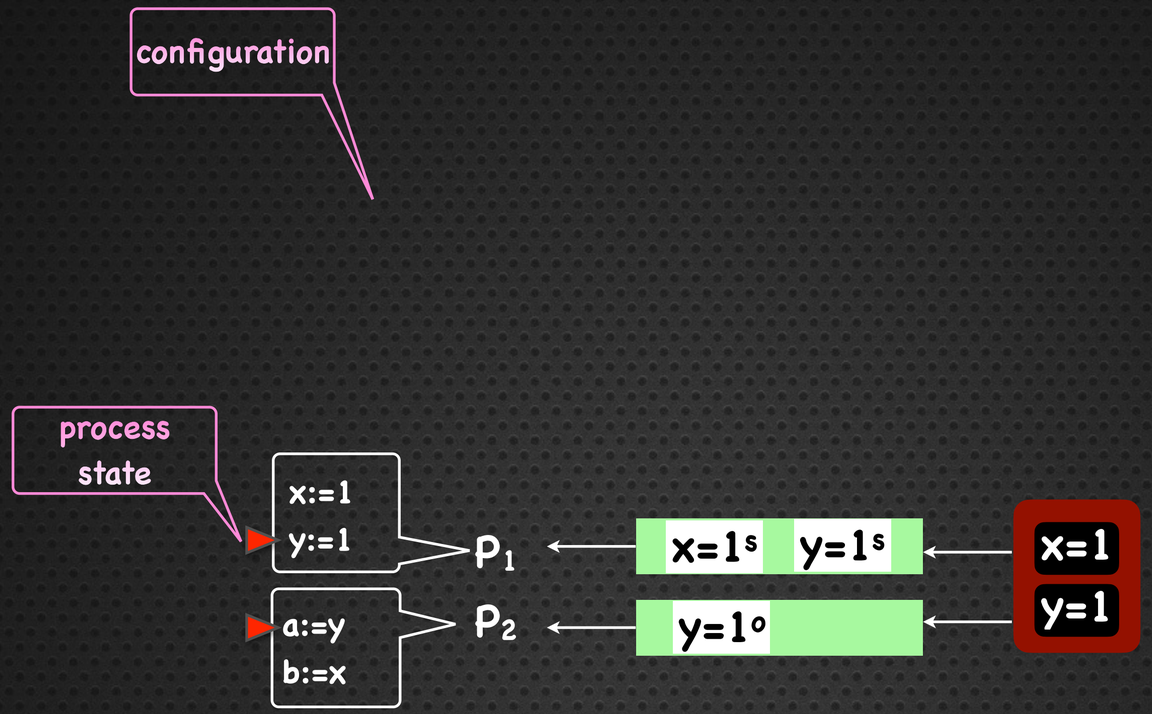
WALLPAPERSWIDE.COM



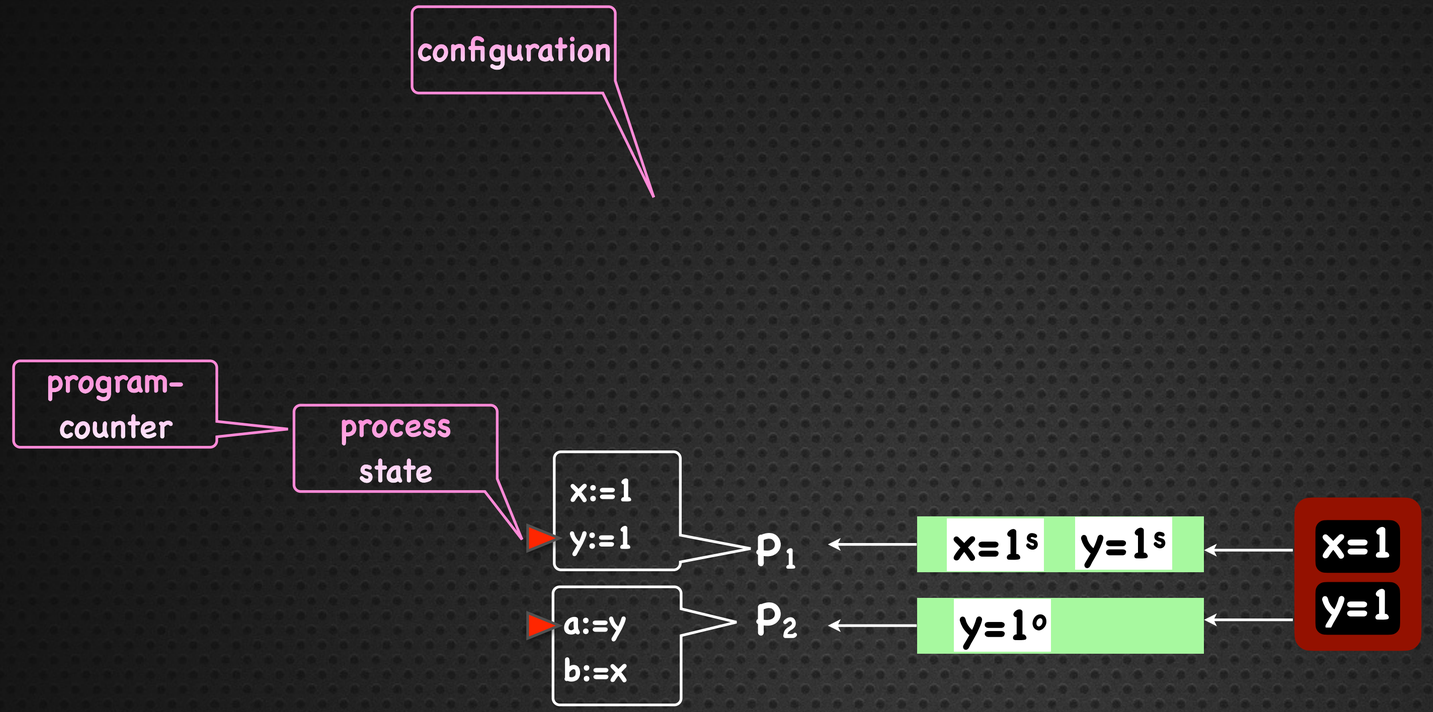
WALLPAPERSWIDE.COM



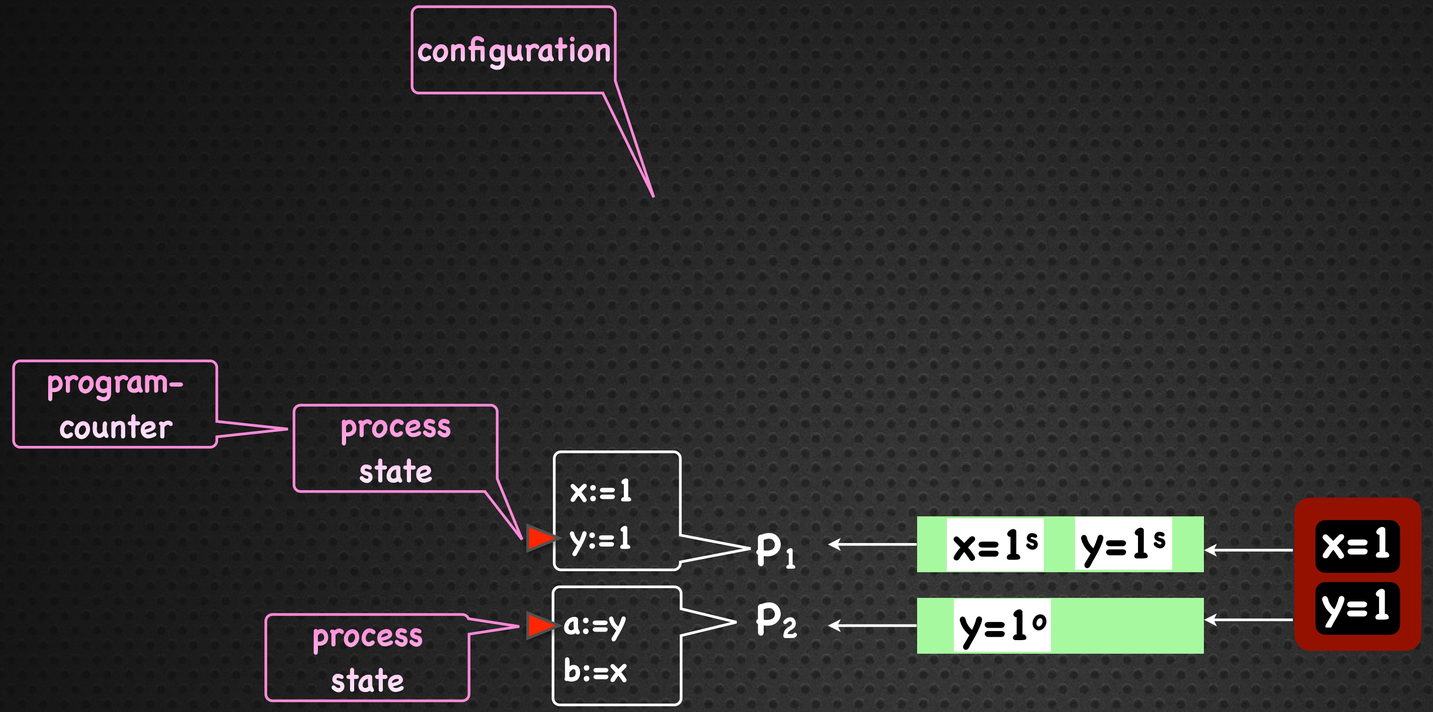
WALLPAPERSWIDE.COM



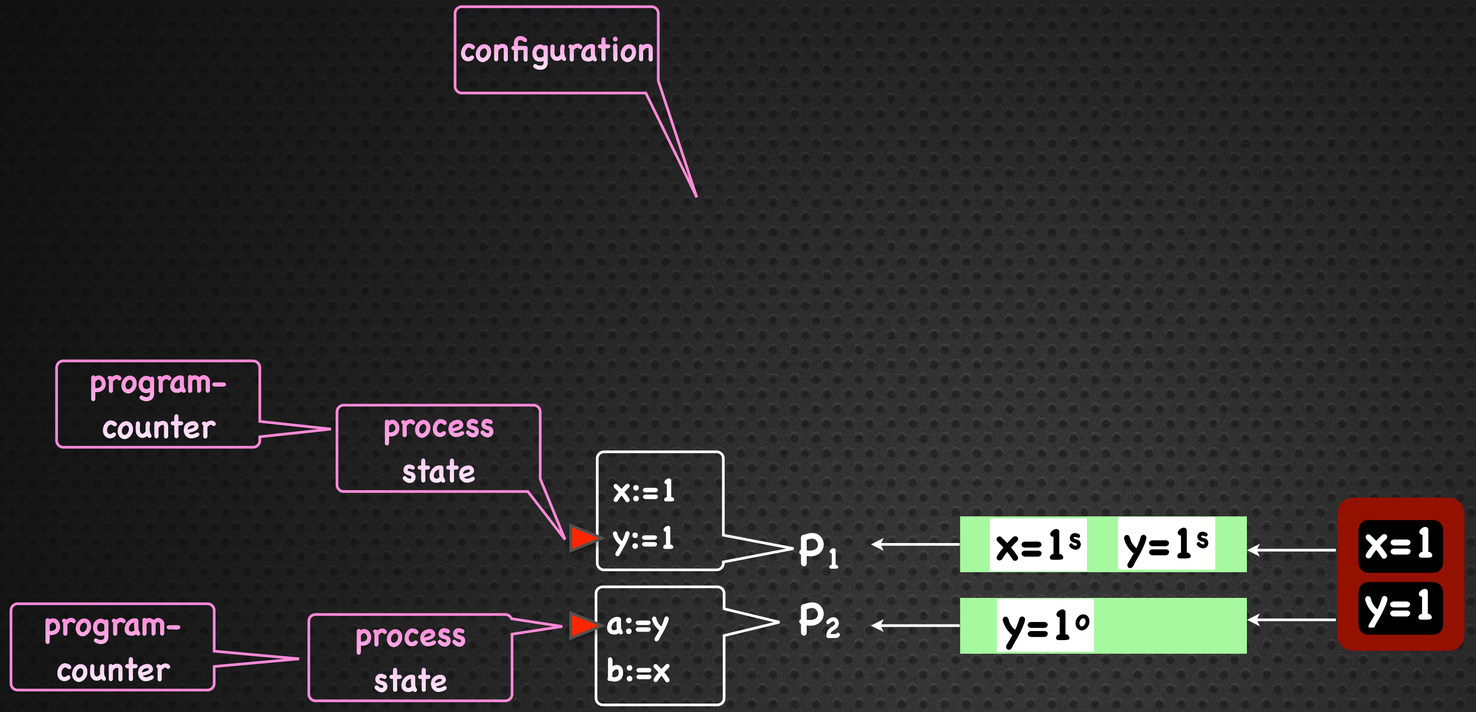
WALLPAPERSWIDE.COM

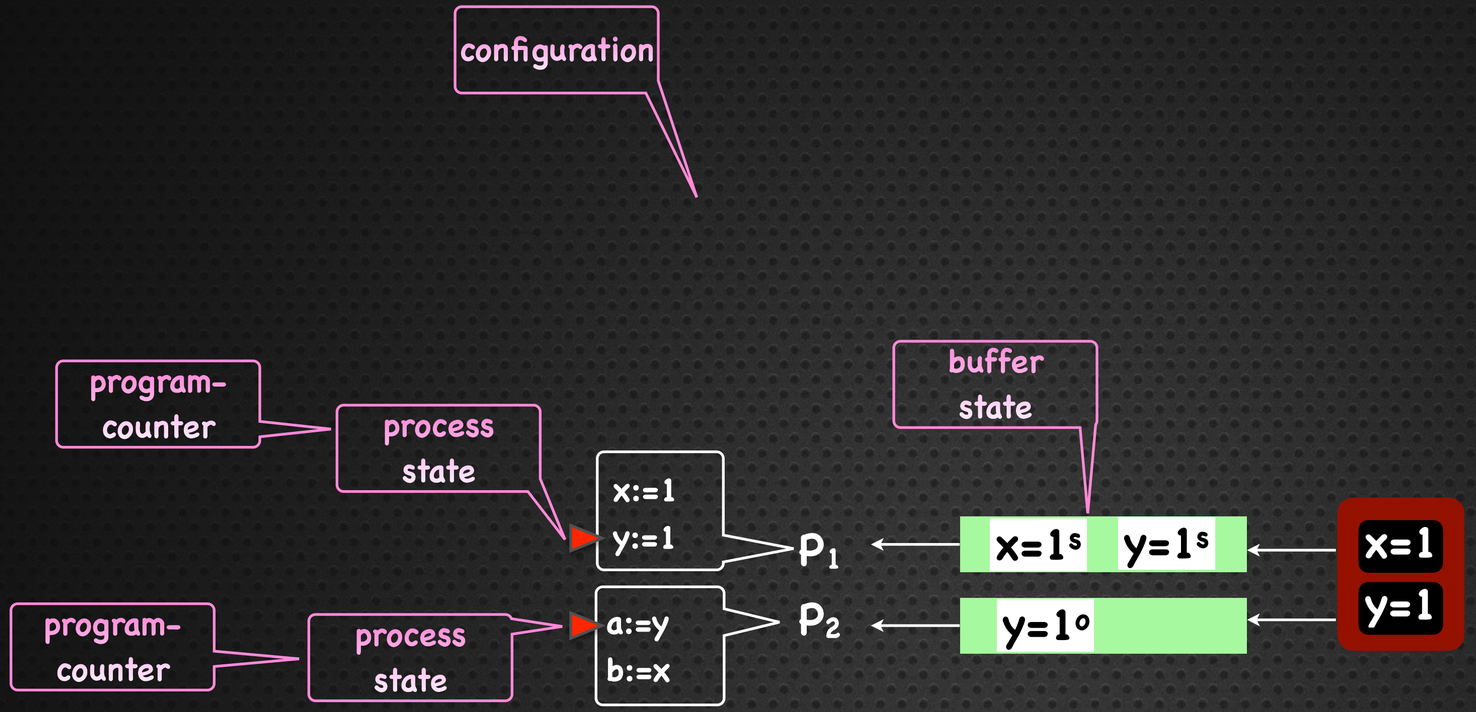


WALLPAPERSWIDE.COM

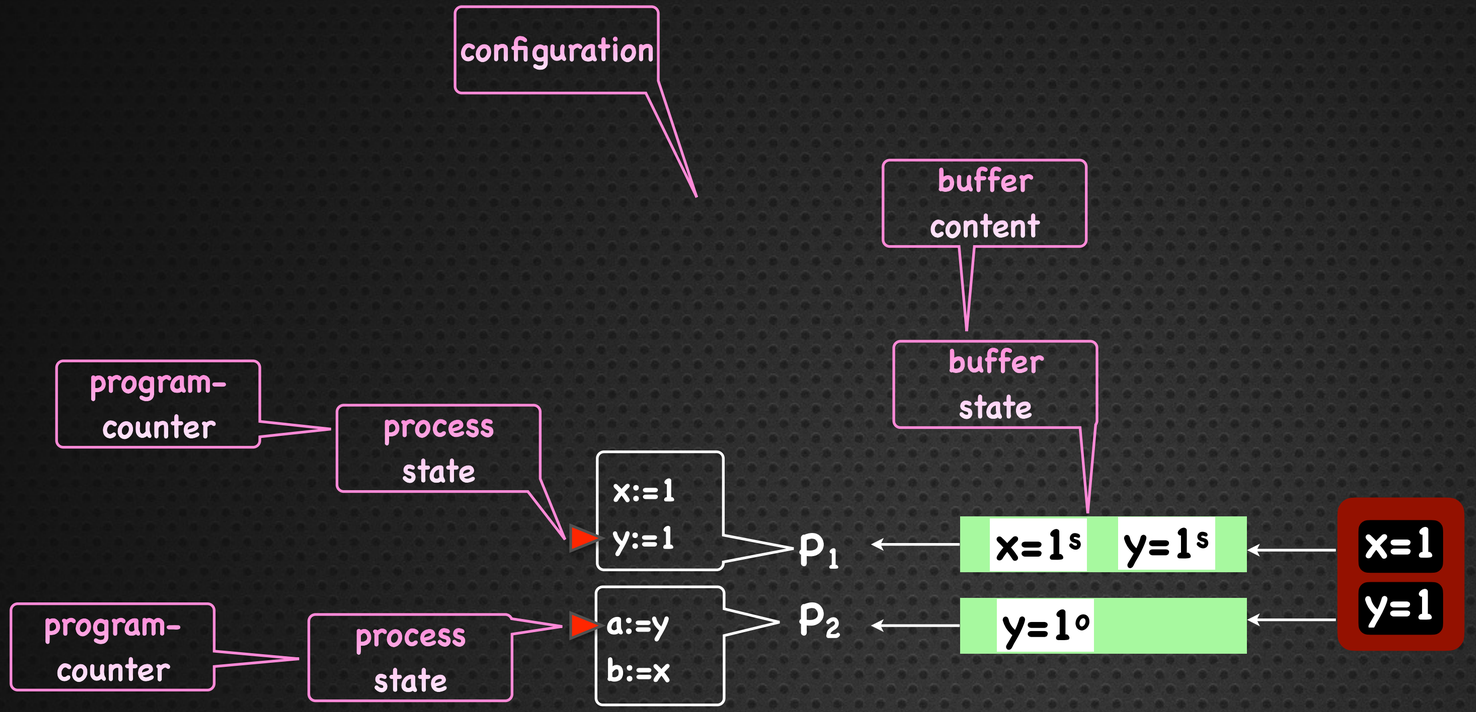


WALLPAPERSWIDE.COM

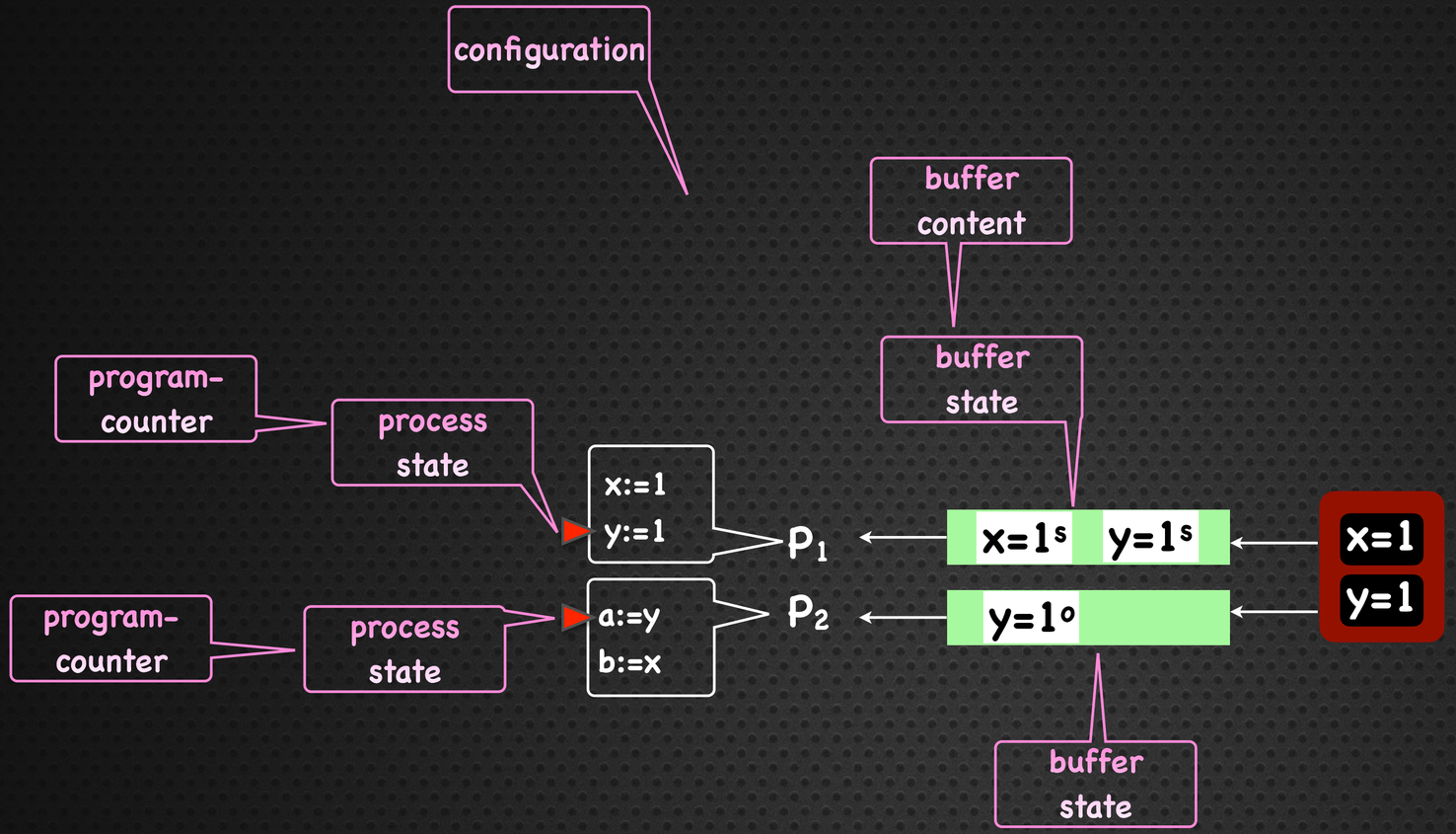

configuration
program-counter
process state
x:=1
y:=1
p1
x=1s
y=1s
x=1
y=1
program-counter
process state
a:=y
b:=x
p2
y=1o



