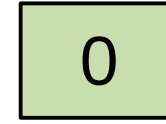
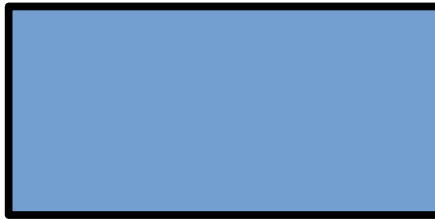
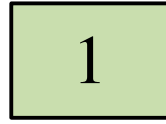


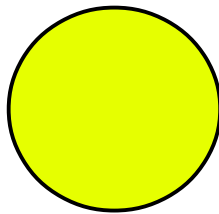
Monitor with Semaphores

Monitor

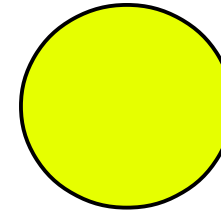
Full –
0 when full



Empty –
0 when empty



Producer



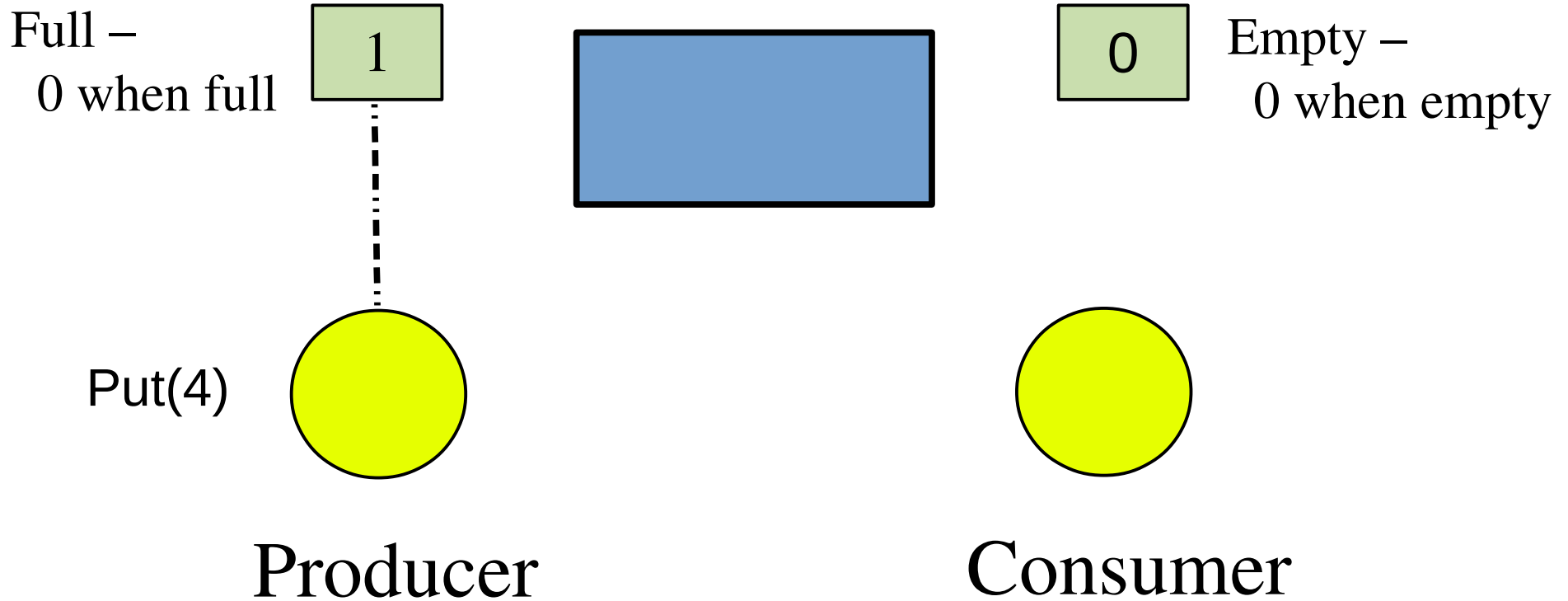