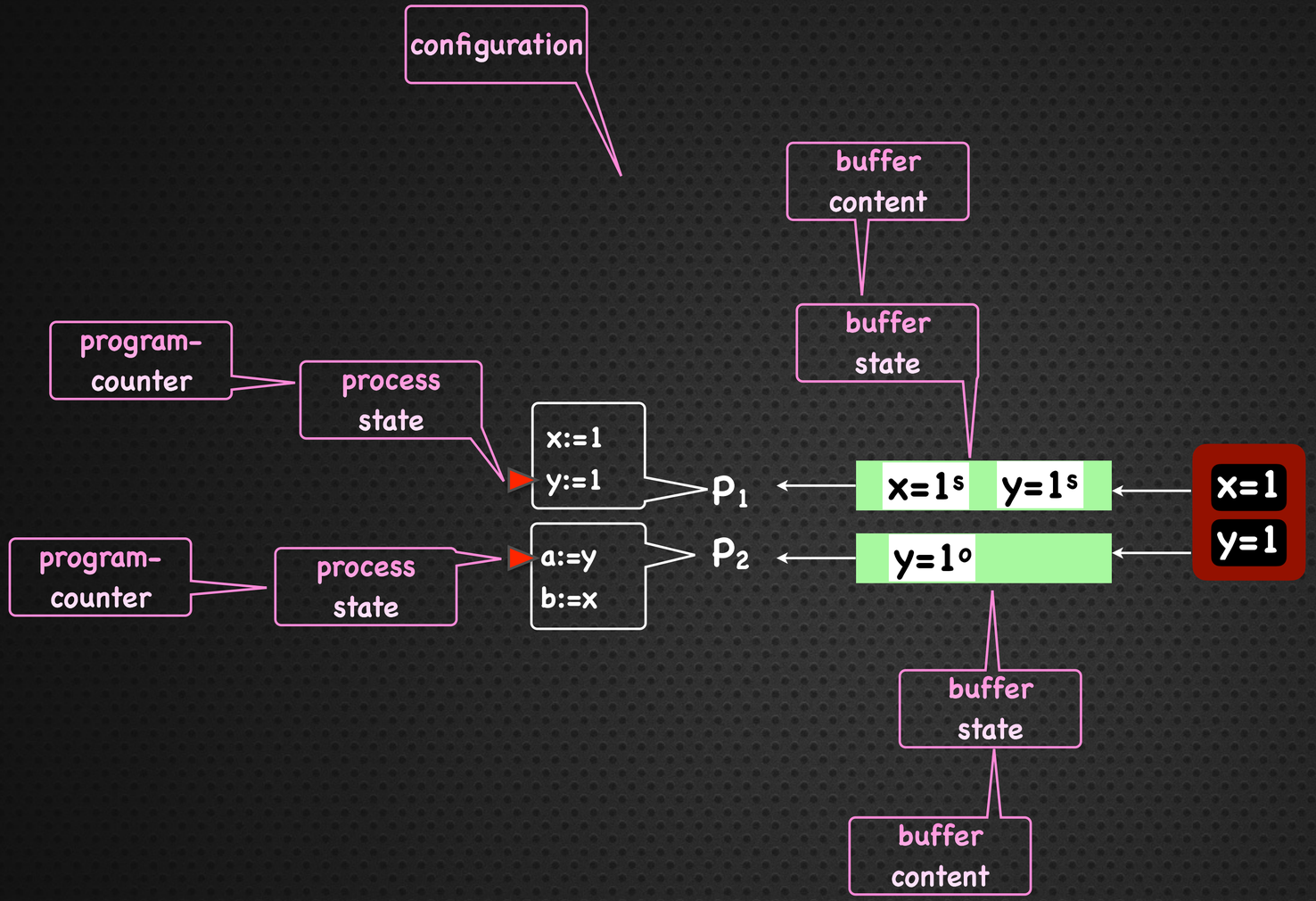
WALLPAPERSWIDE.COM



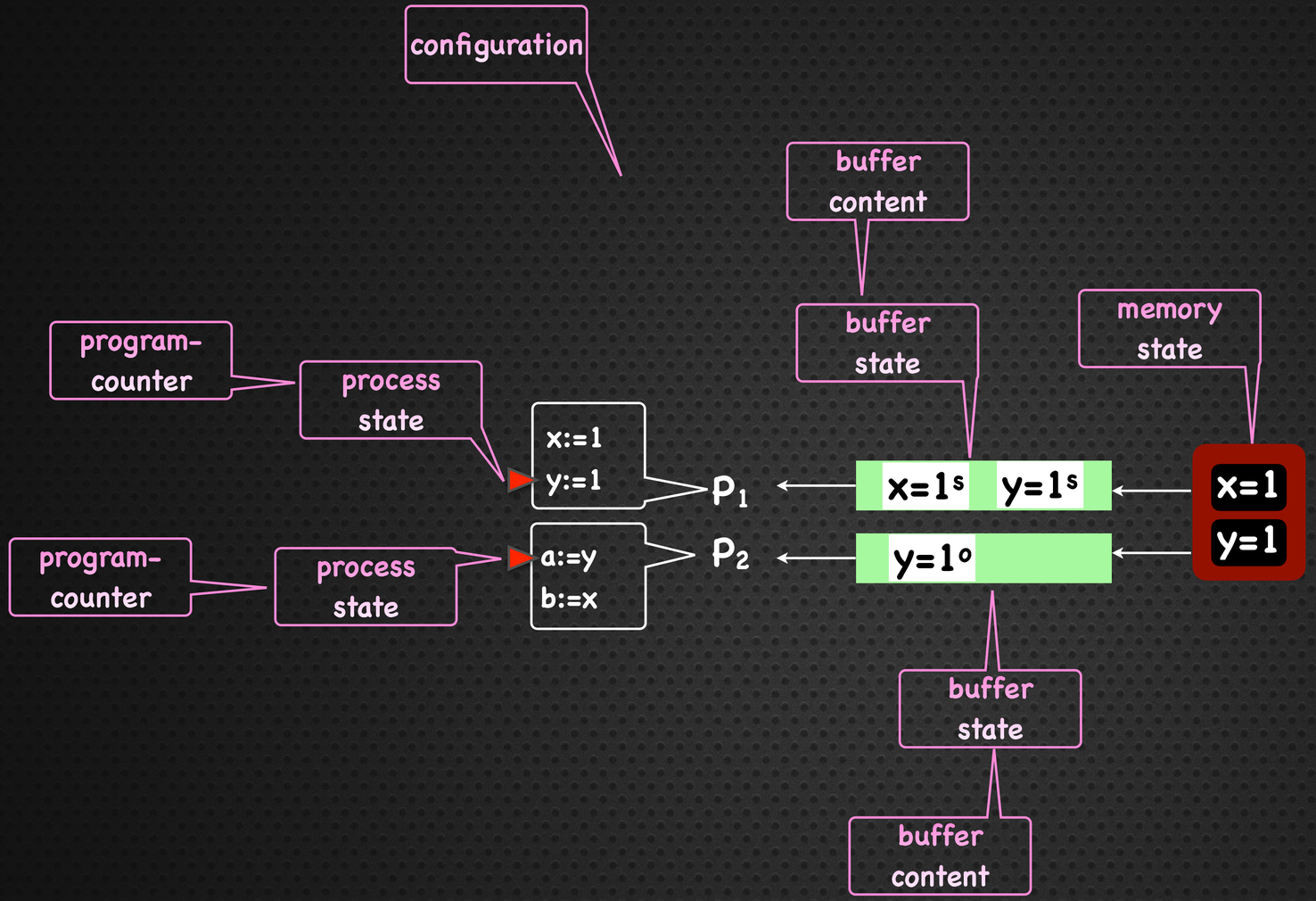
WALLPAPERSWIDE.COM



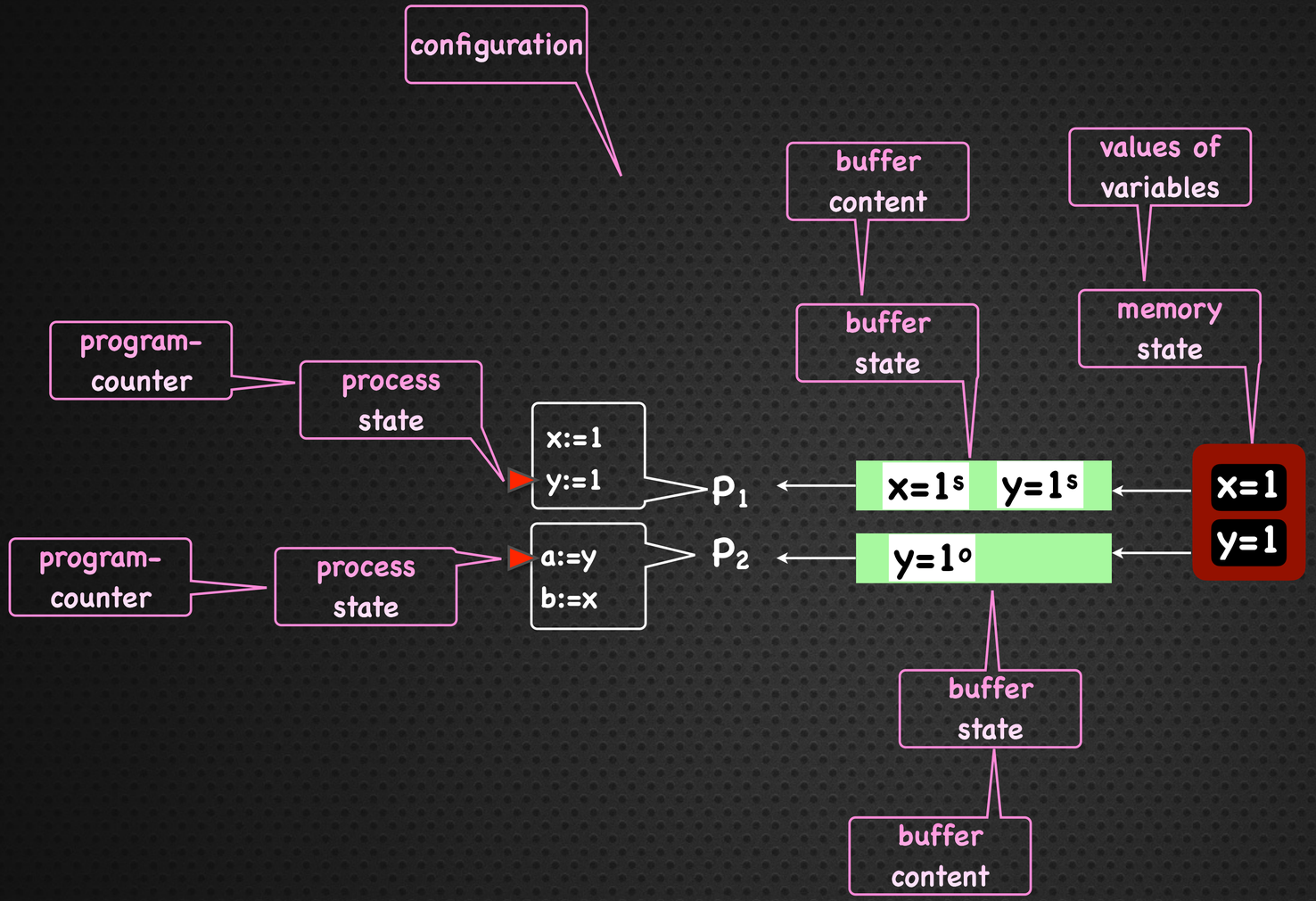
WALLPAPERSWIDE.COM



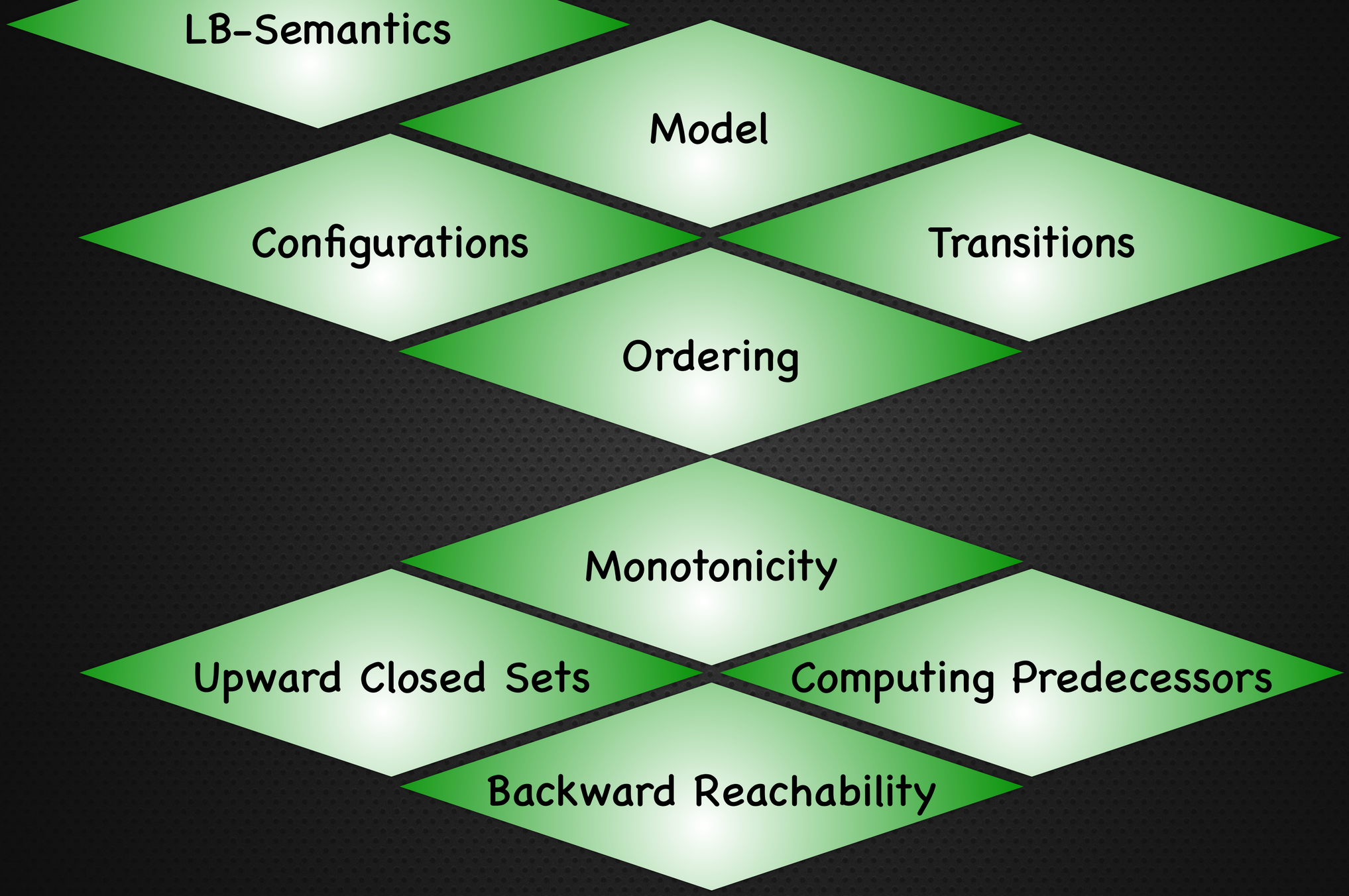
WALLPAPERSWIDE.COM

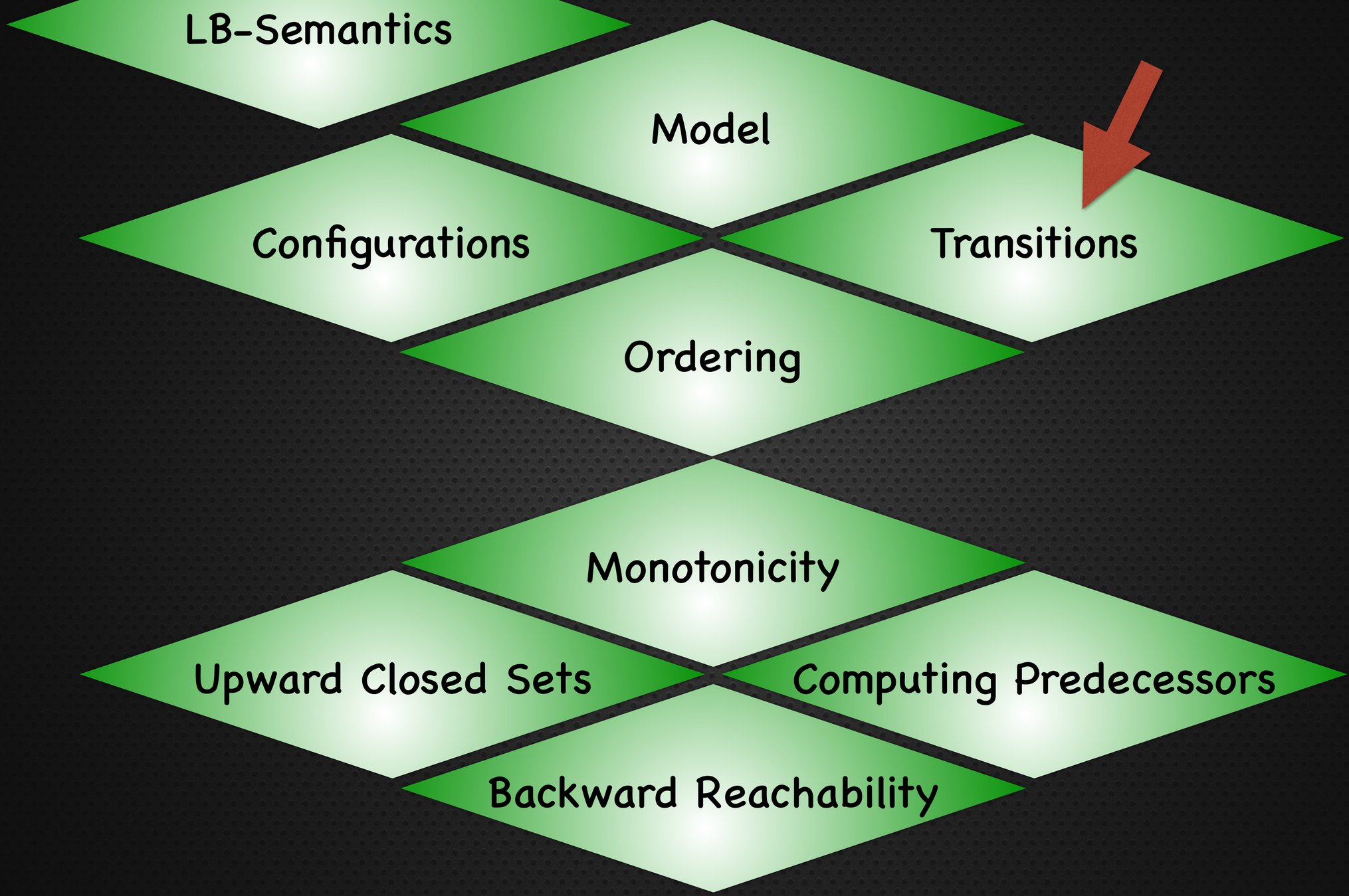


WALLPAPERSWIDE.COM



WALLPAPERSWIDE.COM


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability



WALLPAPERSWIDE.COM

WALLPAPERSWIDE.COM



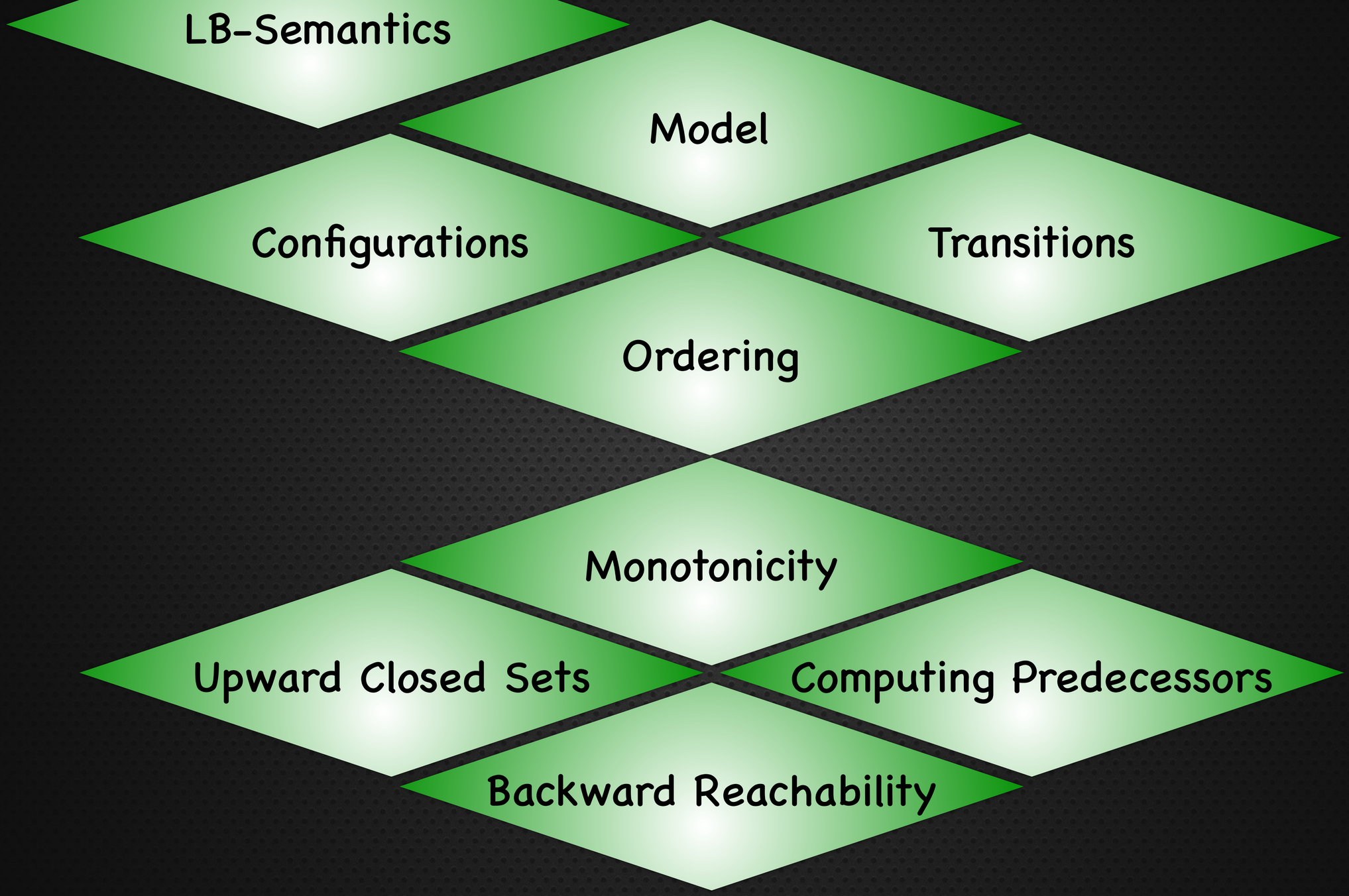
WALLPAPERSWIDE.COM

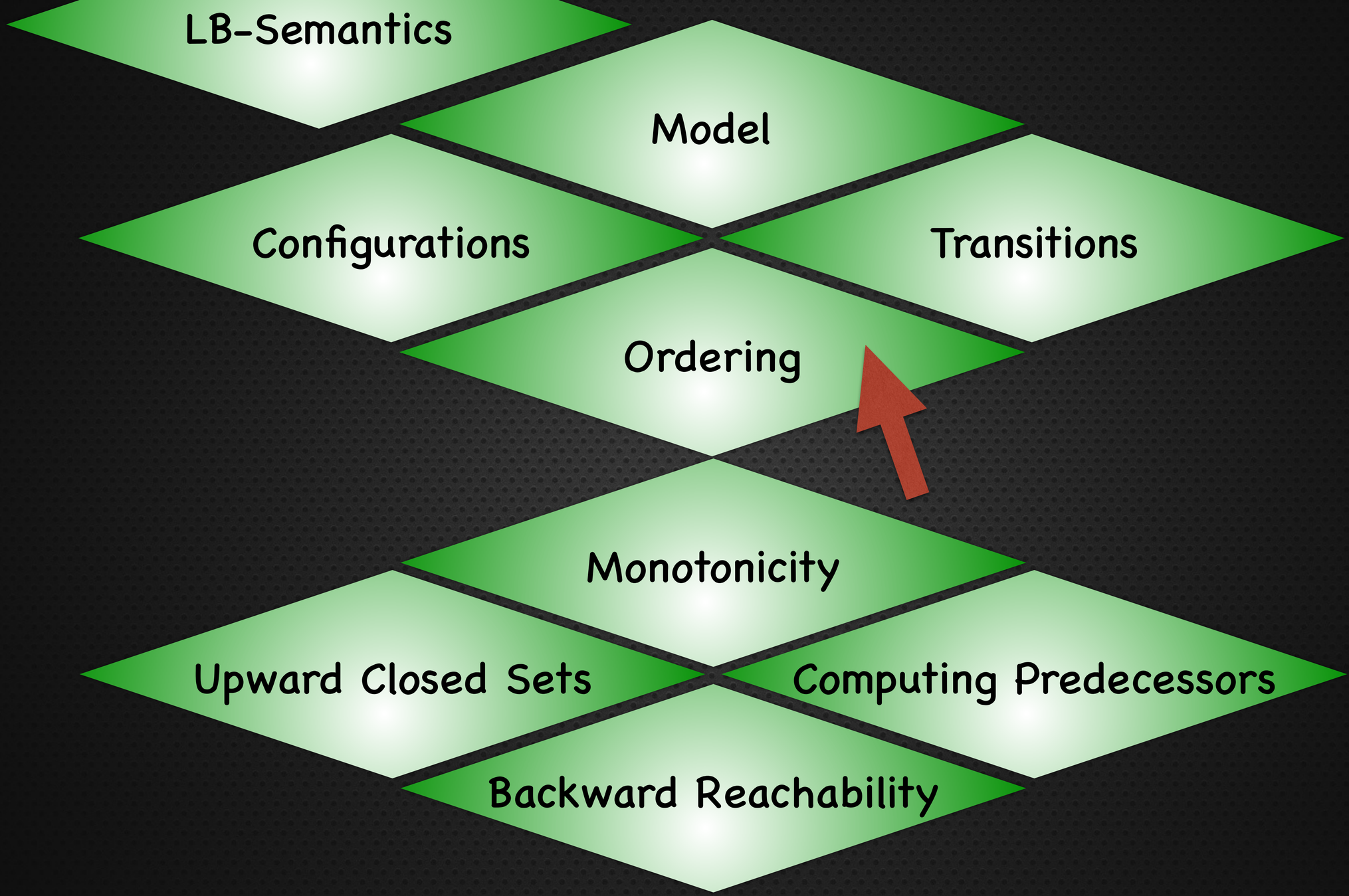
Transitions

Load-Buffer Semantics

- write-to-memory
- propagate-other-message
- read-own-write
- delete
- read-buffer-head

WALLPAPERSWIDE.COM


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability

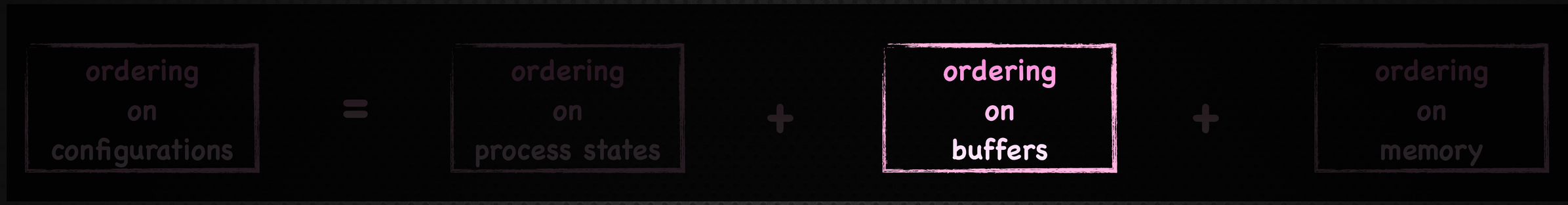
WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

WALLPAPERSWIDE.COM



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

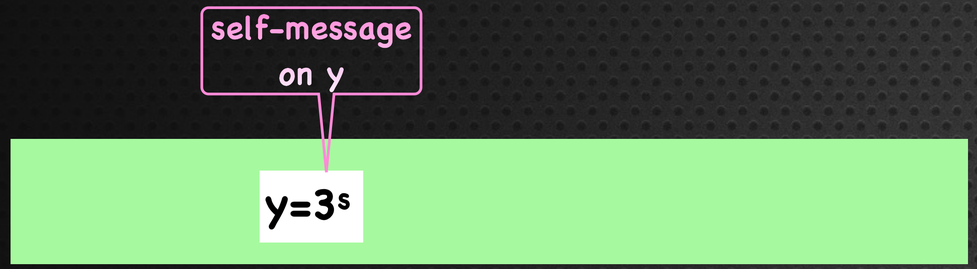
- self-messages:
  - same order
  - same value
  - "sub-word in-between"

$y=3^s$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

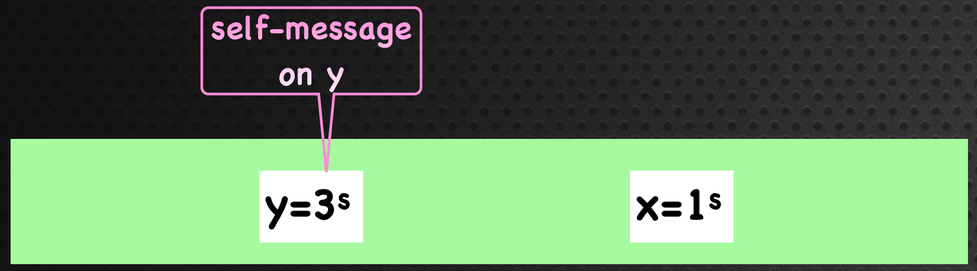
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

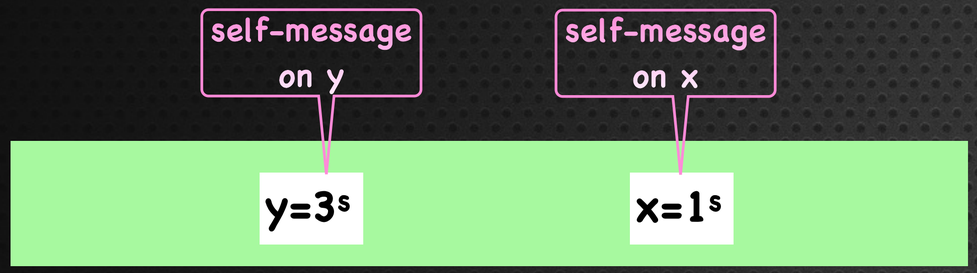
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

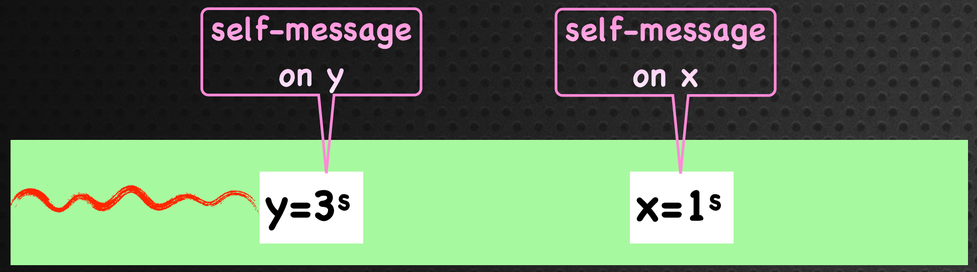
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

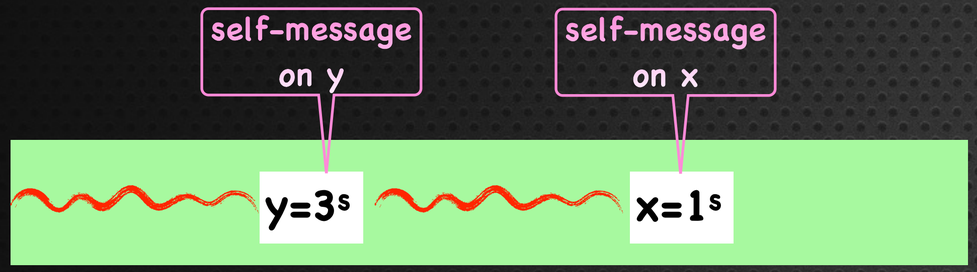
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

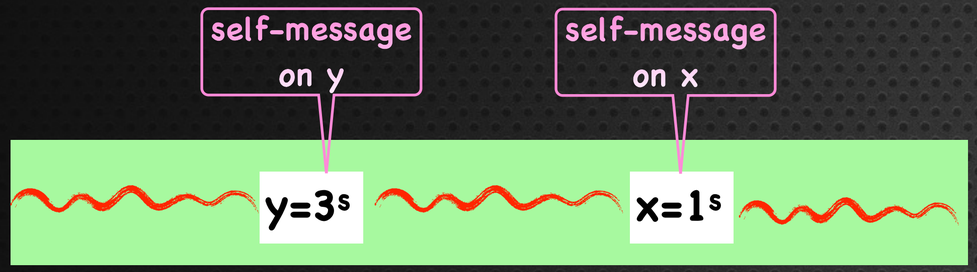
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

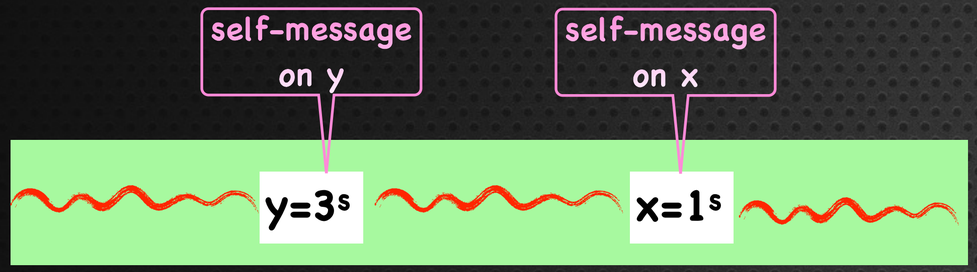
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

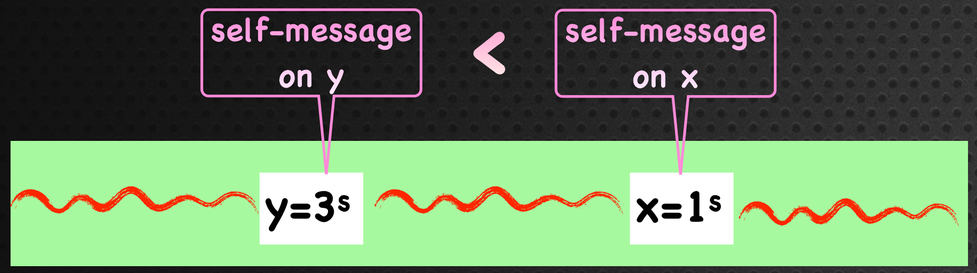
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

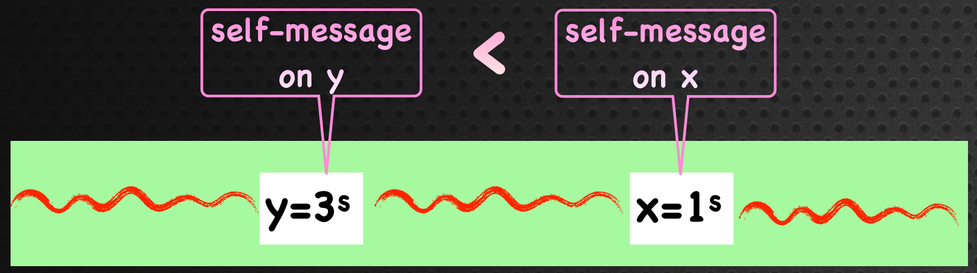
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

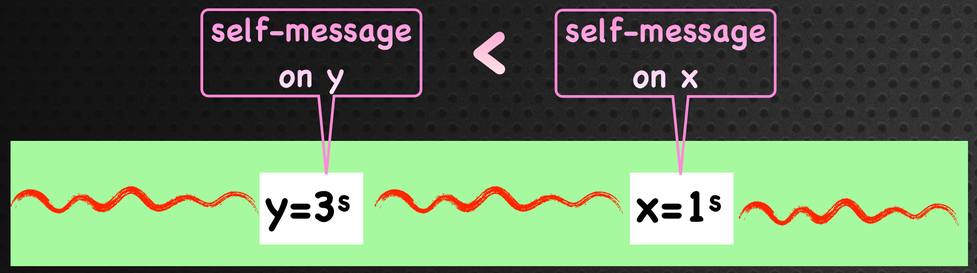
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

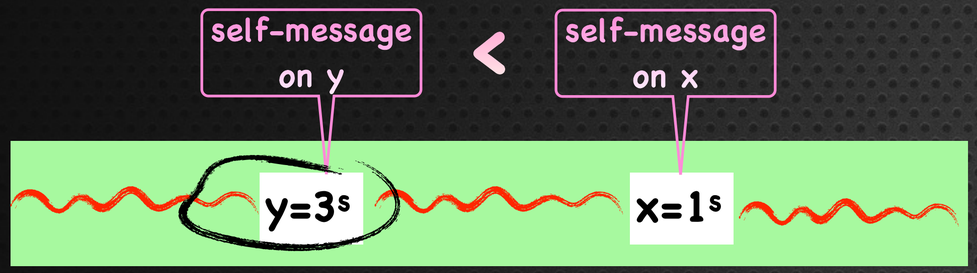
- self-messages:
  - ~~same order~~
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

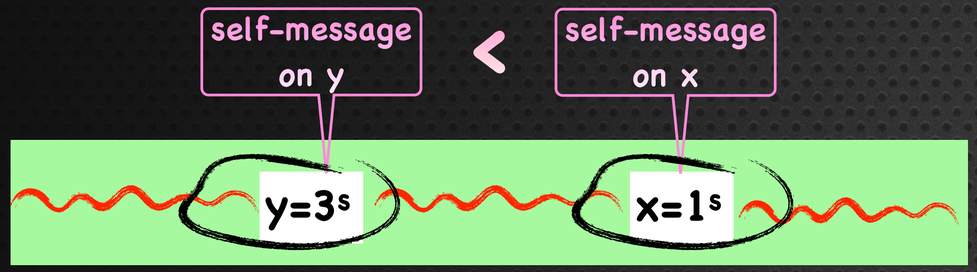
- self-messages:
  - ~~same order~~
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

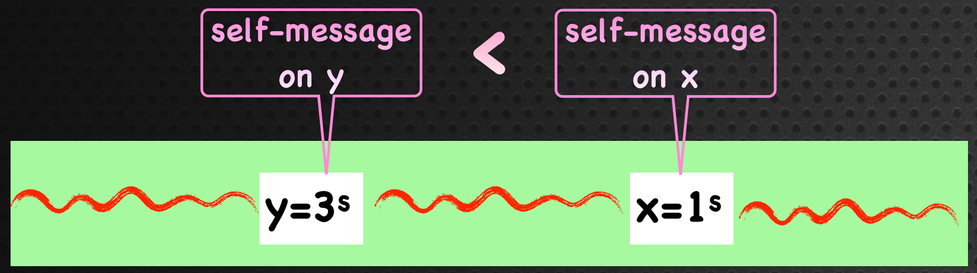
- self-messages:
  - ~~same order~~
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

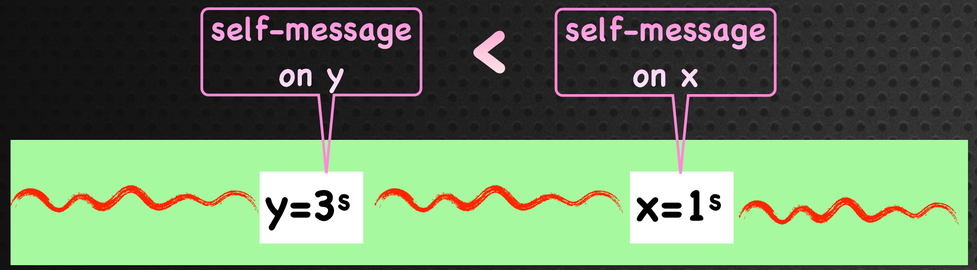
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

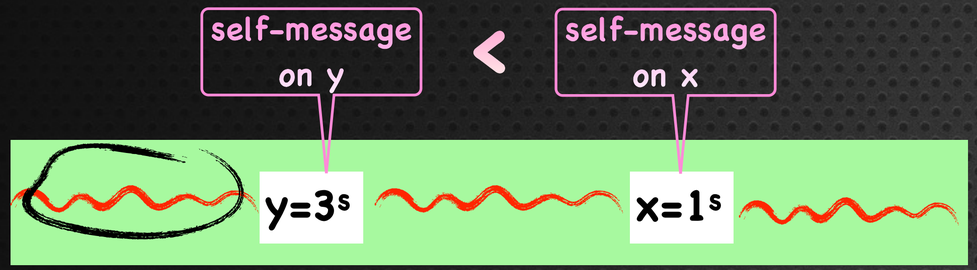
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

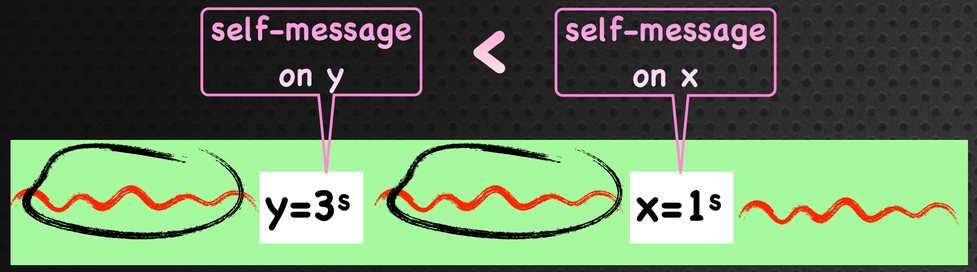
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

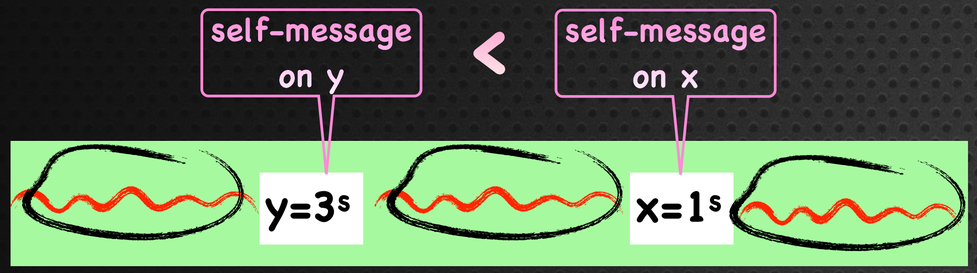
- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

x=5^{o}	y=3^{s}	y=2^{o}	x=4^{o}	x=1^{s}	y=2^{o}	x=8^{o}

y=4^{o}	x=5^{o}	x=7^{o}	y=3^{s}	y=2^{o}	x=9^{o}	x=4^{o}	x=1^{s}	x=2^{o}	y=2^{o}	x=6^{o}	x=8^{o}	y=8^{o}

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

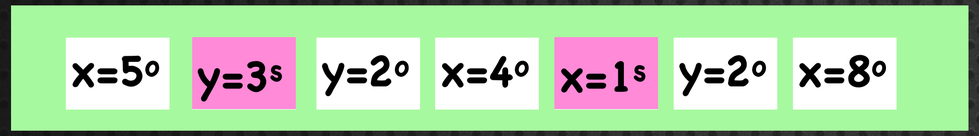
x=5^{o}	y=3^{s}	y=2^{o}	x=4^{o}	x=1^{s}	y=2^{o}	x=8^{o}

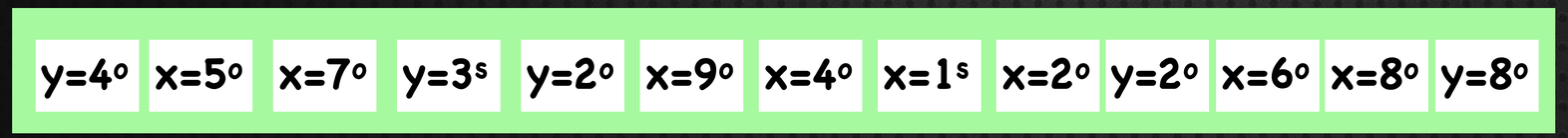
y=4^{o}	x=5^{o}	x=7^{o}	y=3^{s}	y=2^{o}	x=9^{o}	x=4^{o}	x=1^{s}	x=2^{o}	y=2^{o}	x=6^{o}	x=8^{o}	y=8^{o}

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

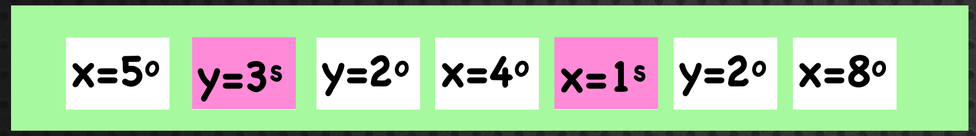


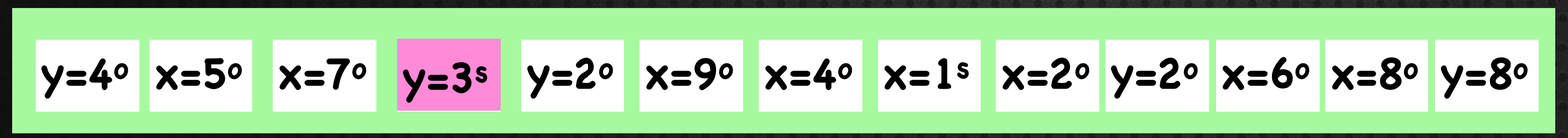


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

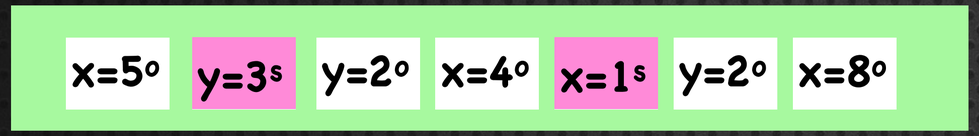


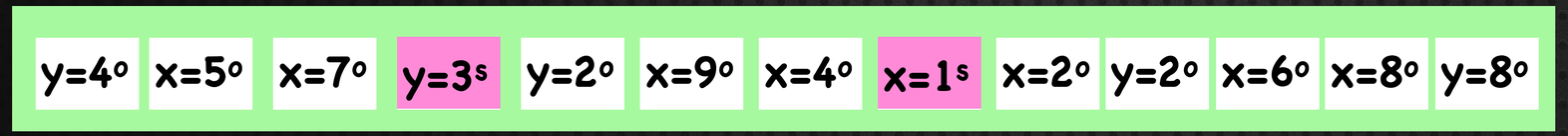


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"





WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

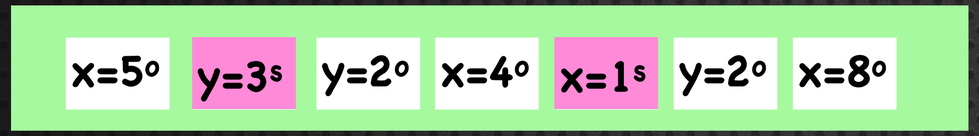
$x=5^{o}$ $y=3^{s}$ $y=2^{o}$ $x=4^{o}$ $x=1^{s}$ $y=2^{o}$ $x=8^{o}$

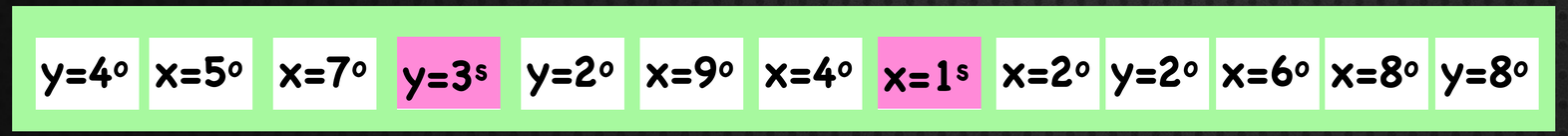
$y=4^{o}$ $x=5^{o}$ $x=7^{o}$ $y=3^{s}$ $y=2^{o}$ $x=9^{o}$ $x=4^{o}$ $x=1^{s}$ $x=2^{o}$ $y=2^{o}$ $x=6^{o}$ $x=8^{o}$ $y=8^{o}$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"





WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

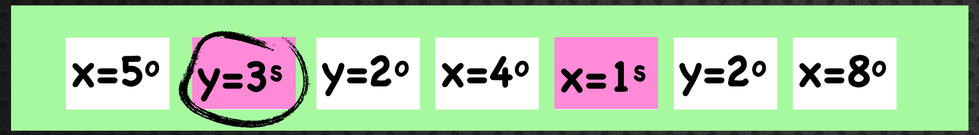
$x=5^{o}$ $y=3^{s}$ $y=2^{o}$ $x=4^{o}$ $x=1^{s}$ $y=2^{o}$ $x=8^{o}$

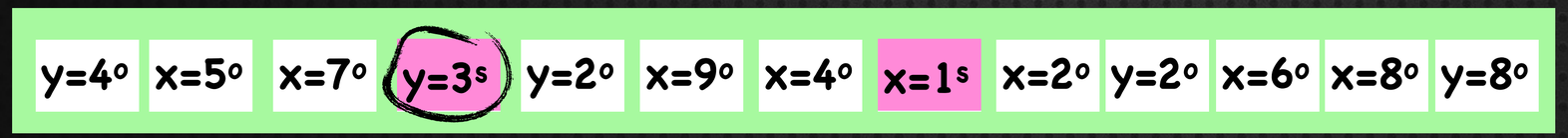
$y=4^{o}$ $x=5^{o}$ $x=7^{o}$ $y=3^{s}$ $y=2^{o}$ $x=9^{o}$ $x=4^{o}$ $x=1^{s}$ $x=2^{o}$ $y=2^{o}$ $x=6^{o}$ $x=8^{o}$ $y=8^{o}$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"





WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

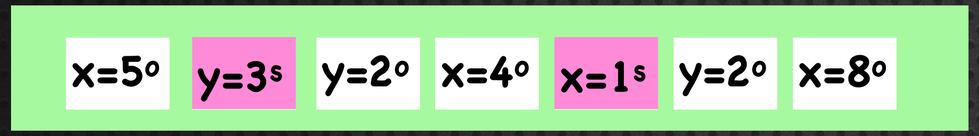
x=5^o y=3^s y=2^o x=4^o x=1^s y=2^o x=8^o

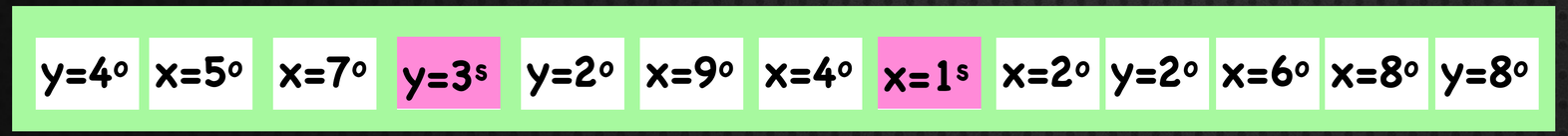
y=4^o x=5^o x=7^o y=3^s y=2^o x=9^o x=4^o x=1^s x=2^o y=2^o x=6^o x=8^o y=8^o

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"





WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

$x=5^{o}$ $y=3^{s}$ $y=2^{o}$ $x=4^{o}$ $x=1^{s}$ $y=2^{o}$ $x=8^{o}$

$y=4^{o}$ $x=5^{o}$ $x=7^{o}$ $y=3^{s}$ $y=2^{o}$ $x=9^{o}$ $x=4^{o}$ $x=1^{s}$ $x=2^{o}$ $y=2^{o}$ $x=6^{o}$ $x=8^{o}$ $y=8^{o}$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

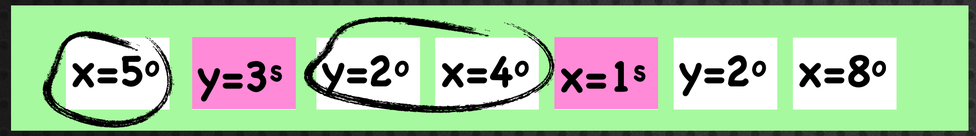
$x=5^{o}$ $y=3^{s}$ $y=2^{o}$ $x=4^{o}$ $x=1^{s}$ $y=2^{o}$ $x=8^{o}$

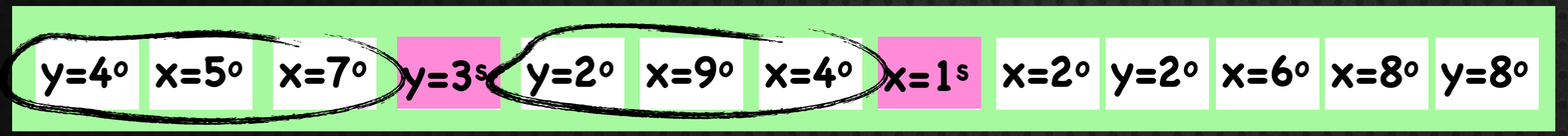
$y=4^{o}$ $x=5^{o}$ $x=7^{o}$ $y=3^{s}$ $y=2^{o}$ $x=9^{o}$ $x=4^{o}$ $x=1^{s}$ $x=2^{o}$ $y=2^{o}$ $x=6^{o}$ $x=8^{o}$ $y=8^{o}$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

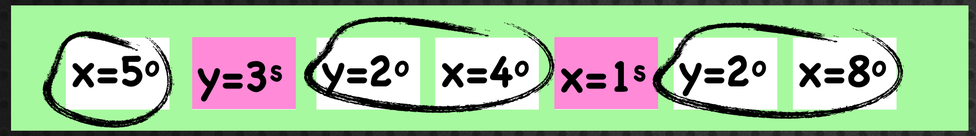


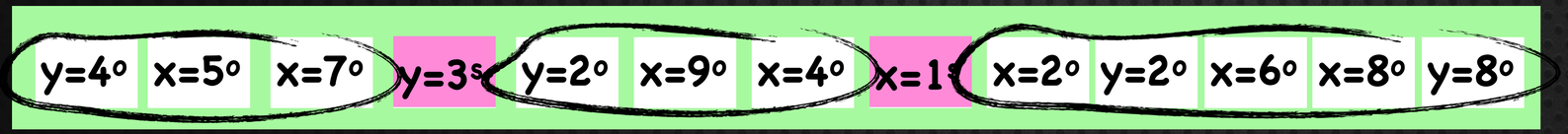


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

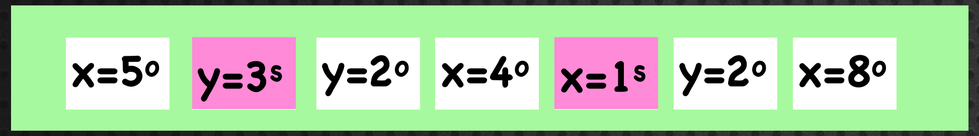


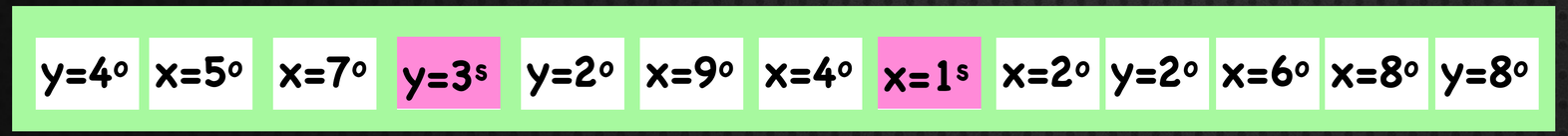


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"





WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

$x=5^{o}$ $y=3^{s}$ $y=2^{o}$ $x=4^{o}$ $x=1^{s}$ $y=2^{o}$ $x=8^{o}$

$\sqsubset$

$y=4^{o}$ $x=5^{o}$ $x=7^{o}$ $y=3^{s}$ $y=2^{o}$ $x=9^{o}$ $x=4^{o}$ $x=1^{s}$ $x=2^{o}$ $y=2^{o}$ $x=6^{o}$ $x=8^{o}$ $y=8^{o}$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