Consumer

Monitor uses a buffer capable of holding one token

Count of Full semaphore goes to 0 when the buffer is full

Count of Empty semaphore goes to 0 when the buffer is empty

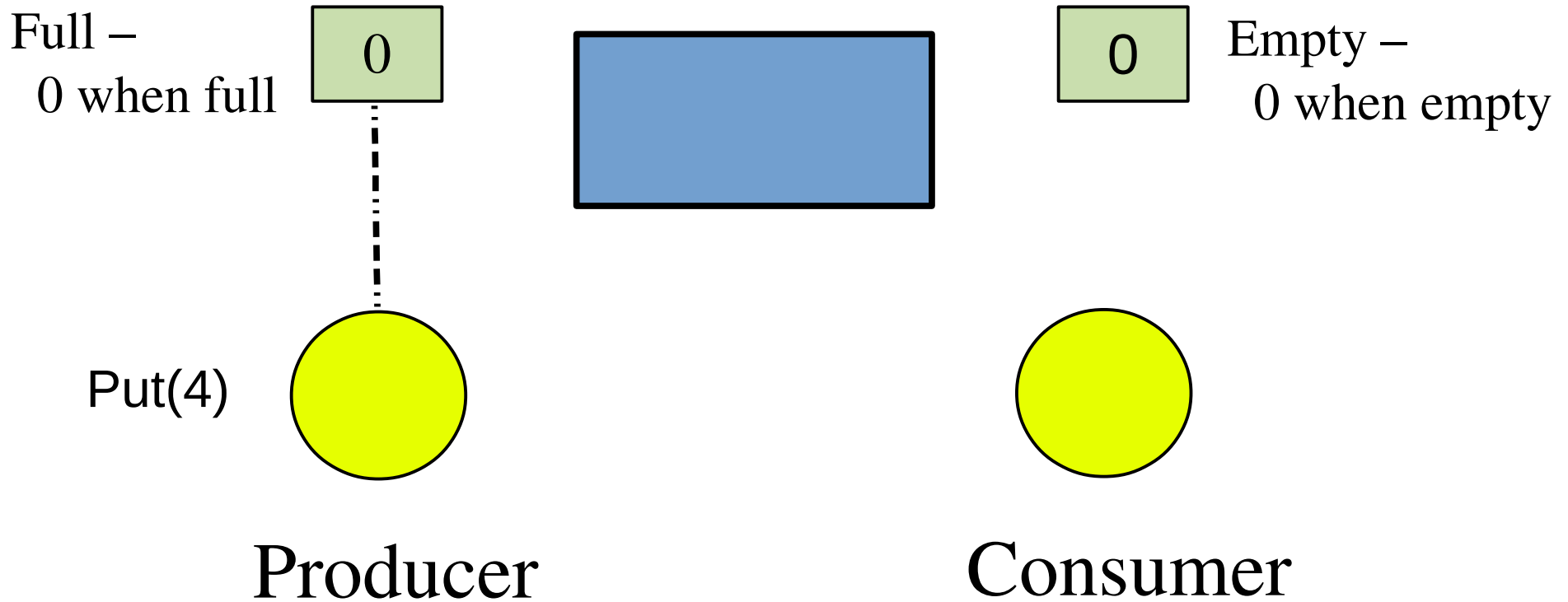
Monitor



Monitor buffer is empty

The Full semaphore's count is checked to be > 0