$x=5^o$ $y=3^s$ $y=2^o$ $x=4^o$ $x=1^s$ $y=2^o$ $x=8^o$

WQO

$\sqsubset$

$y=4^o$ $x=5^o$ $x=7^o$ $y=3^s$ $y=2^o$ $x=9^o$ $x=4^o$ $x=1^s$ $x=2^o$ $y=2^o$ $x=6^o$ $x=8^o$ $y=8^o$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

equality on finite sets

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

$x=5^o$ $y=3^s$ $y=2^o$ $x=4^o$ $x=1^s$ $y=2^o$ $x=8^o$

WQO

$\sqsubset$

$y=4^o$ $x=5^o$ $x=7^o$ $y=3^s$ $y=2^o$ $x=9^o$ $x=4^o$ $x=1^s$ $x=2^o$ $y=2^o$ $x=6^o$ $x=8^o$ $y=8^o$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

equality on finite sets

equality on finite sets

$x=5^o$ $y=3^s$ $y=2^o$ $x=4^o$ $x=1^s$ $y=2^o$ $x=8^o$

WQO

$\sqsubseteq$

$y=4^o$ $x=5^o$ $x=7^o$ $y=3^s$ $y=2^o$ $x=9^o$ $x=4^o$ $x=1^s$ $x=2^o$ $y=2^o$ $x=6^o$ $x=8^o$ $y=8^o$

WALLPAPERSWIDE.COM

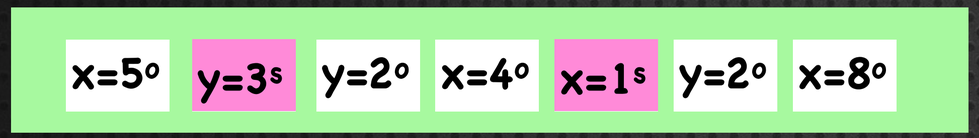
ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

equality on finite sets

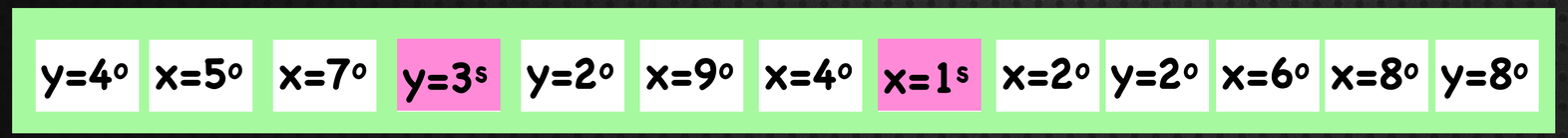
finite data domain

equality on finite sets



WQO

$\sqsubseteq$



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

equality on finite sets

finite data domain

equality on finite sets

Higman's theorem

WQO

$x=5^{o}$ $y=3^{s}$ $y=2^{o}$ $x=4^{o}$ $x=1^{s}$ $y=2^{o}$ $x=8^{o}$

$\sqsubset$

$y=4^{o}$ $x=5^{o}$ $x=7^{o}$ $y=3^{s}$ $y=2^{o}$ $x=9^{o}$ $x=4^{o}$ $x=1^{s}$ $x=2^{o}$ $y=2^{o}$ $x=6^{o}$ $x=8^{o}$ $y=8^{o}$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

equality on finite sets

finite data domain

equality on finite sets

Higman's theorem

WQO

$x=5^{o}$ $y=3^{s}$ $y=2^{o}$ $x=4^{o}$ $x=1^{s}$ $y=2^{o}$ $x=8^{o}$

$\sqsubset$

$y=4^{o}$ $x=5^{o}$ $x=7^{o}$ $y=3^{s}$ $y=2^{o}$ $x=9^{o}$ $x=4^{o}$ $x=1^{s}$ $x=2^{o}$ $y=2^{o}$ $x=6^{o}$ $x=8^{o}$ $y=8^{o}$

vectors of words

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

equality on finite sets

finite data domain

equality on finite sets

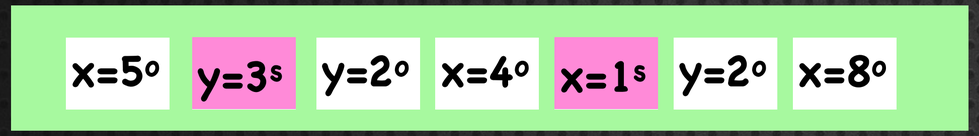
vectors of WQOs

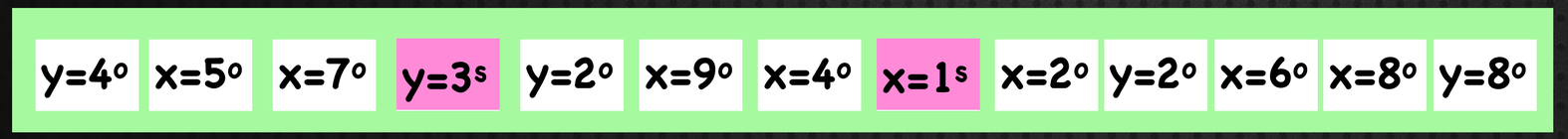
Higman's theorem

WQO

$\sqsubseteq$

vectors of words





WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

equality on finite sets

finite data domain

equality on finite sets

vectors of WQOs

Higman's theorem

WQO

$\sqsubseteq$

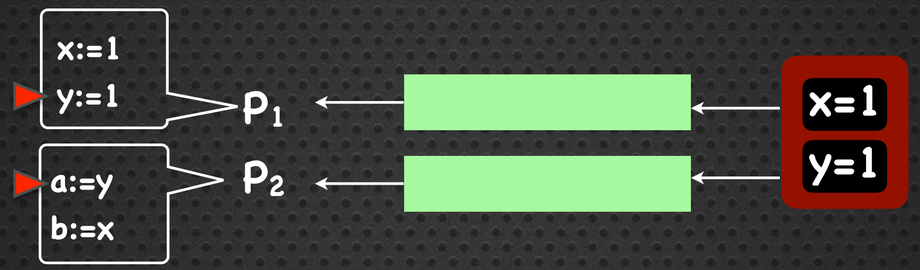
vectors of words

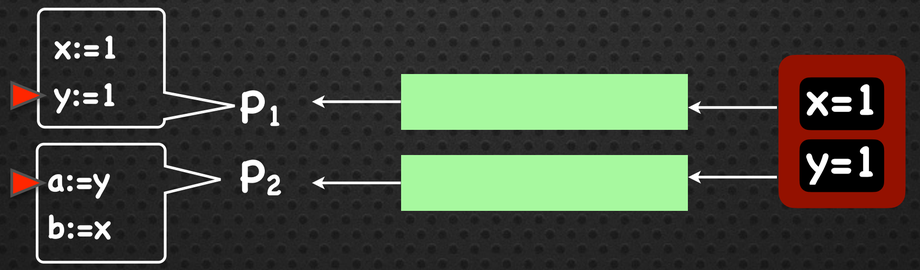
$x=5^o$	$y=3^s$	$y=2^o$	$x=4^o$	$x=1^s$	$y=2^o$	$x=8^o$

$y=4^o$	$x=5^o$	$x=7^o$	$y=3^s$	$y=2^o$	$x=9^o$	$x=4^o$	$x=1^s$	$x=2^o$	$y=2^o$	$x=6^o$	$x=8^o$	$y=8^o$

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

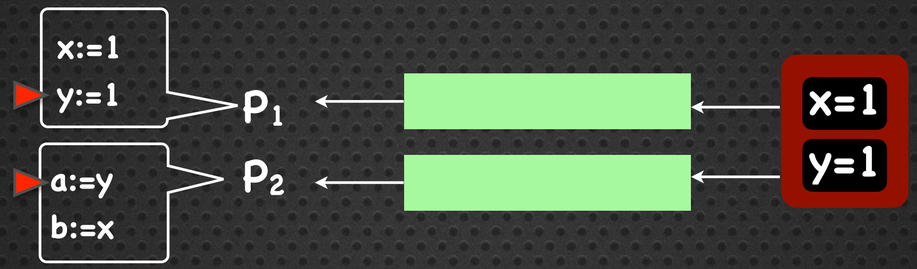


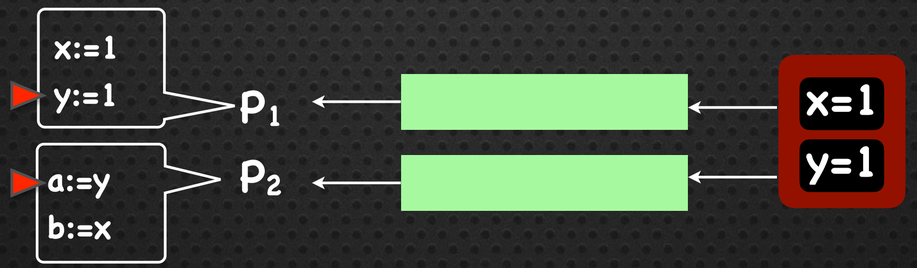


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters



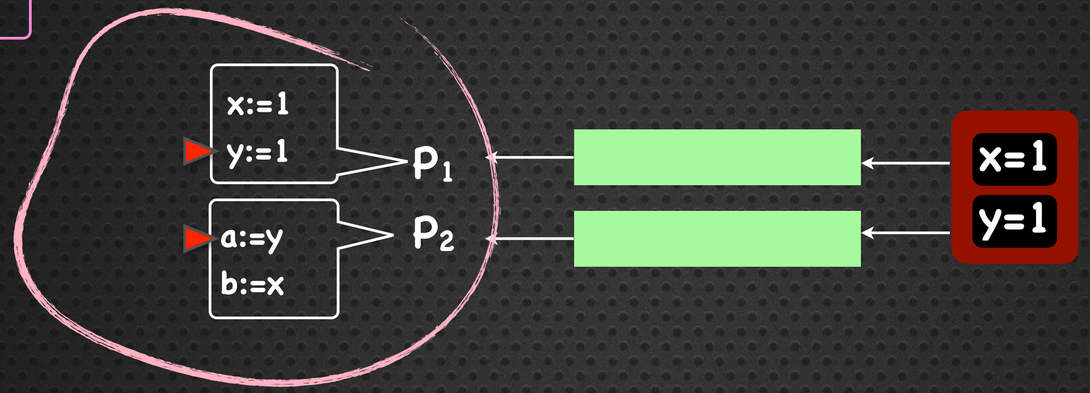


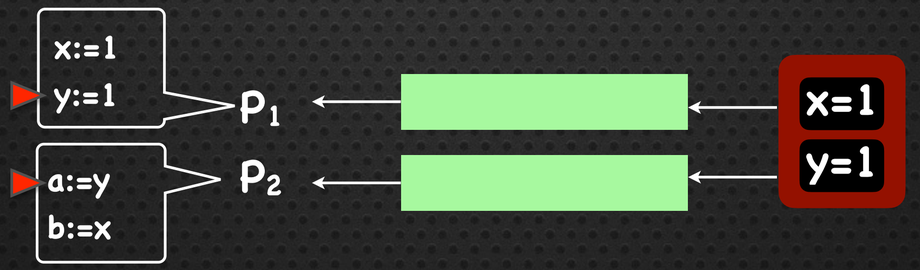
WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

=

identical program counters



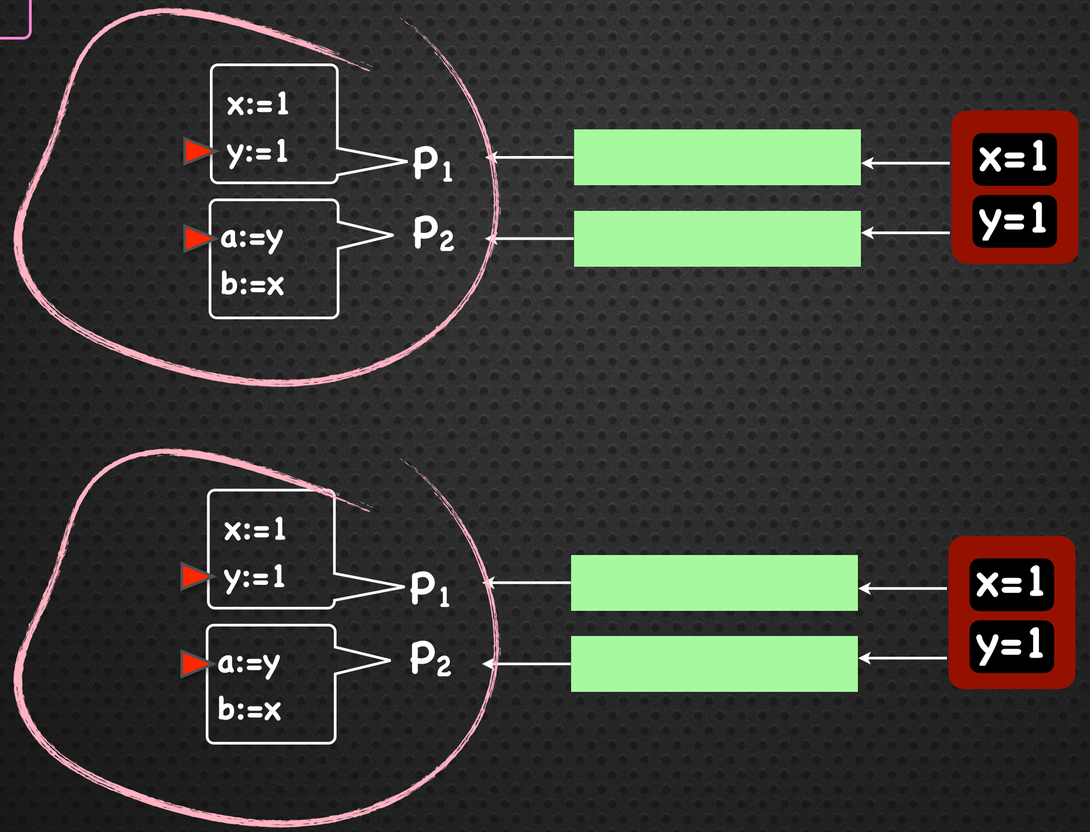


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

=

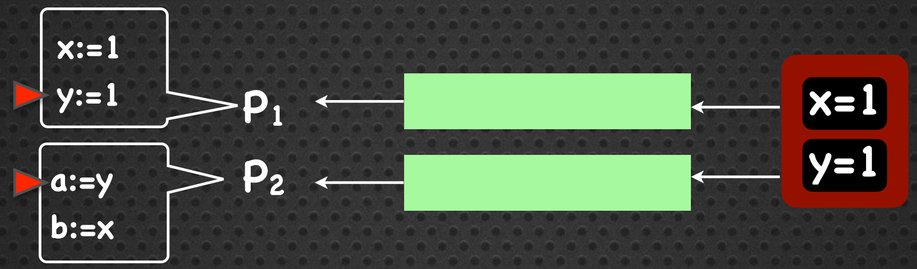
identical program counters

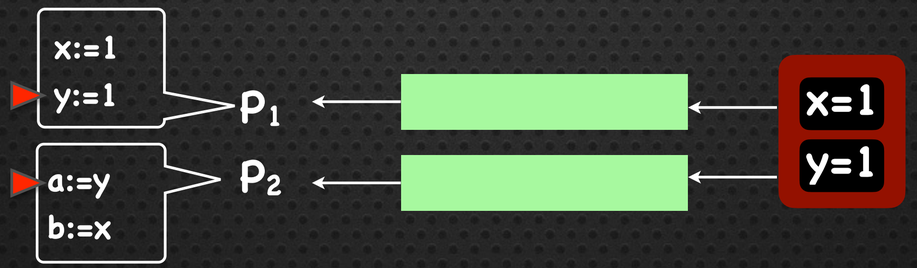


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

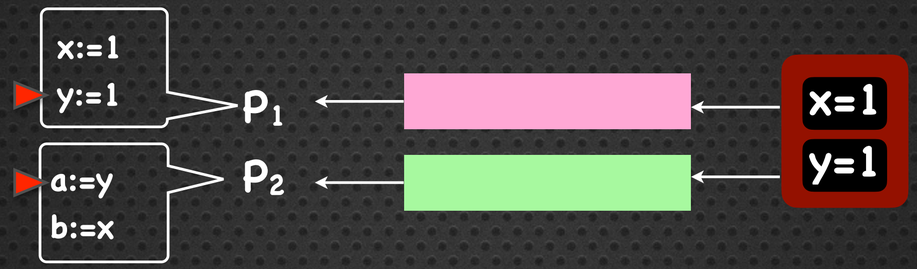


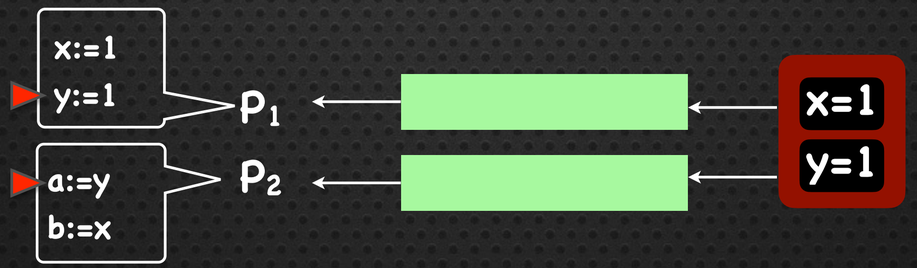


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

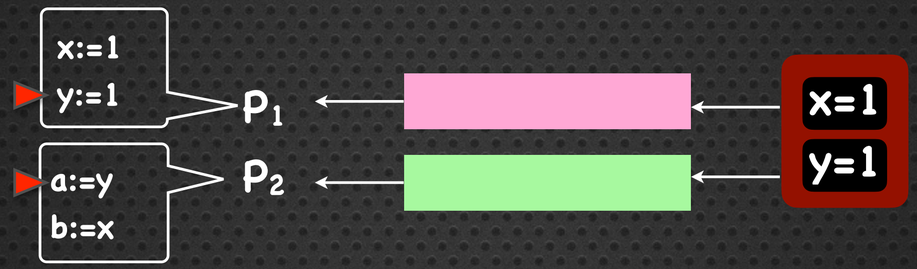


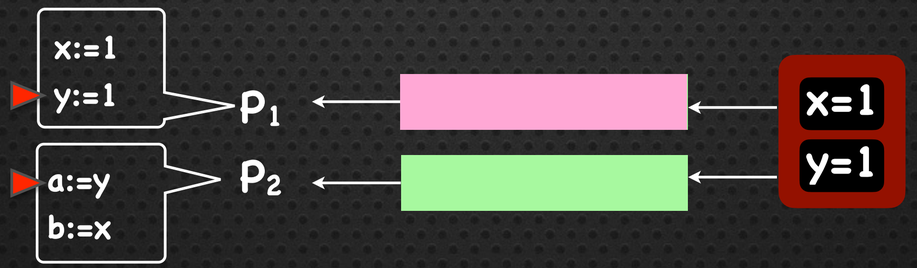


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

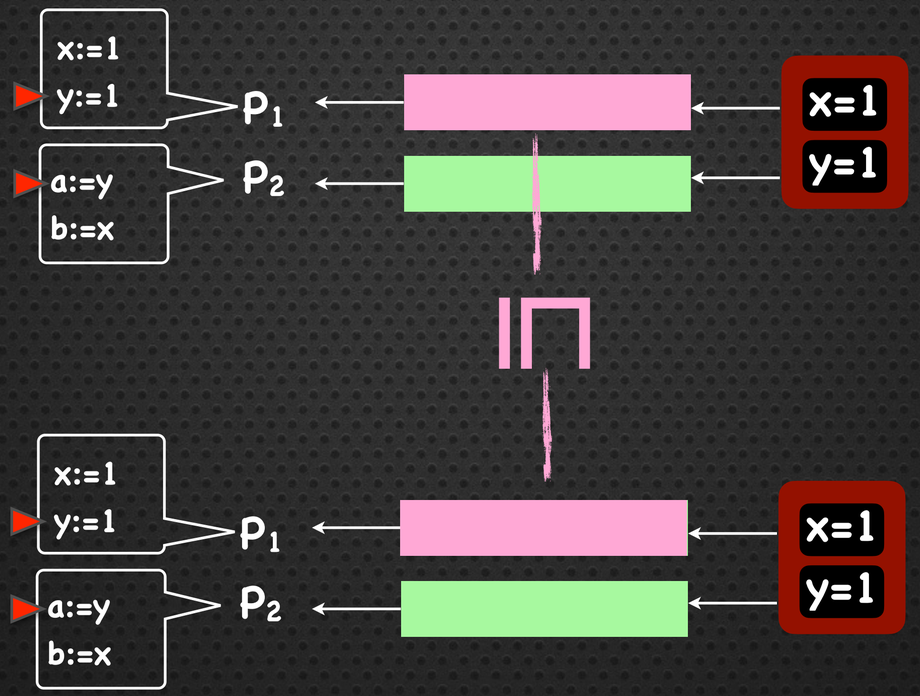




WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

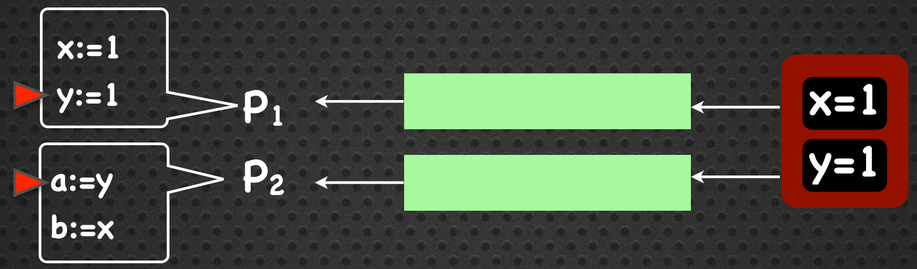
= identical program counters

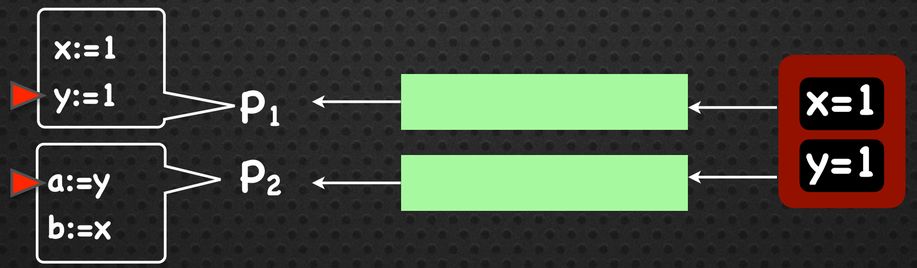


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

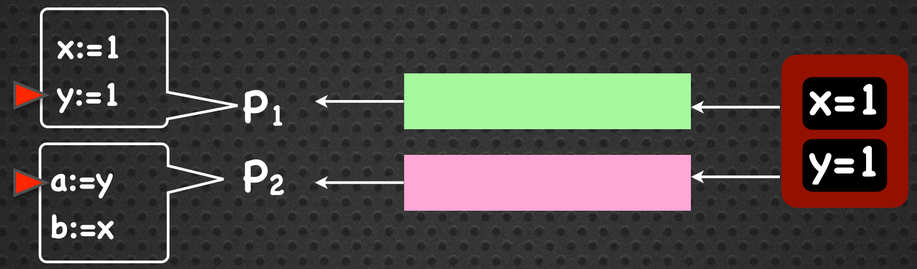


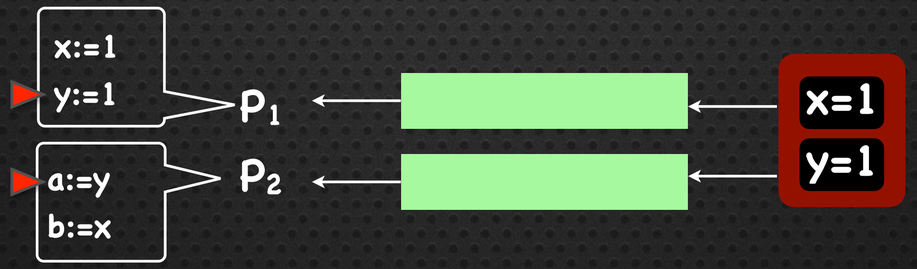


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

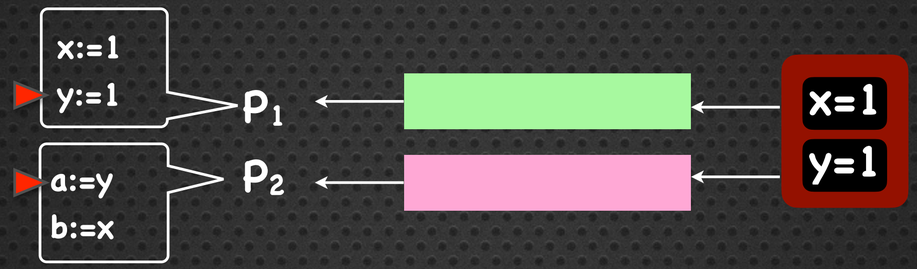


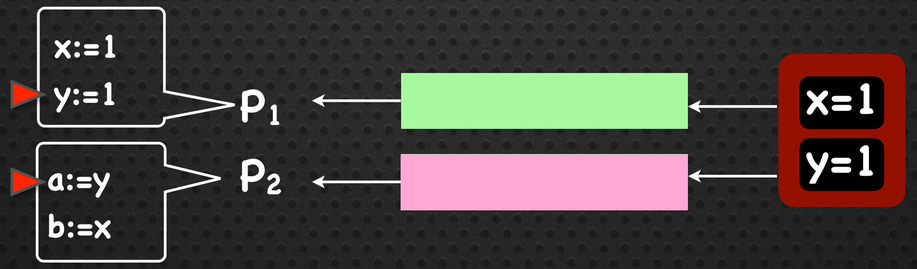


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

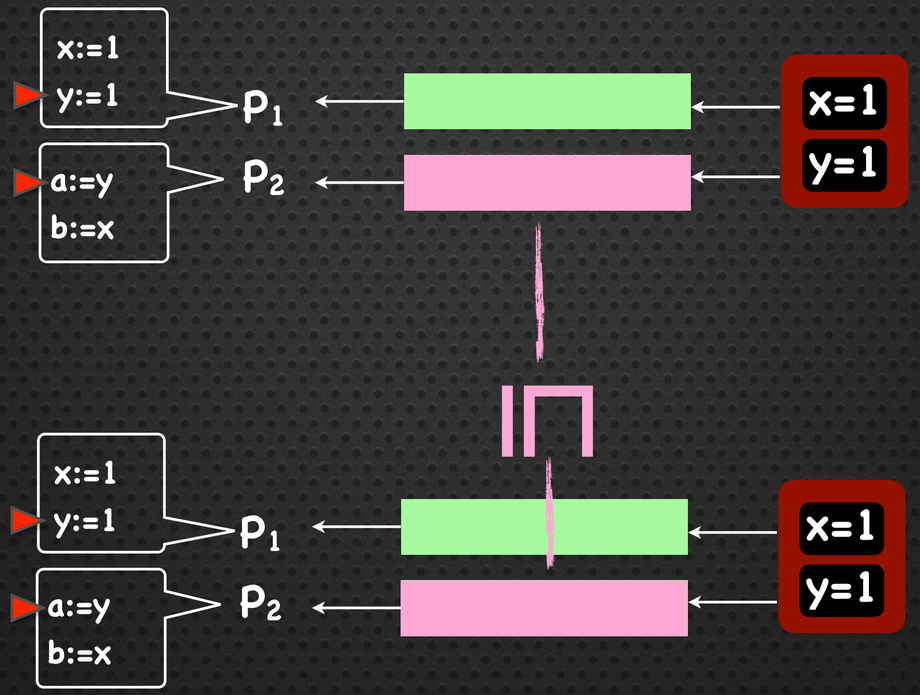




WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

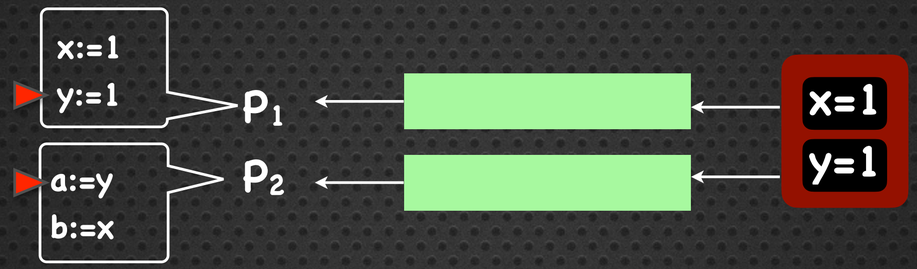
= identical program counters

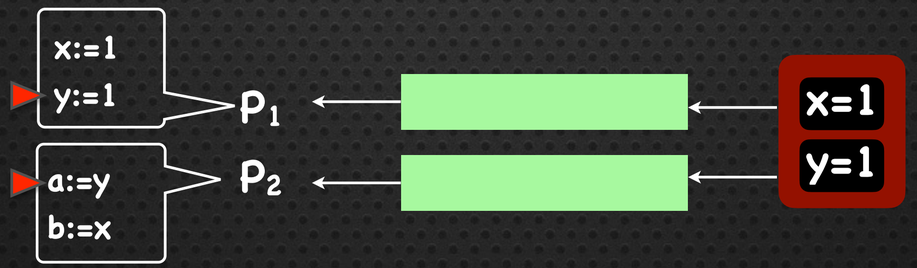


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

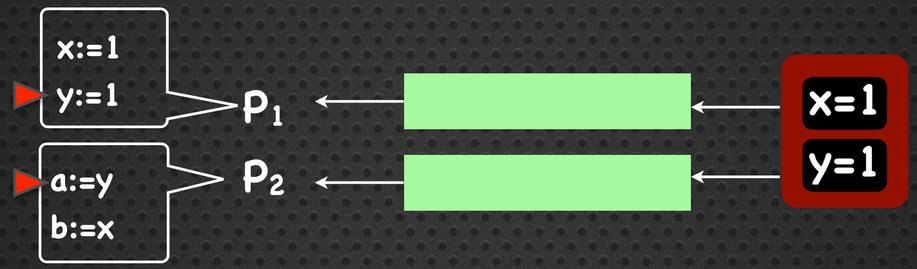


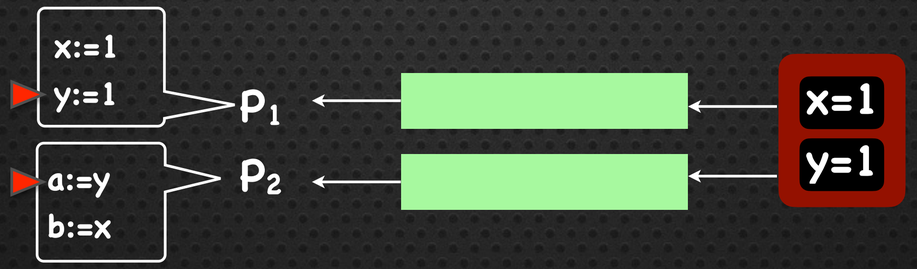


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory =

identical program counters =



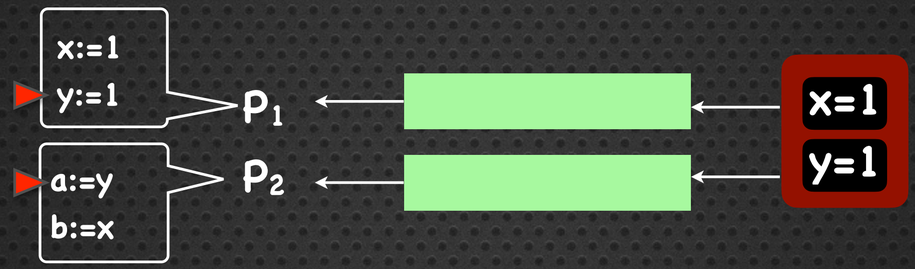


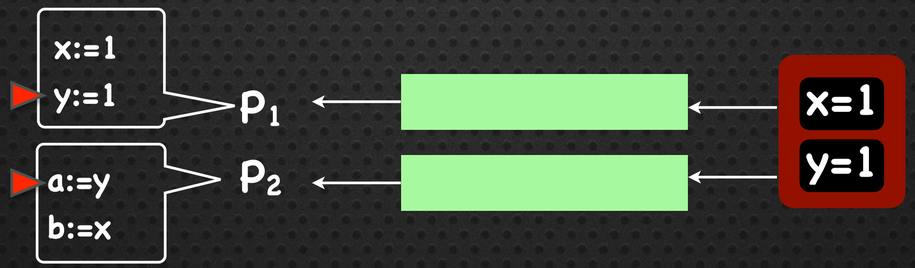
WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

= identical memory states



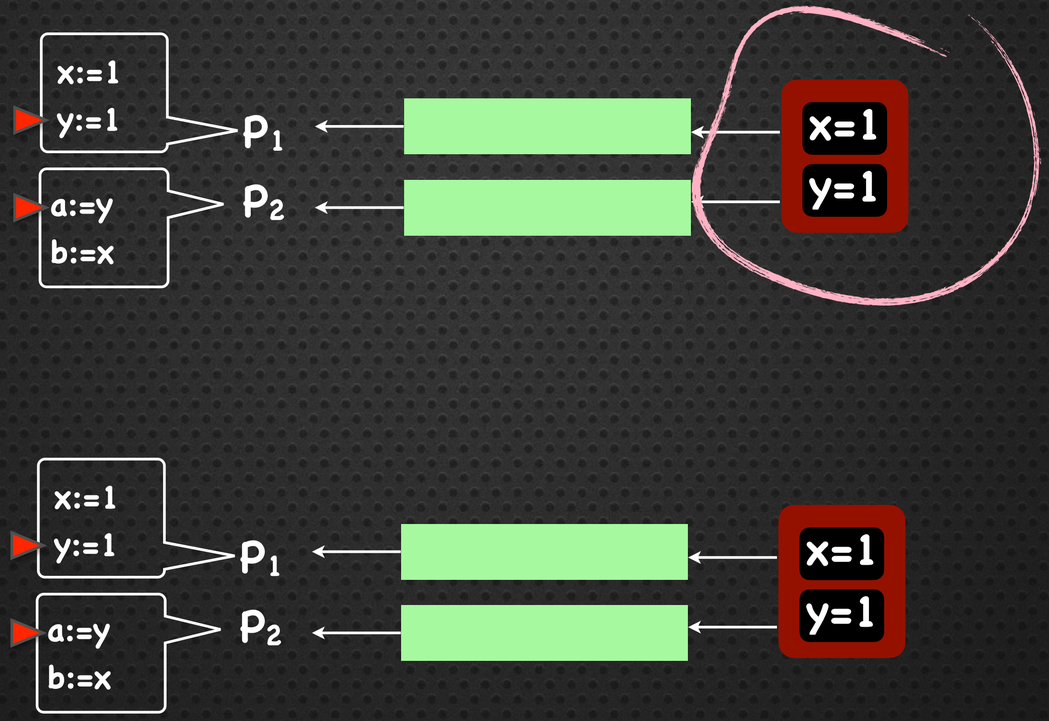


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

= identical memory states

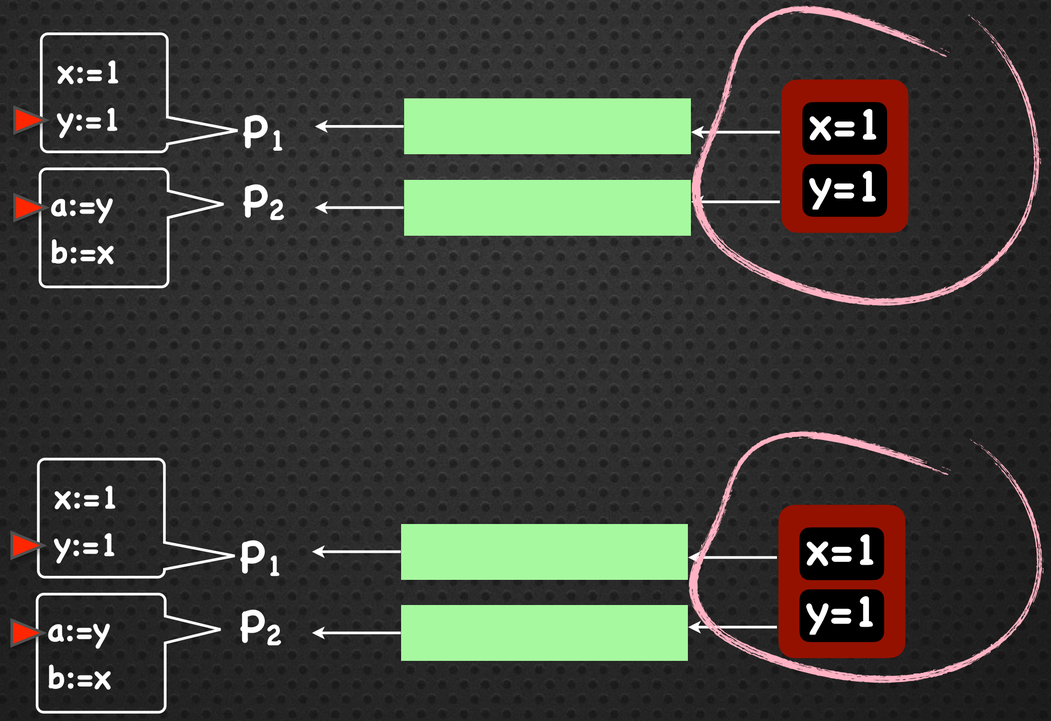


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

identical program counters =

identical memory states =

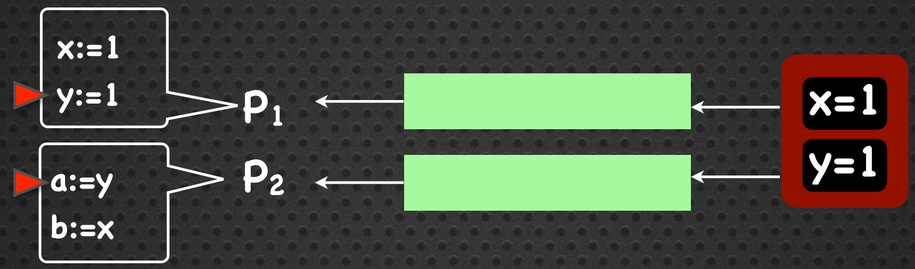


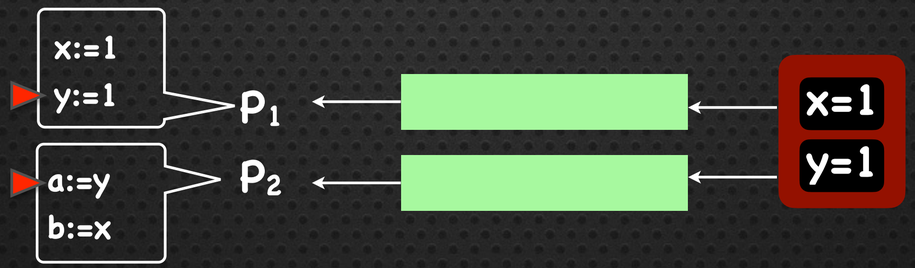
WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

= identical memory states



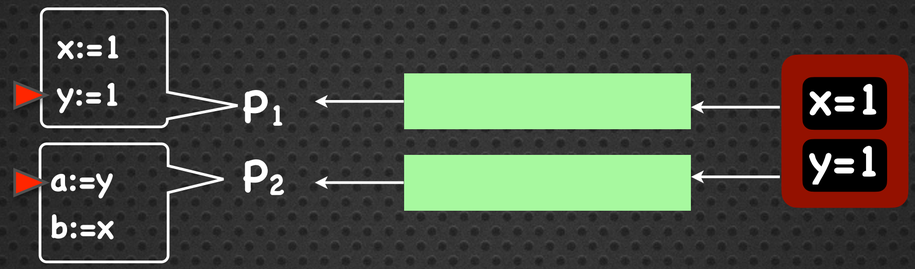


WALLPAPERSWIDE.COM

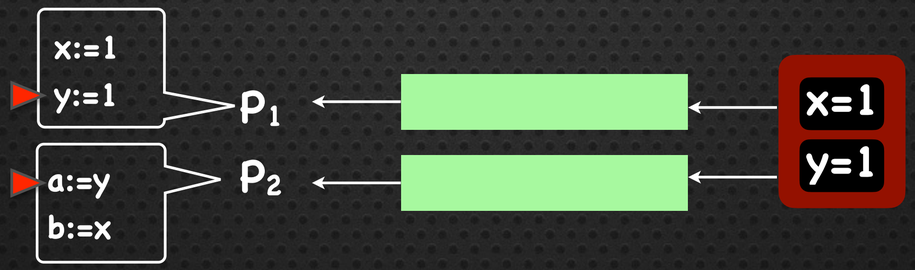
ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

= identical memory states



$\sqsubseteq$

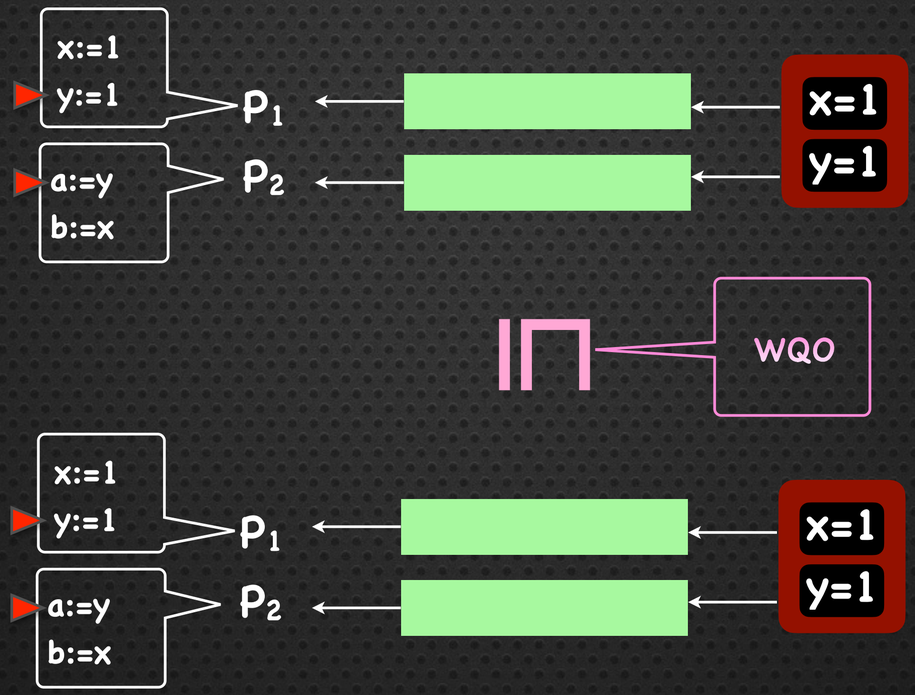


WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

= identical memory states



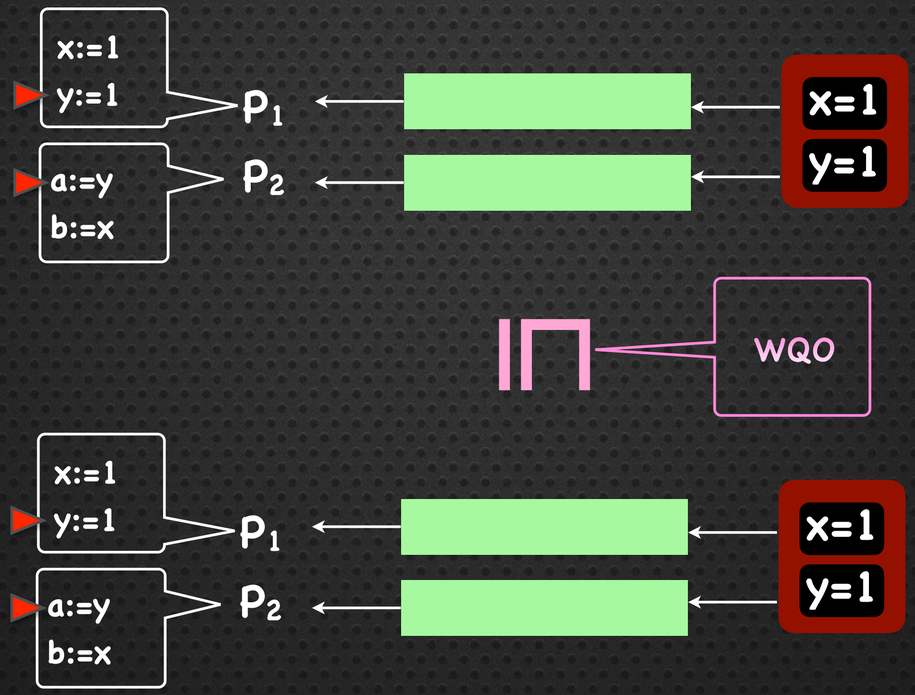
WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

= WQO

= identical memory states



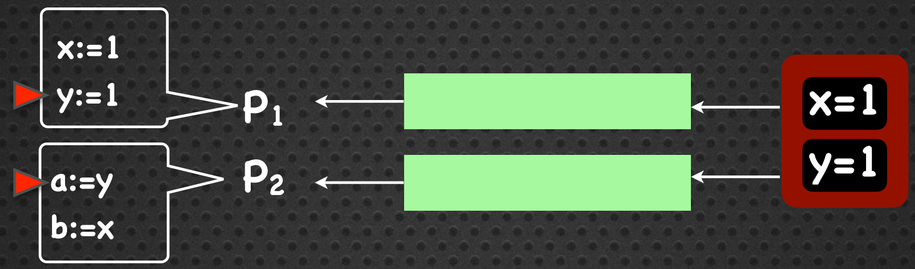
ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

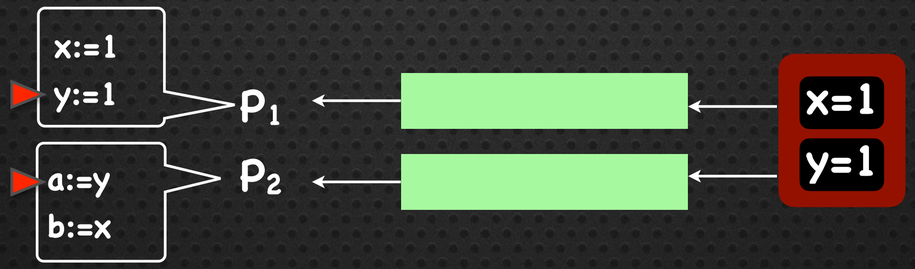
= WQO

= identical memory states

= equality on finite sets



⊓ WQO



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

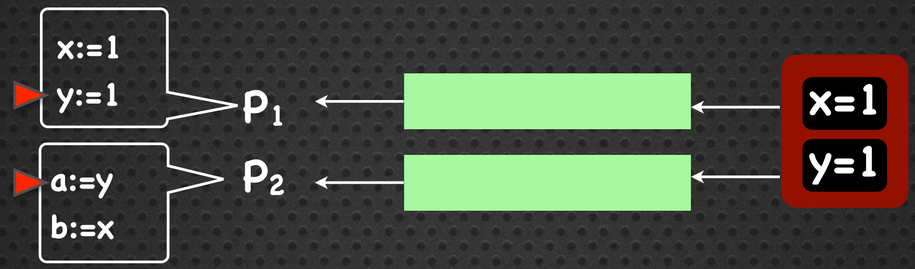
= identical program counters

WQO

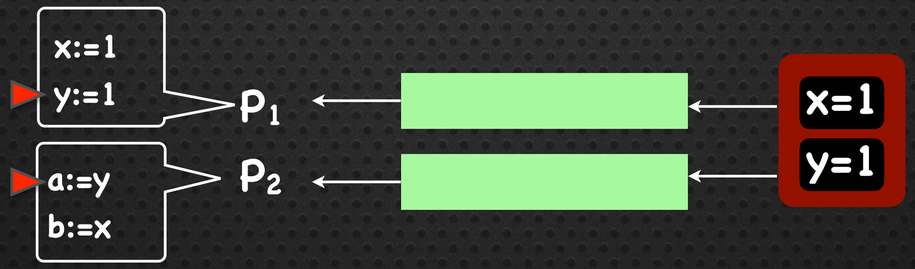
= identical memory states

finite data domain

equality on finite sets



⊑ WQO



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

=

identical program counters

equality on finite sets

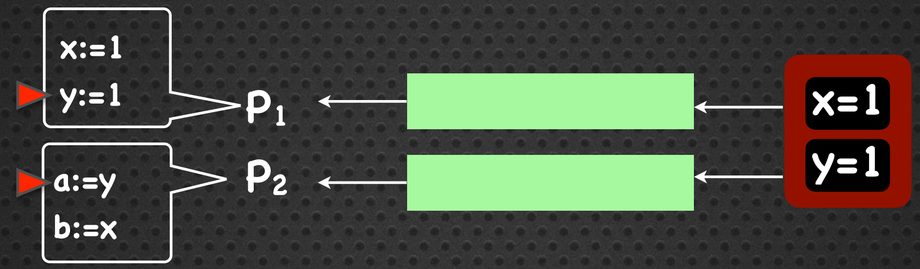
WQO

=

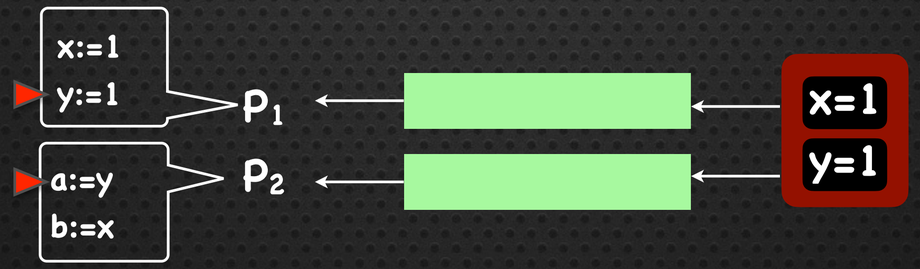
identical memory states

finite data domain

equality on finite sets



⊓ WQO



WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

= identical program counters

finite-state processes

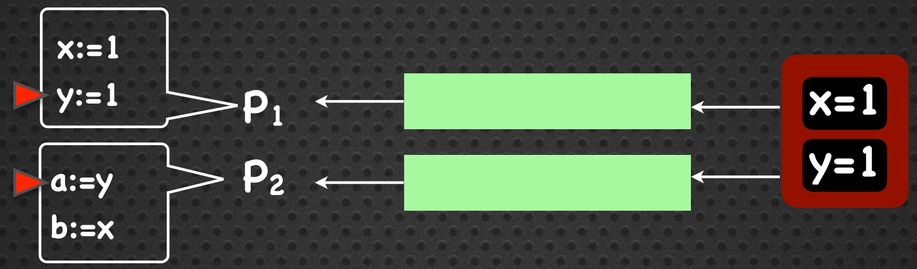
equality on finite sets

WQO

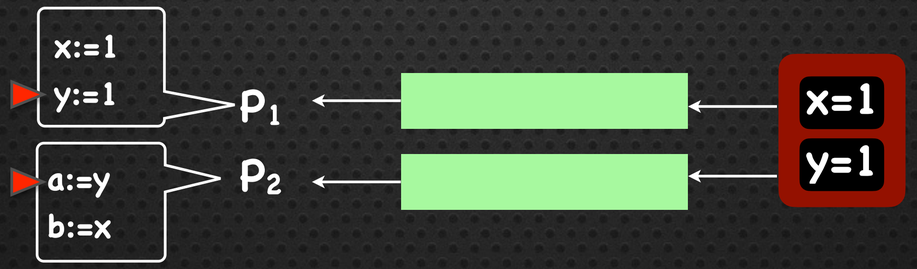
= identical memory states

finite data domain

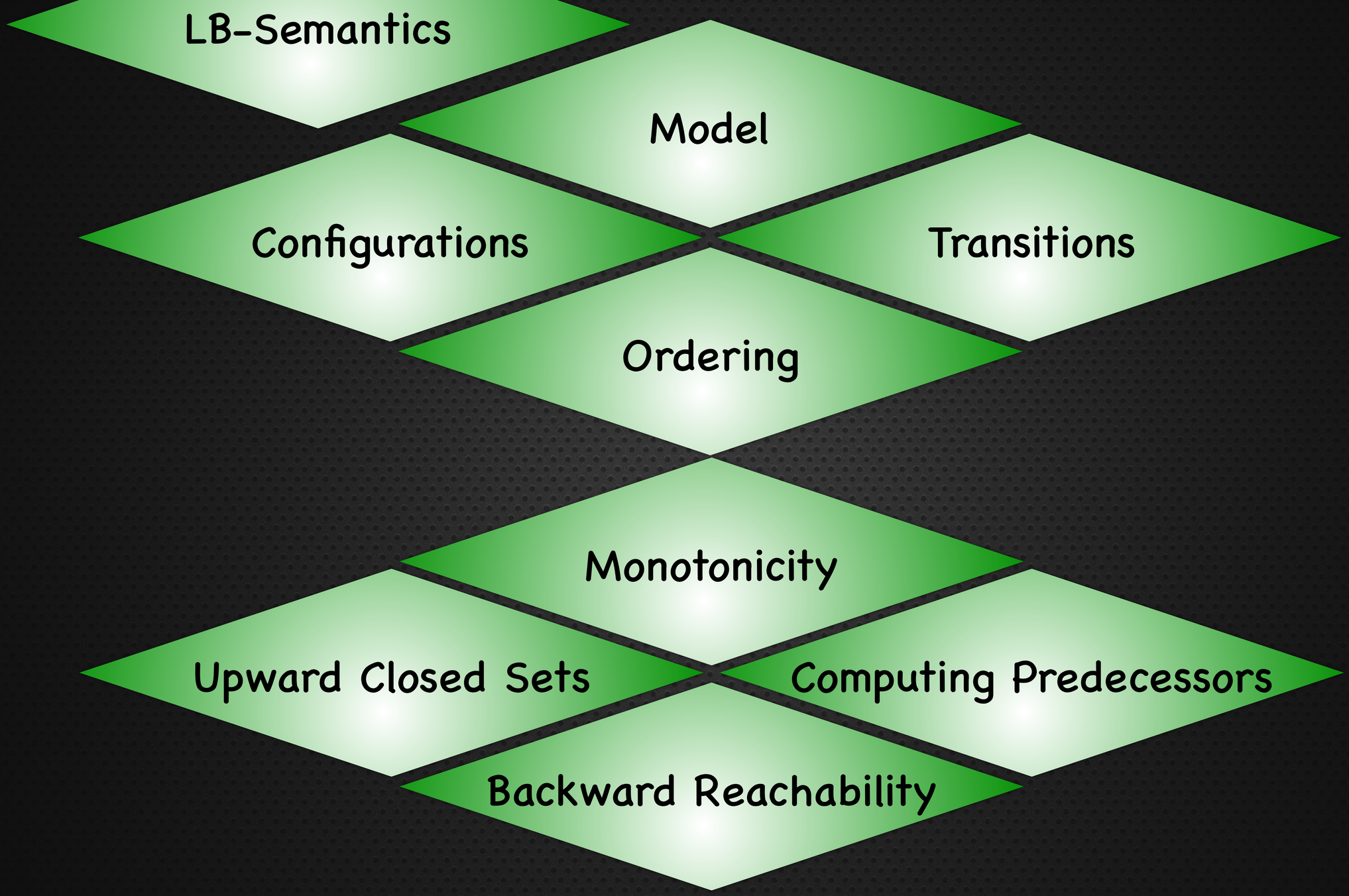
equality on finite sets

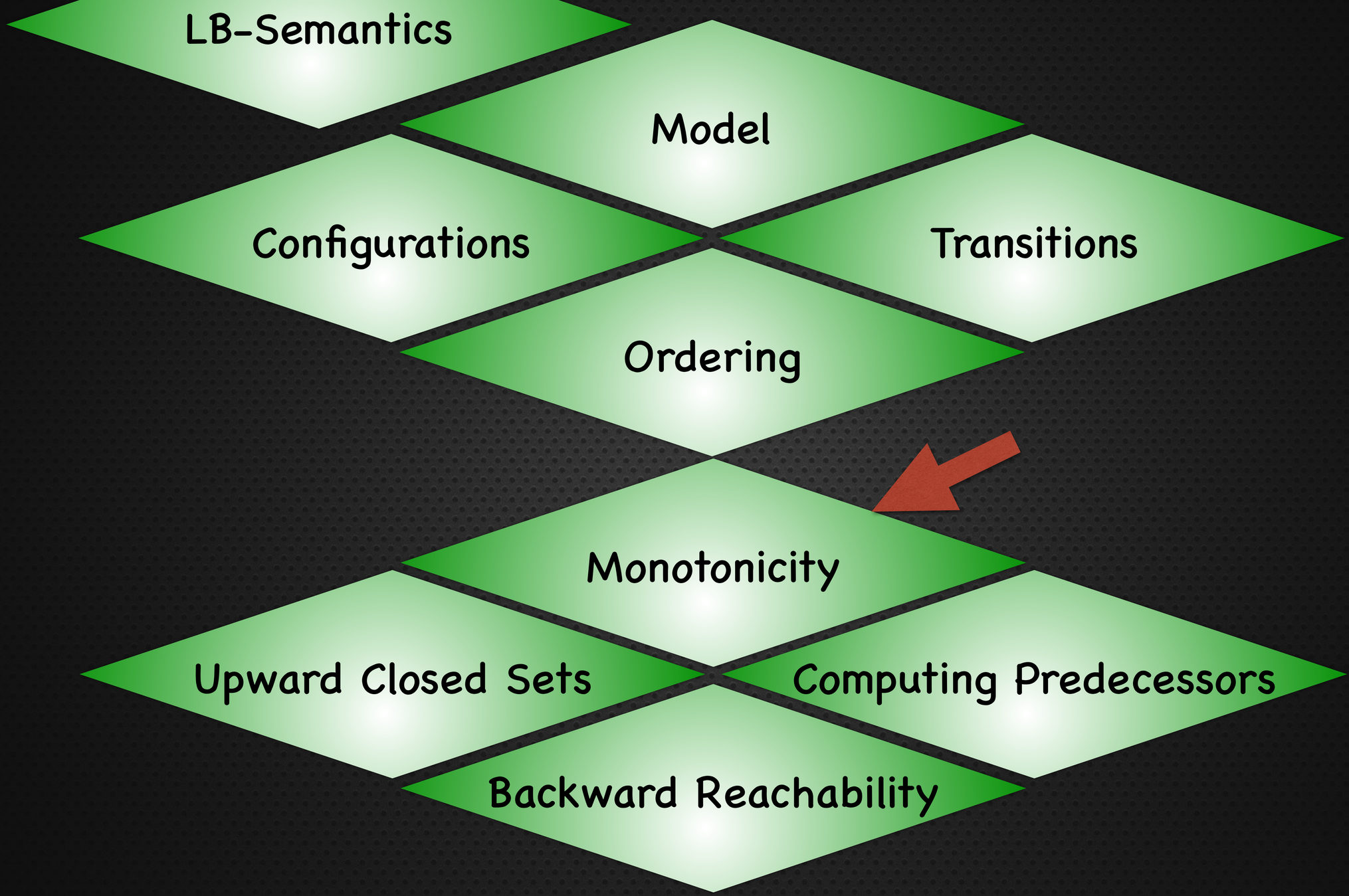


⊓ WQO



WALLPAPERSWIDE.COM


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

equality on finite sets

finite data domain

equality on finite sets

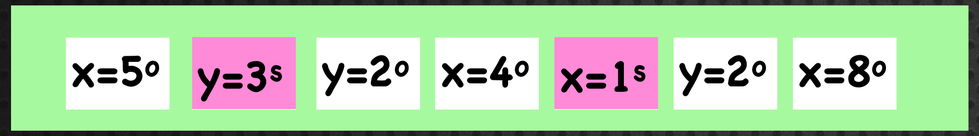
vectors of WQOs

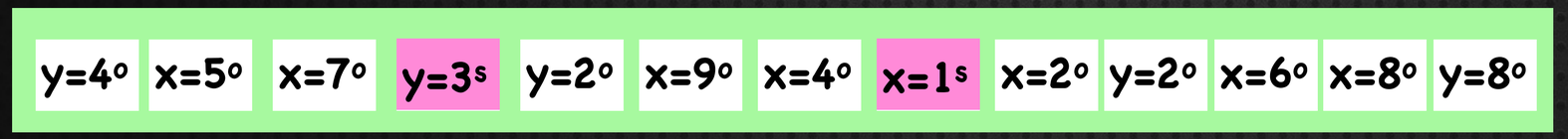
Higman's theorem

WQO

$\sqsubseteq$

vectors of words





WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

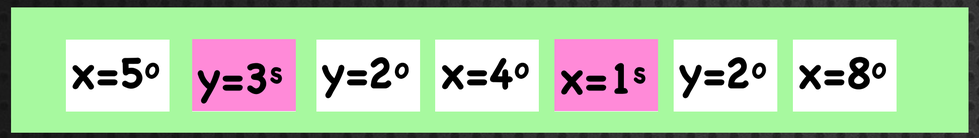
equality on finite sets

finite data domain

equality on finite sets

vectors of WQOs

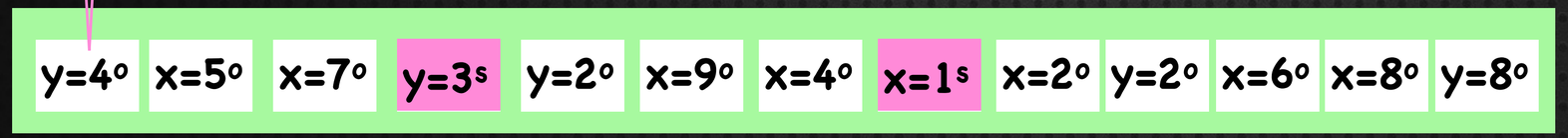
"extra messages don't harm"



WQO

Higman's theorem





vectors of words

WALLPAPERSWIDE.COM

ordering on configurations = ordering on process states + ordering on buffers + ordering on memory

- self-messages:
  - same order
  - same value
  - "sub-word in-between"

equality on finite sets

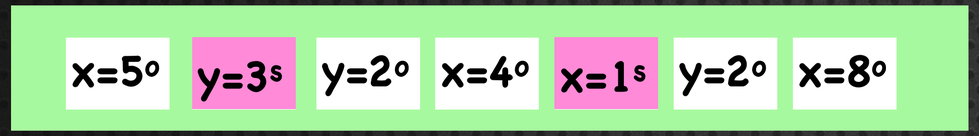
finite data domain

equality on finite sets

vectors of WQOs

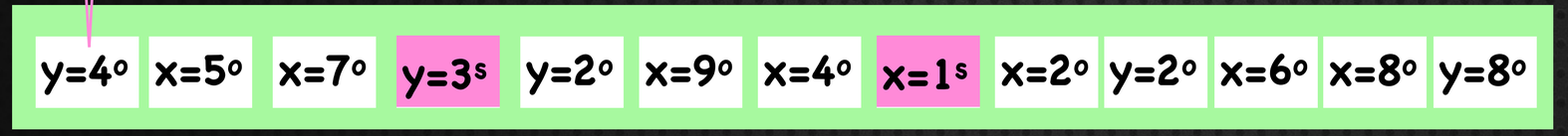
"process can throw them away"

"extra messages don't harm"



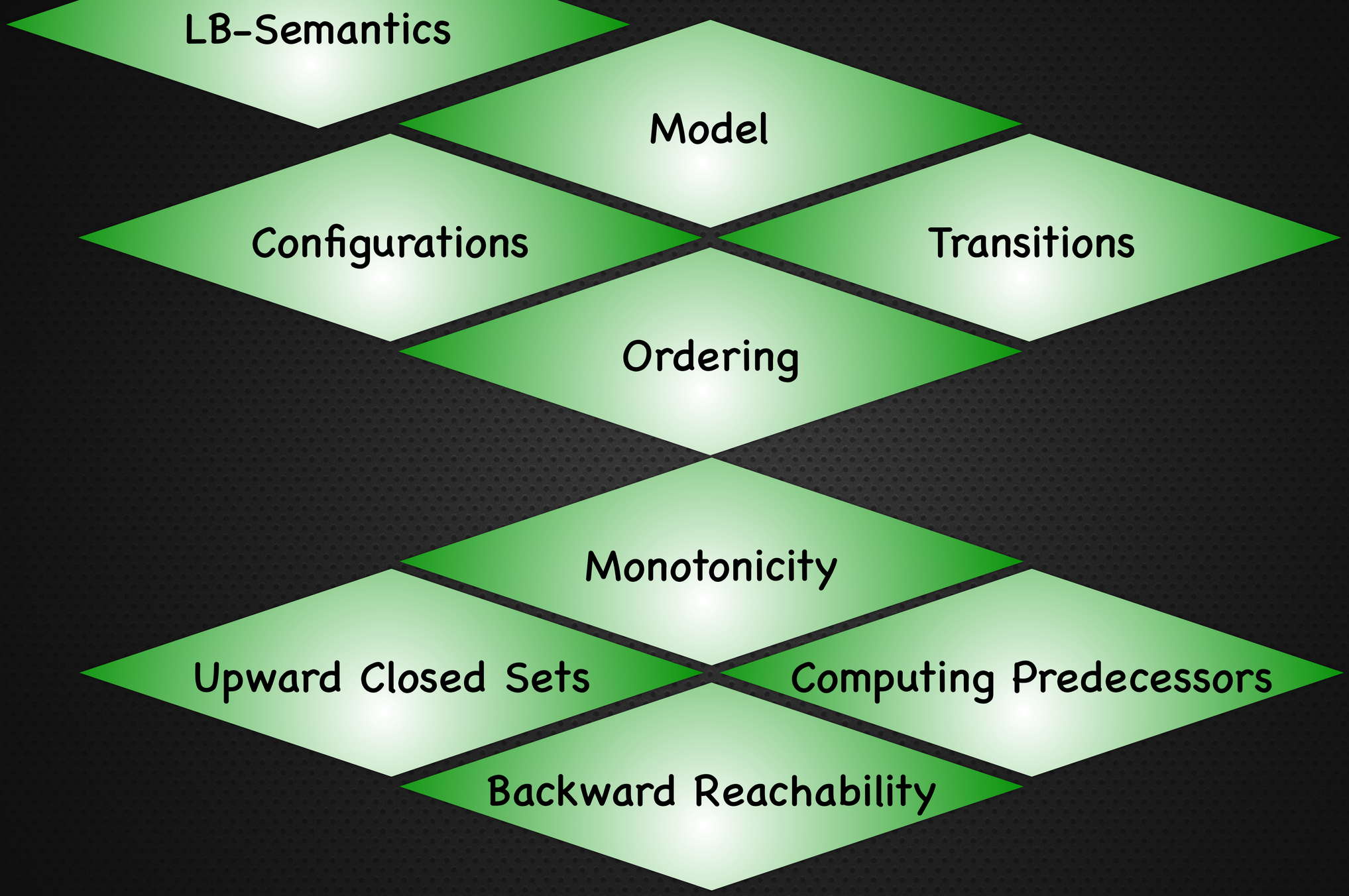
WQO

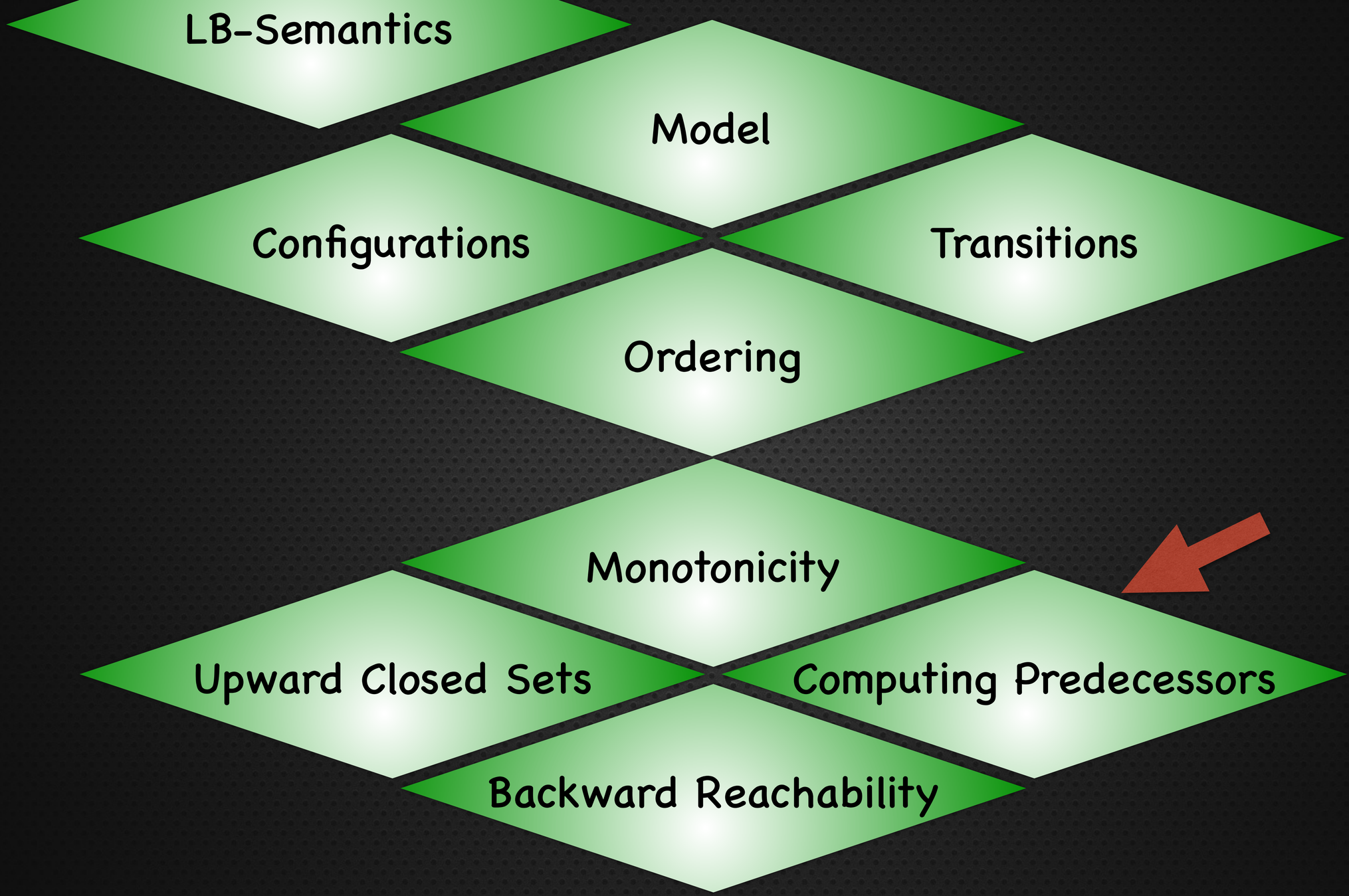
Higman's theorem



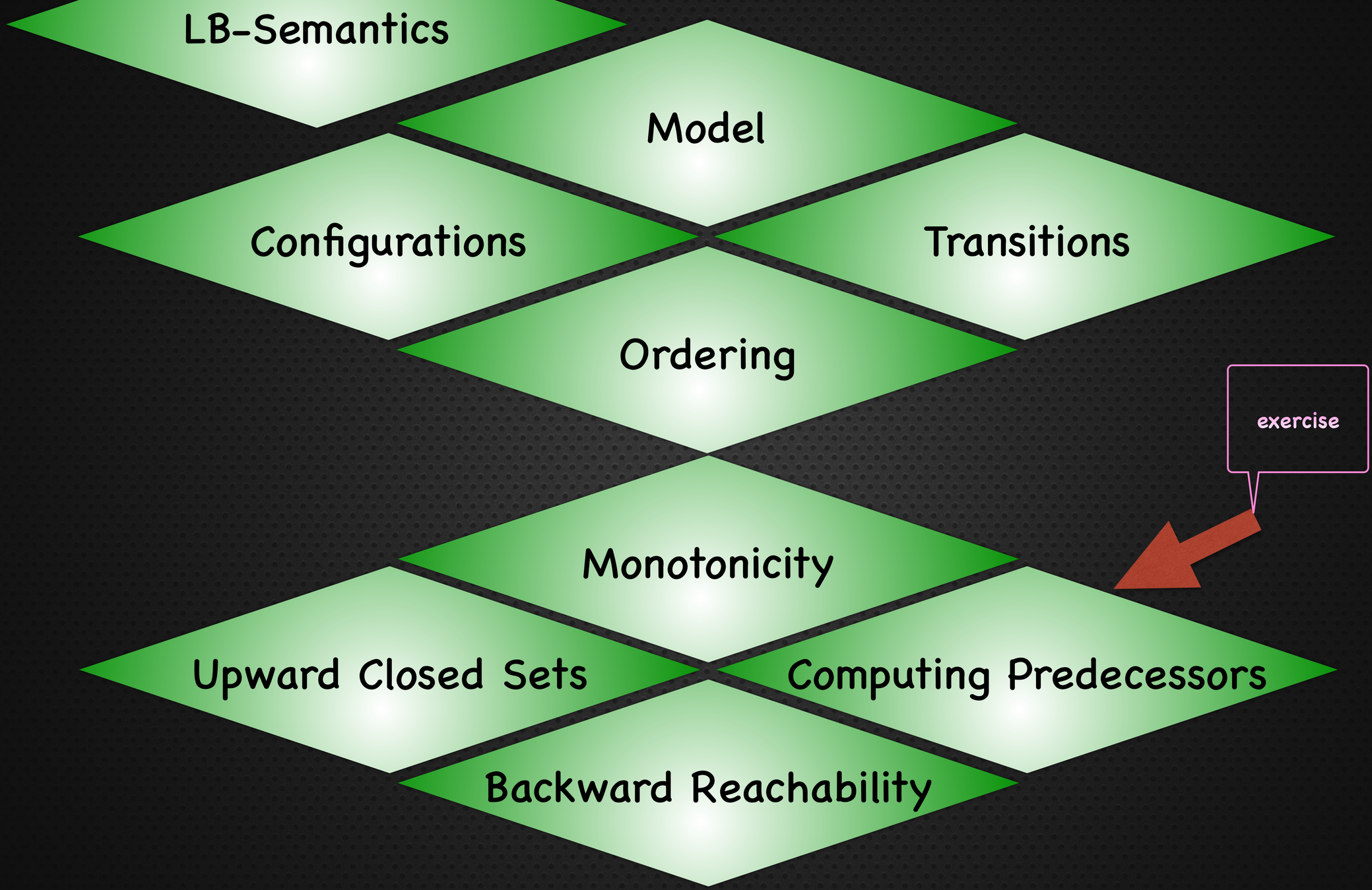
vectors of words

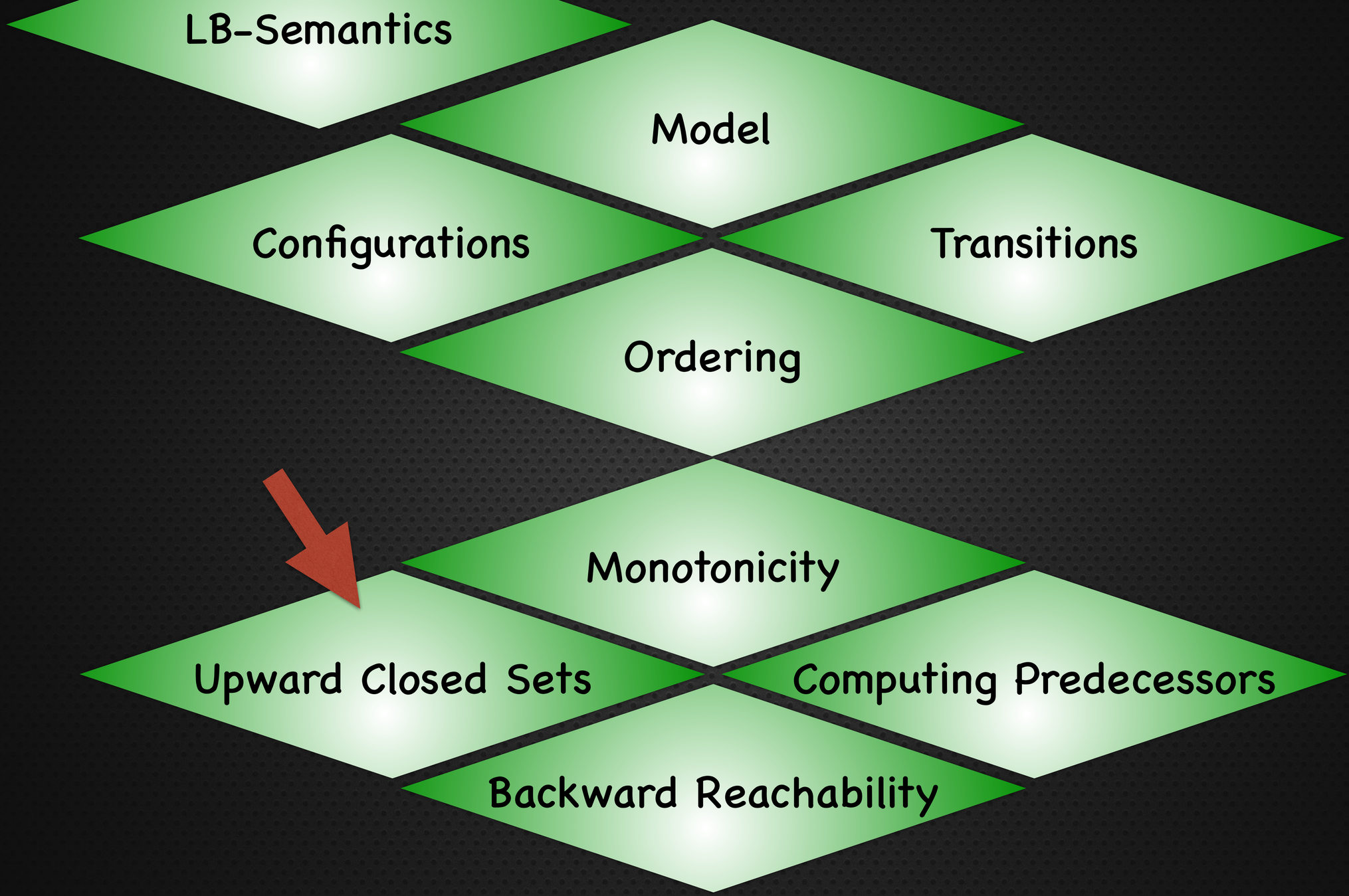
WALLPAPERSWIDE.COM

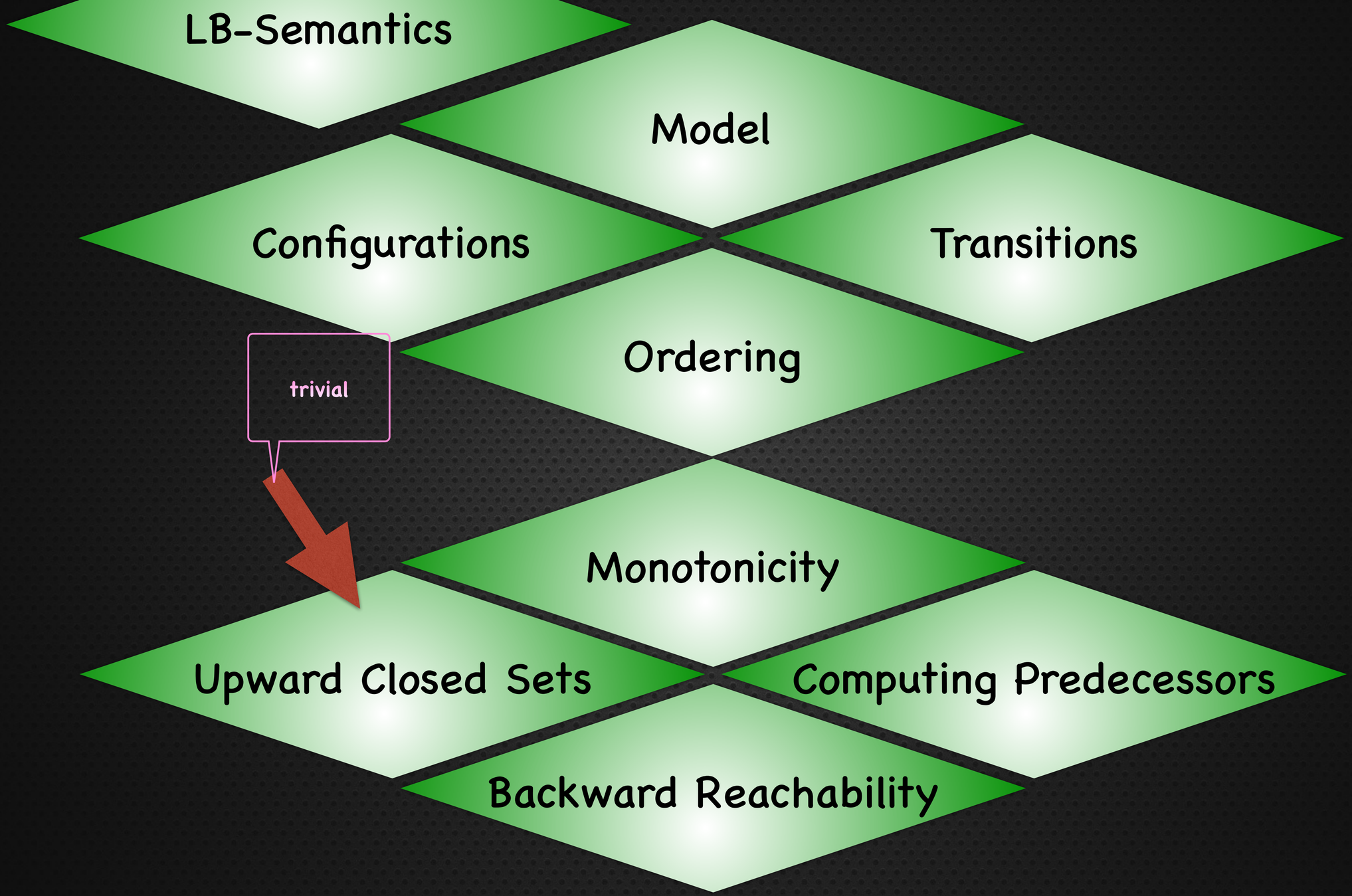

LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability



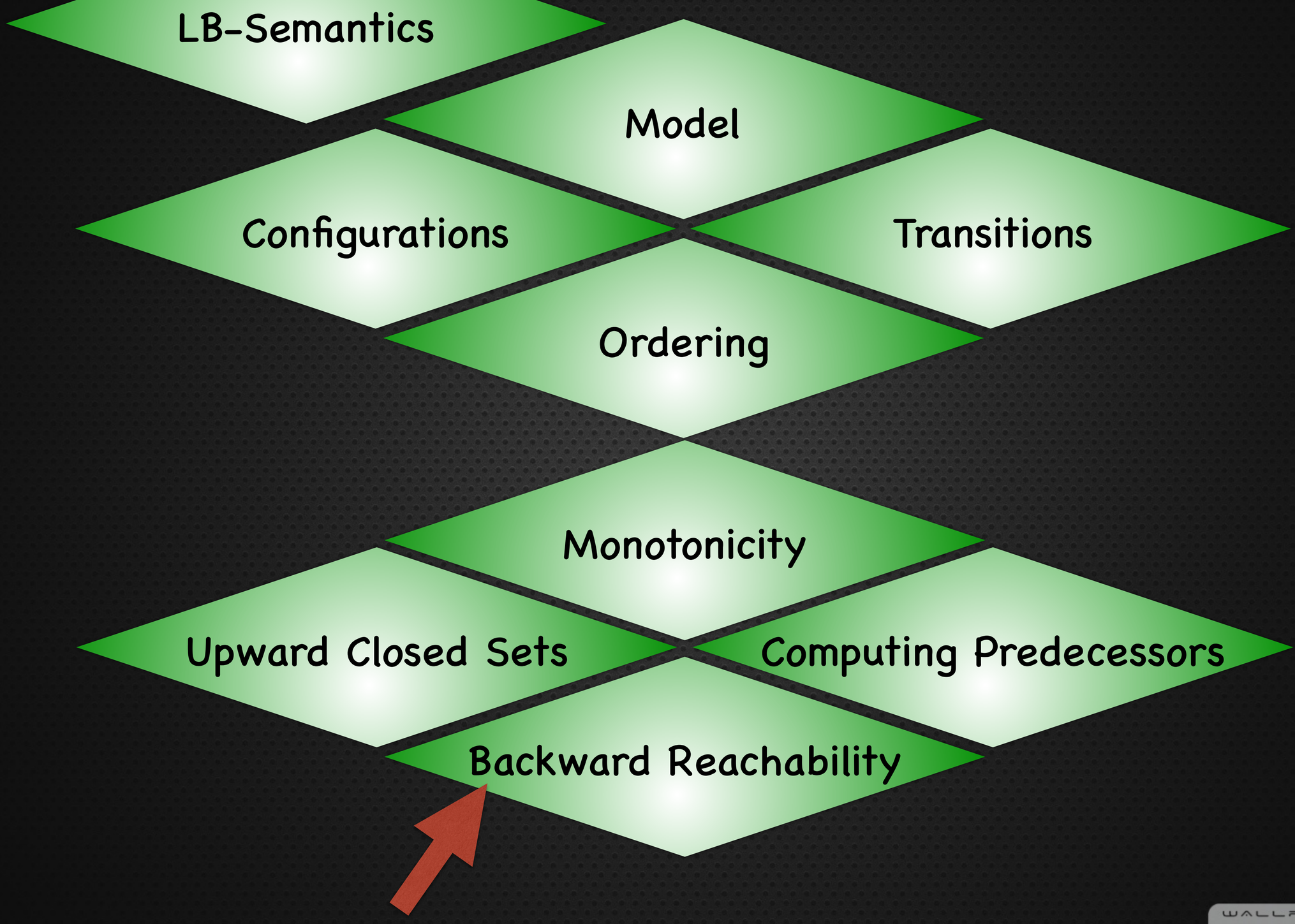
WALLPAPERSWIDE.COM

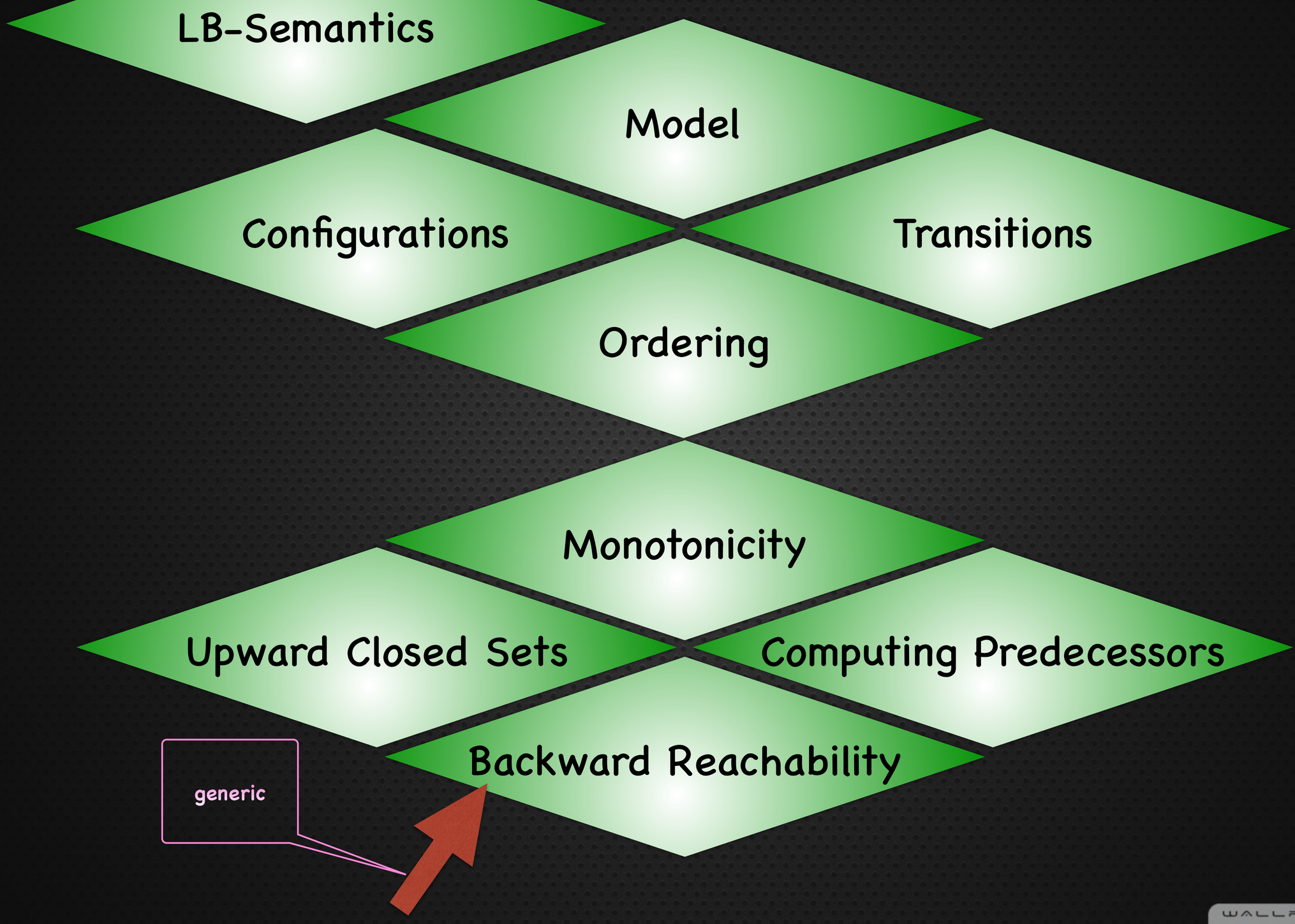

LB-Semantics
Model
Configurations
Transitions
Ordering
exercise
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability

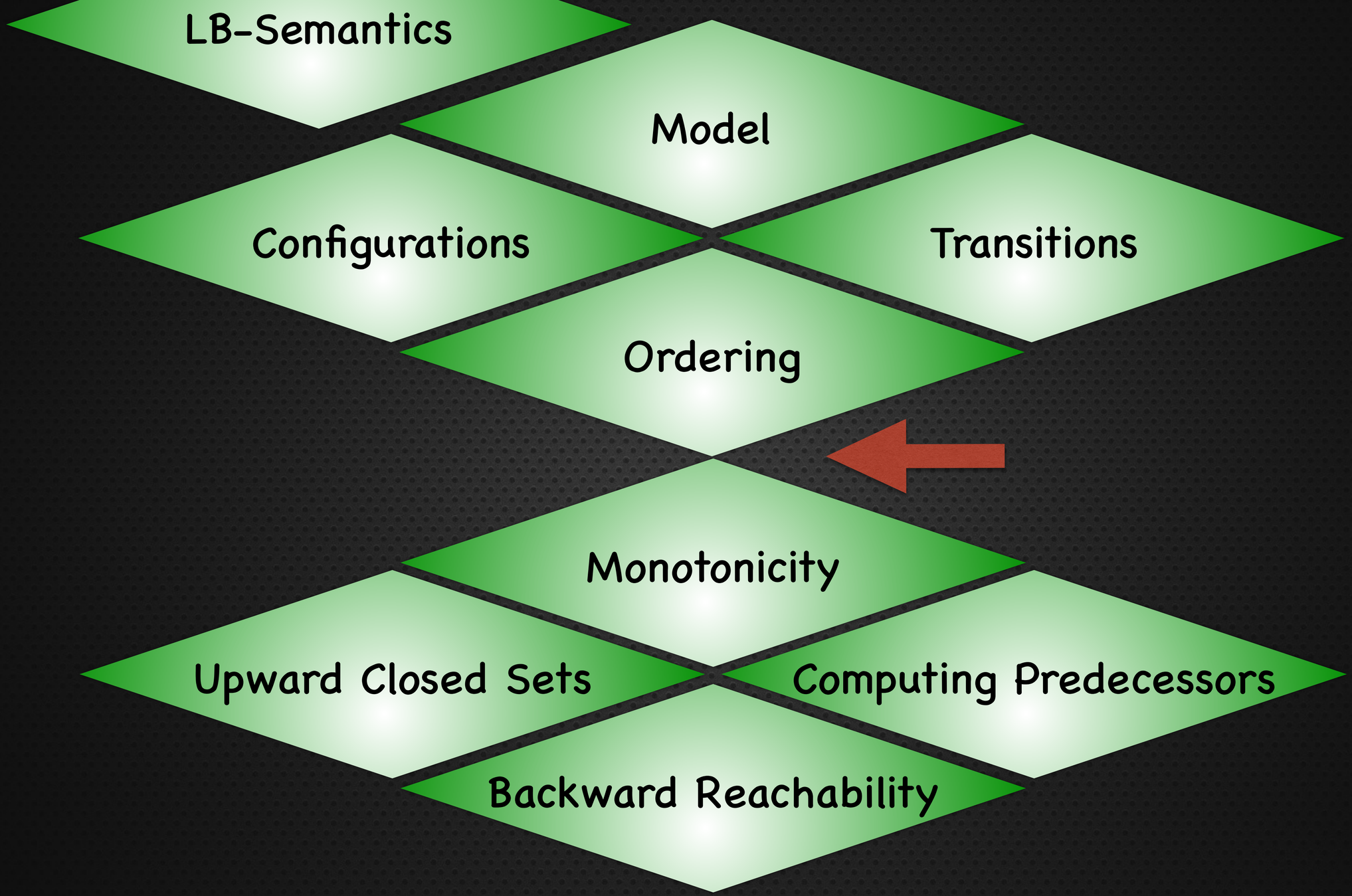

LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability

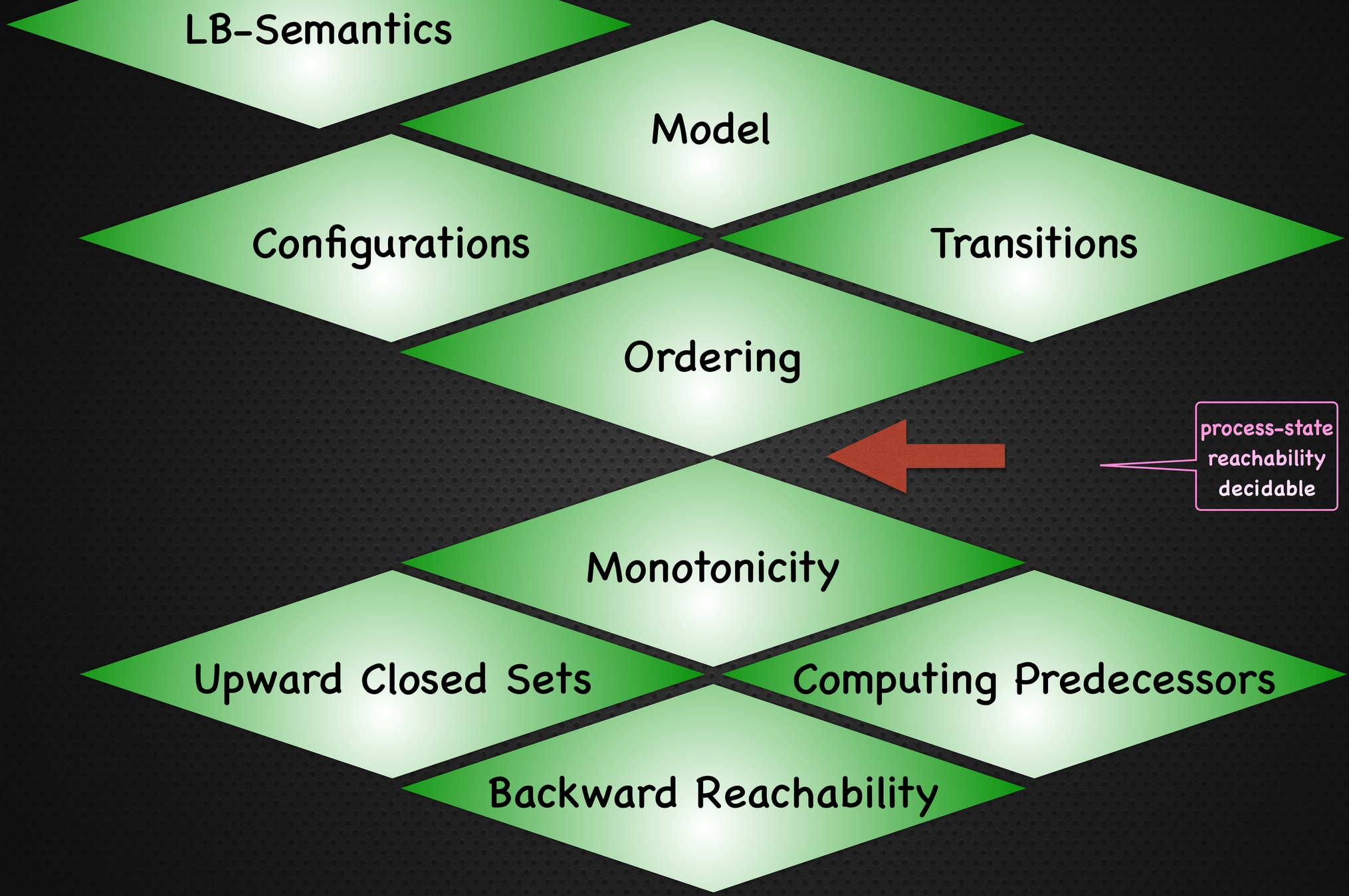


WALLPAPERSWIDE.COM

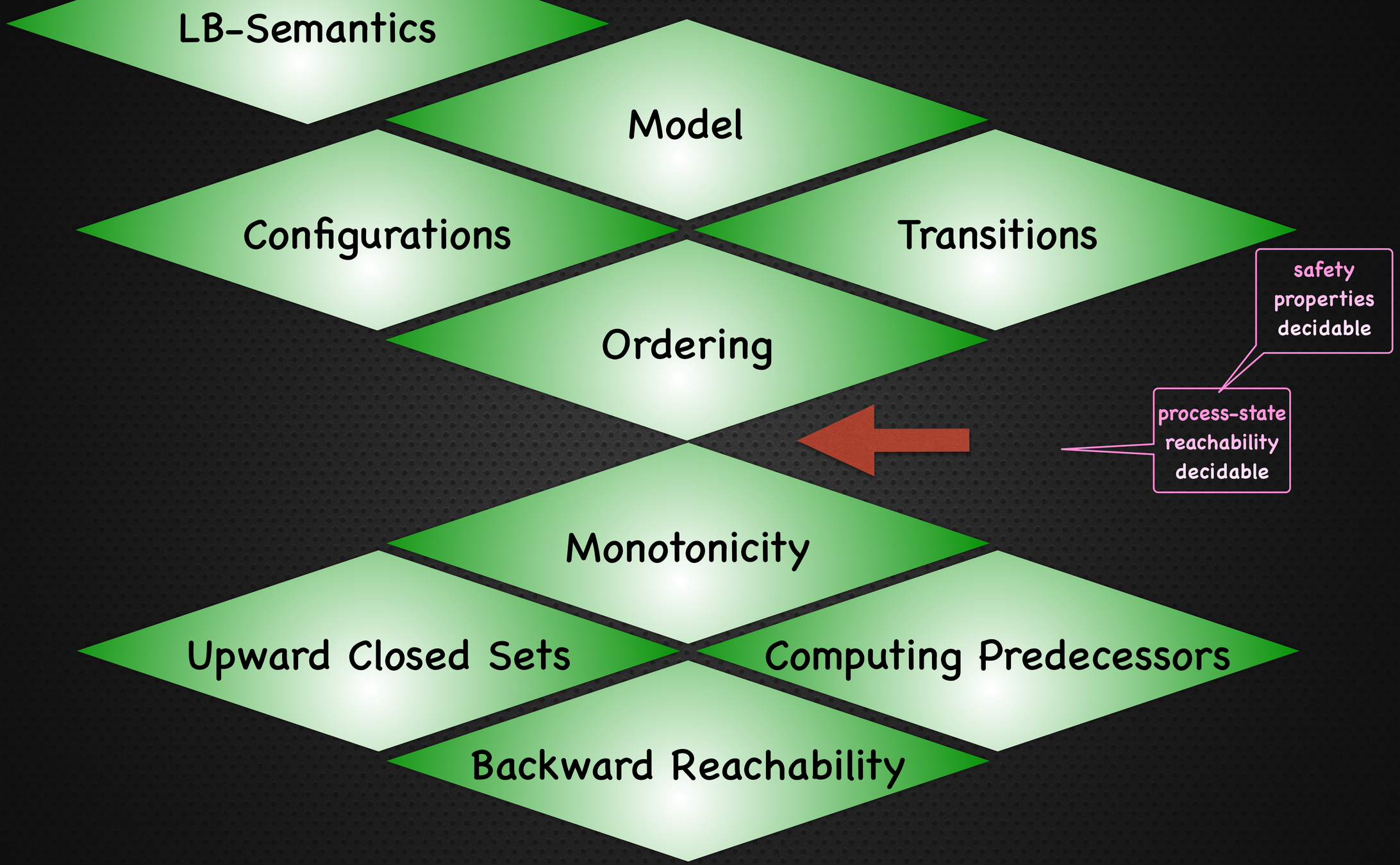

LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability

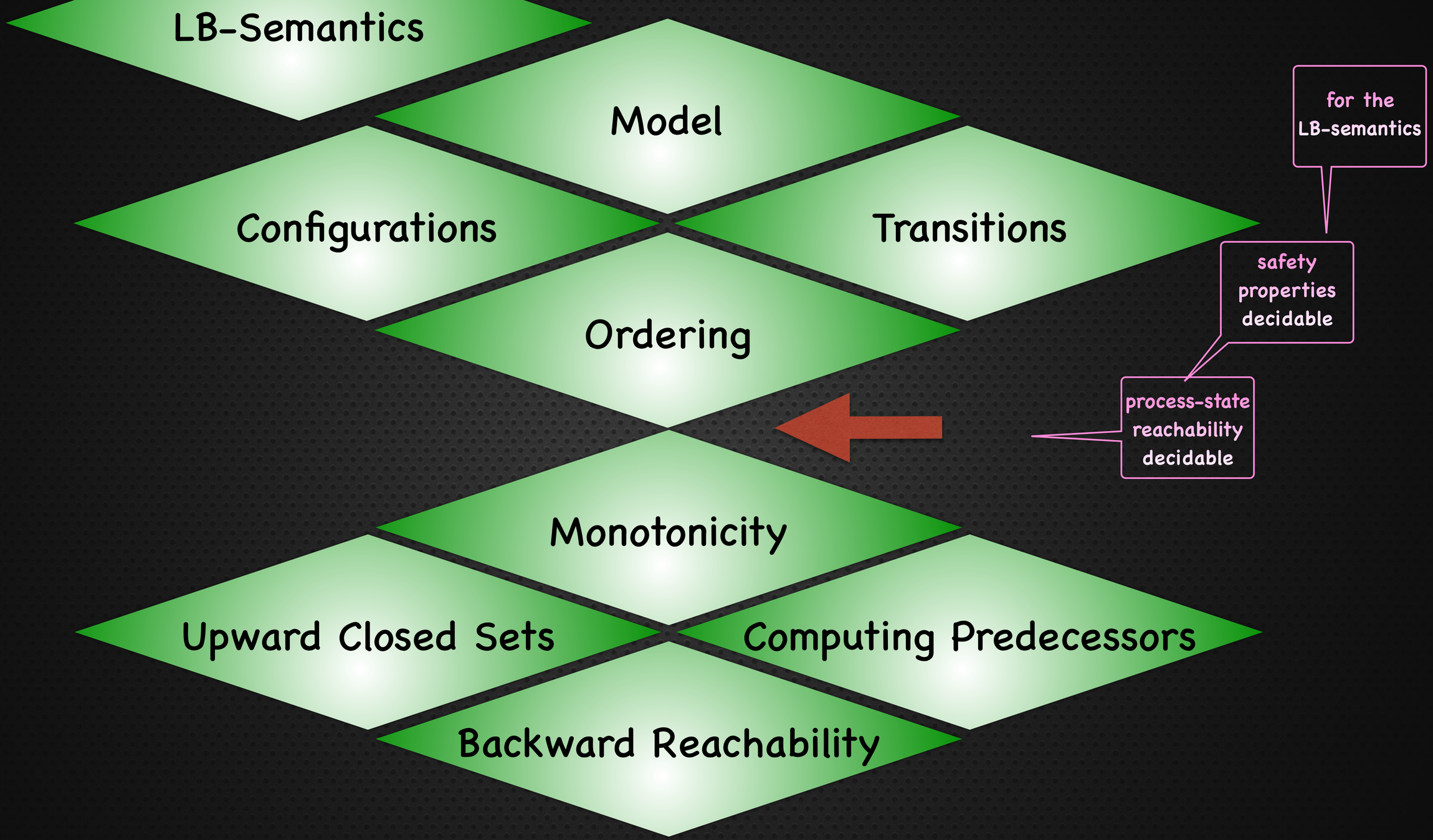

LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability
generic

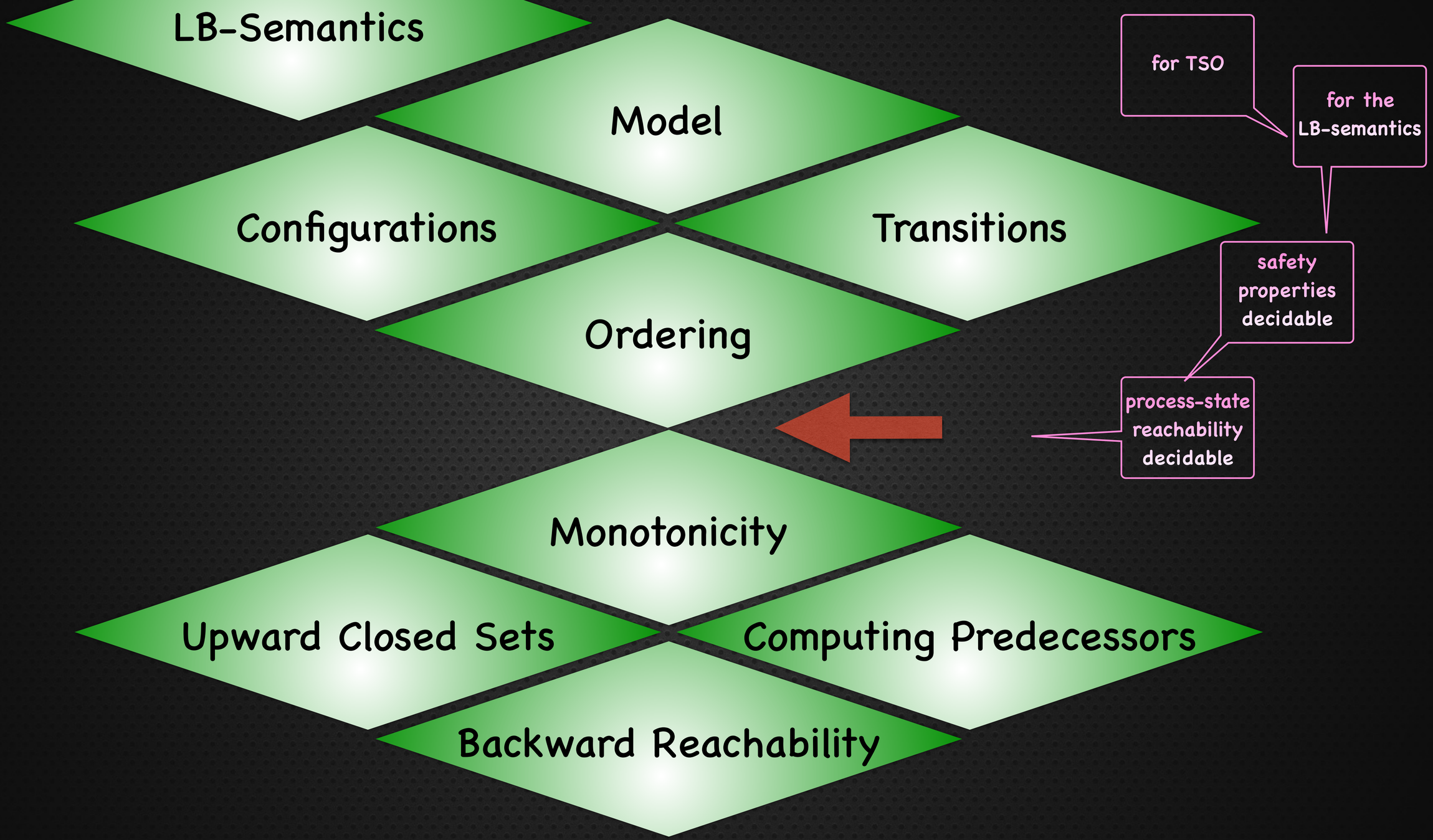

LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability



WALLPAPERSWIDE.COM


LB-Semantics
Model
Configurations
Transitions
Ordering
safety properties decidable
process-state reachability decidable
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability
for the LB-semantics
safety properties decidable
process-state reachability decidable


LB-Semantics
Model
Configurations
Transitions
Ordering
Monotonicity
Upward Closed Sets
Computing Predecessors
Backward Reachability
for TSO
for the LB-semantics
safety properties decidable
process-state reachability decidable

WALLPAPERSWIDE.COM

"Given a finite-state program running under the TSO semantics:
We can check whether the program violates a given safety property"

WALLPAPERSWIDE.COM

Experimental Results

Dual-TSO vs Memorax

- **Running time**
- **Memory consumption**

Program	#P	Dual-TSO		Memorax	
		#T	#C	#T	#C
SB	5	0.3	10641	559.7	10515914
LB	3	0.0	2048	71.4	1499475
WRC	4	0.0	1507	63.3	1398393
ISA2	3	0.0	509	21.1	226519
RWC	5	0.1	4277	61.5	1196988
W+RWC	4	0.0	1713	83.6	1389009
IRIW	4	0.0	520	34.4	358057
Nbw_w_wr	2	0.0	222	10.7	200844
Sense_rev_bar	2	0.1	1704	0.8	20577
Dekker	2	0.1	5053	1.1	19788
Dekker_simple	2	0.0	98	0.0	595
Peterson	2	0.1	5442	5.2	90301
Peterson_loop	2	0.2	7632	5.6	100082
Szymanski	2	0.6	29018	1.0	26003
MP	4	0.0	883	TO	•
Ticket_spin_lock	3	0.9	18963	TO	•
Bakery	2	2.6	82050	TO	•
Dijkstra	2	0.2	8324	TO	•
Lamport_fast	3	17.7	292543	TO	•
Burns	4	124.3	2762578	TO	•

https://www.it.uu.se/katalog/tuang296/dual-tso

Experimental Results

Single buffer approach (exact method [TACAS12+13])

Dual-TSO vs Memorax

- **Running time**
- **Memory consumption**

Program	#P	Dual-TSO		Memorax	
		#T	#C	#T	#C
SB	5	0.3	10641	559.7	10515914
LB	3	0.0	2048	71.4	1499475
WRC	4	0.0	1507	63.3	1398393
ISA2	3	0.0	509	21.1	226519
RWC	5	0.1	4277	61.5	1196988
W+RWC	4	0.0	1713	83.6	1389009
IRIW	4	0.0	520	34.4	358057
Nbw_w_wr	2	0.0	222	10.7	200844
Sense_rev_bar	2	0.1	1704	0.8	20577
Dekker	2	0.1	5053	1.1	19788
Dekker_simple	2	0.0	98	0.0	595
Peterson	2	0.1	5442	5.2	90301
Peterson_loop	2	0.2	7632	5.6	100082
Szymanski	2	0.6	29018	1.0	26003
MP	4	0.0	883	TO	•
Ticket_spin_lock	3	0.9	18963	TO	•
Bakery	2	2.6	82050	TO	•
Dijkstra	2	0.2	8324	TO	•
Lamport_fast	3	17.7	292543	TO	•
Burns	4	124.3	2762578	TO	•

https://www.it.uu.se/katalog/tuang296/dual-tso

Experimental Results

Dual-TSO vs Memorax

- **Running time**
- **Memory consumption**

standard benchmarks: litmus tests and mutual exclusion algorithms

Program	#P	Dual-TSO		Memorax	
		#T	#C	#T	#C
SB	5	0.3	10641	559.7	10515914
LB	3	0.0	2048	71.4	1499475
WRC	4	0.0	1507	63.3	1398393
ISA2	3	0.0	509	21.1	226519
RWC	5	0.1	4277	61.5	1196988
W+RWC	4	0.0	1713	83.6	1389009
IRIW	4	0.0	520	34.4	358057
Nbw_w_wr	2	0.0	222	10.7	200844
Sense_rev_bar	2	0.1	1704	0.8	20577
Dekker	2	0.1	5053	1.1	19788
Dekker_simple	2	0.0	98	0.0	595
Peterson	2	0.1	5442	5.2	90301
Peterson_loop	2	0.2	7632	5.6	100082
Szymanski	2	0.6	29018	1.0	26003
MP	4	0.0	883	TO	•
Ticket_spin_lock	3	0.9	18963	TO	•
Bakery	2	2.6	82050	TO	•
Dijkstra	2	0.2	8324	TO	•
Lamport_fast	3	17.7	292543	TO	•
Burns	4	124.3	2762578	TO	•

Experimental R

running time in seconds

Dual-TSO vs Memorax

- **Running time**
- **Memory consumption**

Program	#P	Dual-TS		Memorax	
		#T	#C	#T	#C
SB	5	0.3	10641	559.7	10515914
LB	3	0.0	2048	71.4	1499475
WRC	4	0.0	1507	63.3	1398393
ISA2	3	0.0	509	21.1	226519
RWC	5	0.1	4277	61.5	1196988
W+RWC	4	0.0	1713	83.6	1389009
IRIW	4	0.0	520	34.4	358057
Nbw_w_wr	2	0.0	222	10.7	200844
Sense_rev_bar	2	0.1	1704	0.8	20577
Dekker	2	0.1	5053	1.1	19788
Dekker_simple	2	0.0	98	0.0	595
Peterson	2	0.1	5442	5.2	90301
Peterson_loop	2	0.2	7632	5.6	100082
Szymanski	2	0.6	29018	1.0	26003
MP	4	0.0	883	TO	•
Ticket_spin_lock	3	0.9	18963	TO	•
Bakery	2	2.6	82050	TO	•
Dijkstra	2	0.2	8324	TO	•
Lamport_fast	3	17.7	292543	TO	•
Burns	4	124.3	2762578	TO	•

Experimental Rest

generated configurations

Dual-TSO vs Memorax

- **Running time**
- **Memory consumption**

Program	#P	Dual-TSO		Memorax	
		#T	#C	#T	#C
SB	5	0.3	10641	559.7	10515914
LB	3	0.0	2048	71.4	1499475
WRC	4	0.0	1507	63.3	1398393
ISA2	3	0.0	509	21.1	226519
RWC	5	0.1	4277	61.5	1196988
W+RWC	4	0.0	1713	83.6	1389009
IRIW	4	0.0	520	34.4	358057
Nbw_w_wr	2	0.0	222	10.7	200844
Sense_rev_bar	2	0.1	1704	0.8	20577
Dekker	2	0.1	5053	1.1	19788
Dekker_simple	2	0.0	98	0.0	595
Peterson	2	0.1	5442	5.2	90301
Peterson_loop	2	0.2	7632	5.6	100082
Szymanski	2	0.6	29018	1.0	26003
MP	4	0.0	883	TO	•
Ticket_spin_lock	3	0.9	18963	TO	•
Bakery	2	2.6	82050	TO	•
Dijkstra	2	0.2	8324	TO	•
Lamport_fast	3	17.7	292543	TO	•
Burns	4	124.3	2762578	TO	•

Experimental Rest

generated configurations

Dual-TSO vs Memorax

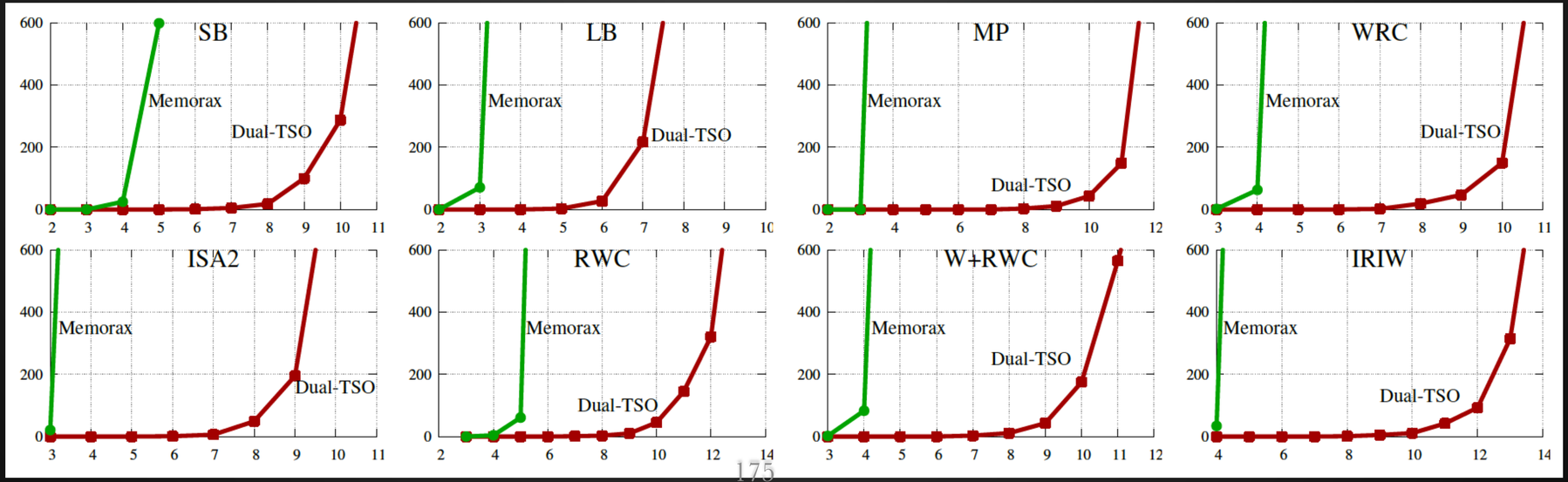
- **Running time**
- **Memory consumption**

Dual-TSO is faster and uses less memory in most of examples

Program	#P	Dual-TSO		Memorax	
		#T	#C	#T	#C
SB	5	0.3	10641	559.7	10515914
LB	3	0.0	2048	71.4	1499475
WRC	4	0.0	1507	63.3	1398393
ISA2	3	0.0	509	21.1	226519
RWC	5	0.1	4277	61.5	1196988
W+RWC	4	0.0	1713	83.6	1389009
IRIW	4	0.0	520	34.4	358057
Nbw_w_wr	2	0.0	222	10.7	200844
Sense_rev_bar	2	0.1	1704	0.8	20577
Dekker	2	0.1	5053	1.1	19788
Dekker_simple	2	0.0	98	0.0	595
Peterson	2	0.1	5442	5.2	90301
Peterson_loop	2	0.2	7632	5.6	100082
Szymanski	2	0.6	29018	1.0	26003
MP	4	0.0	883	TO	•
Ticket_spin_lock	3	0.9	18963	TO	•
Bakery	2	2.6	82050	TO	•
Dijkstra	2	0.2	8324	TO	•
Lamport_fast	3	17.7	292543	TO	•
Burns	4	124.3	2762578	TO	•

Experimental Results
Parameterised Cases

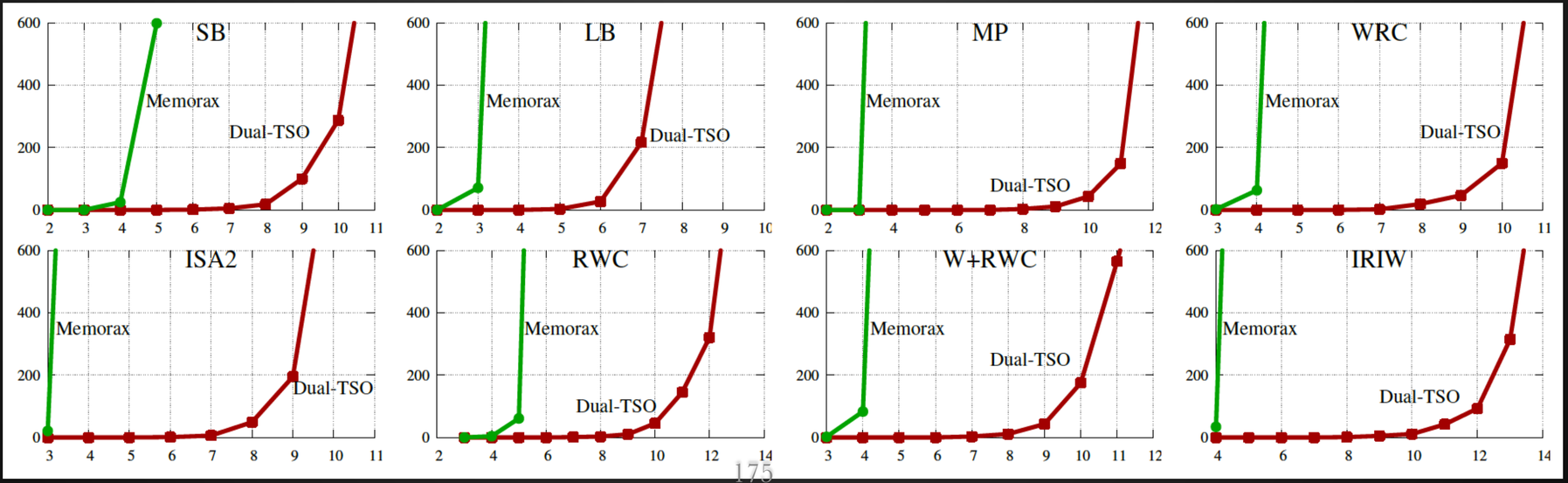
Program	Dual-TSO	
	#T	#C
SB	0.0	147
LB	0.6	1028
MP	0.0	149
WRC	0.8	618
ISA2	4.3	1539
RWC	0.2	293
W+RWC	1.5	828
IRIW	4.6	648



Experimental Results
Parameterised Cases

unbounded number of processes

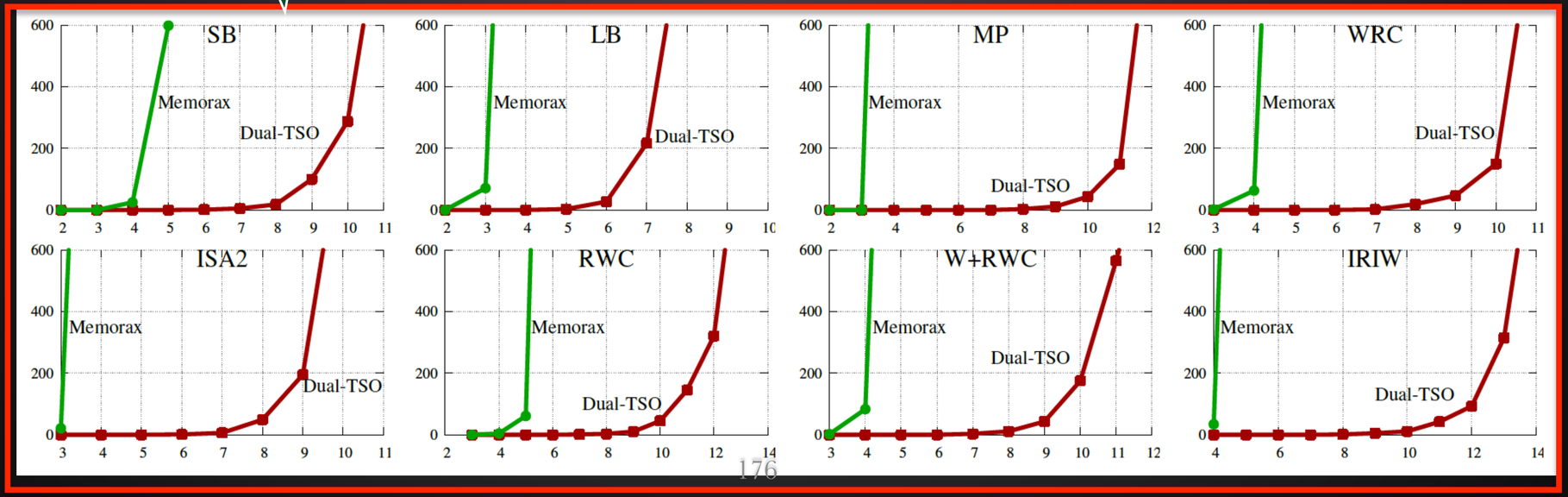
Program	Dual-TSO	
	#T	#C
SB	0.0	147
LB	0.6	1028
MP	0.0	149
WRC	0.8	618
ISA2	4.3	1539
RWC	0.2	293
W+RWC	1.5	828
IRIW	4.6	648



Experimental Results
Parameterised Cases

Program	Dual-TSO	
	#T	#C
SB	0.0	147
LB	0.6	1028
MP	0.0	149
WRC	0.8	618
ISA2	4.3	1539
RWC	0.2	293
W+RWC	1.5	828
IRIW	4.6	648

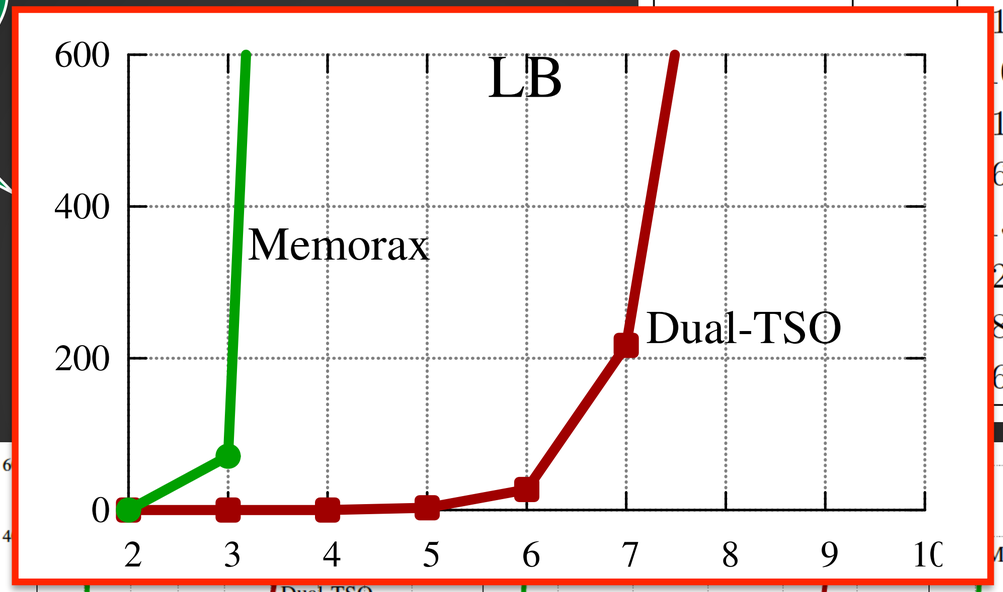
increasing the number of processes

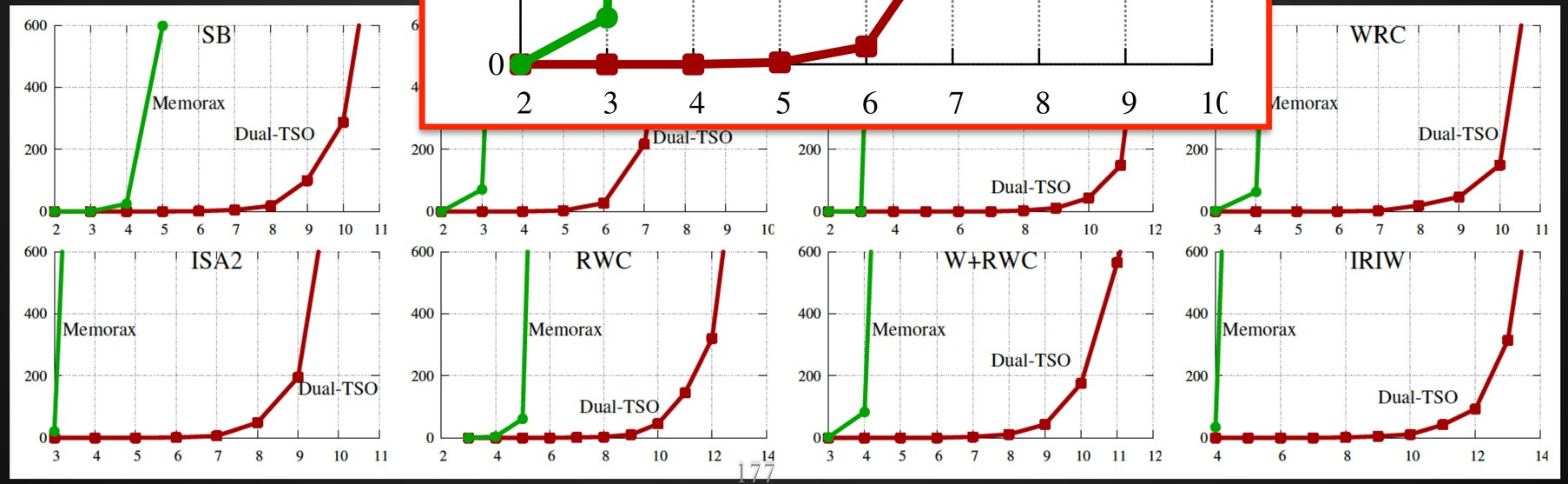


Experimental Results
Parameterised Cases

Dual-TSO is more scalable

Program	Dual-TSO	
	#T	#C
		147
		028
		149
		618
		539
		293
		828
		648

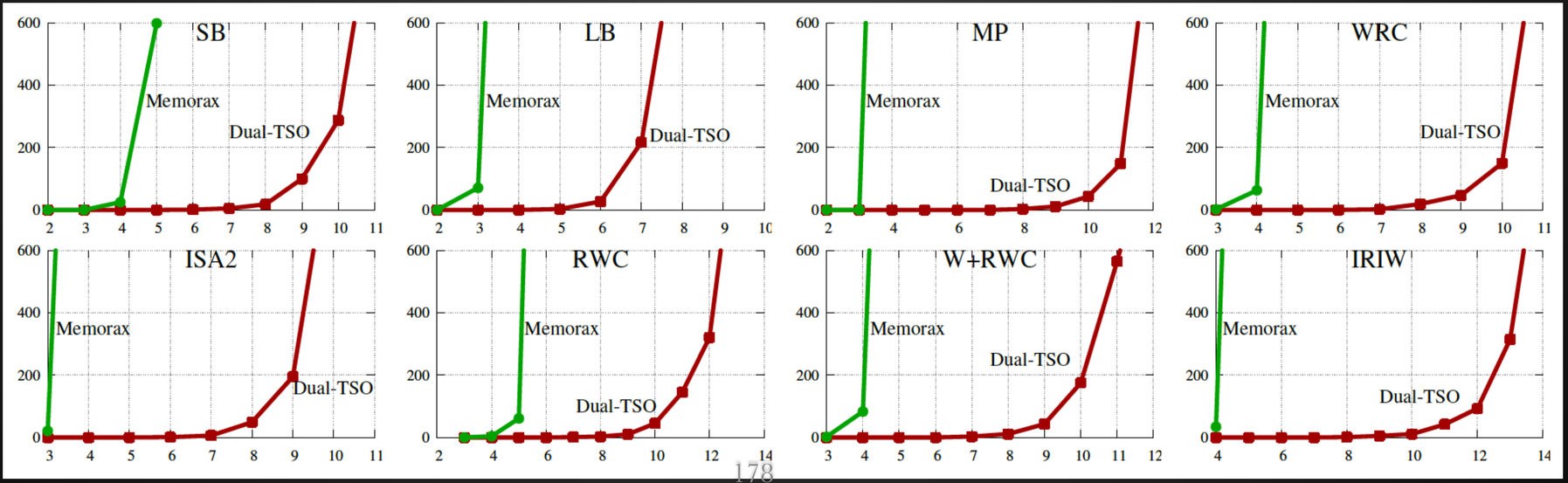




Experimental Results
Parameterised Cases

Dual-TSO is more efficient and scalable

Program	Dual-TSO	
	#T	#C
SB	0.0	147
LB	0.6	1028
MP	0.0	149
WRC	0.8	618
ISA2	4.3	1539
RWC	0.2	293
W+RWC	1.5	828
IRIW	4.6	648



Shared Variable Concurrency

- Sequential Consistency (SC)
- Total Store Ordering (TSO)
- Memory Fences
- Non-monotonicity
- Load-Buffer Semantics

WALLPAPERSWIDE.COM