Monitor

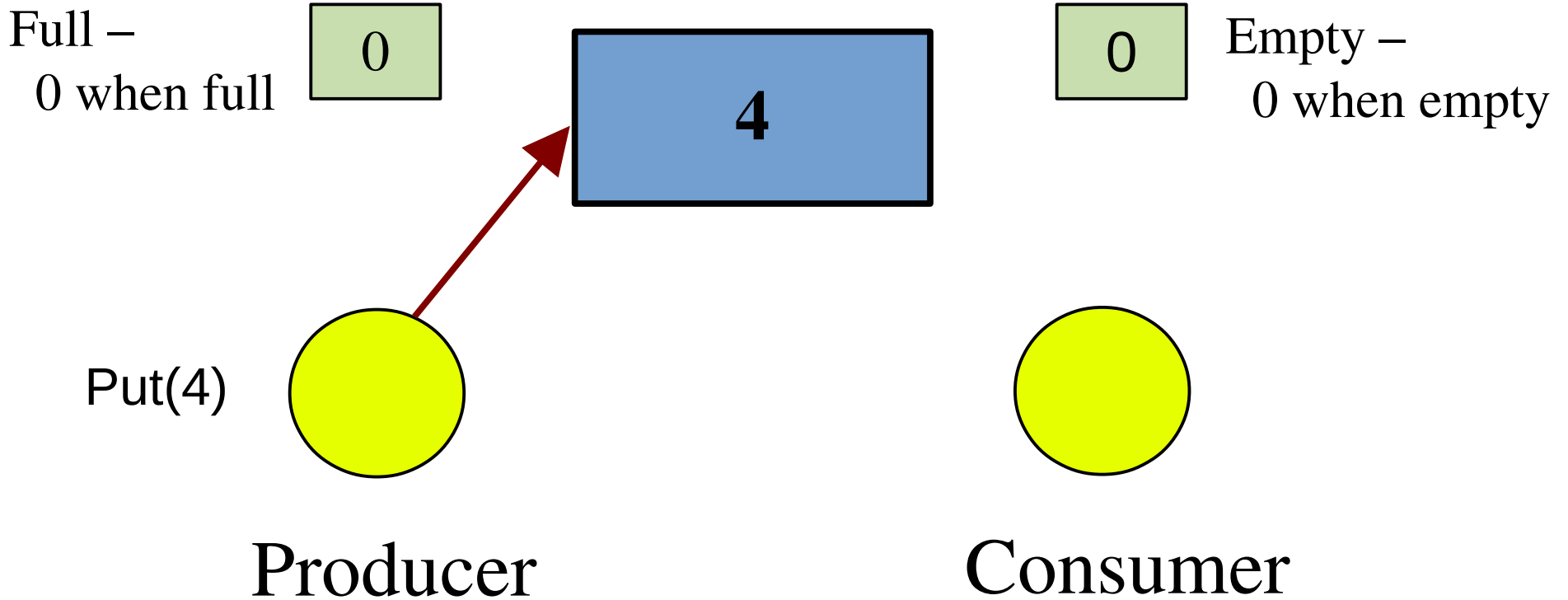


Monitor buffer is empty

The Full semaphore's count is checked to be > 0

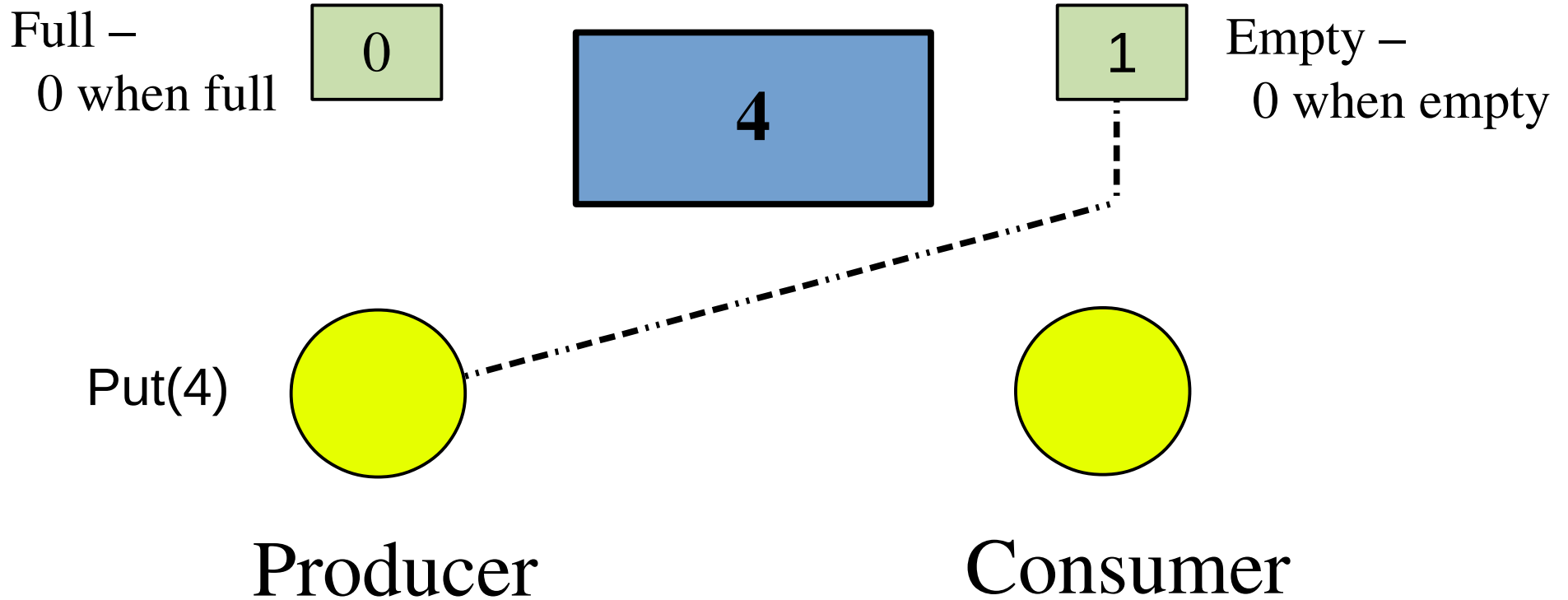
It is! So producer passes through and decrements Full's count

Monitor



Producer sends a 4 into the Monitor's buffer
Monitor's buffer now contains 4

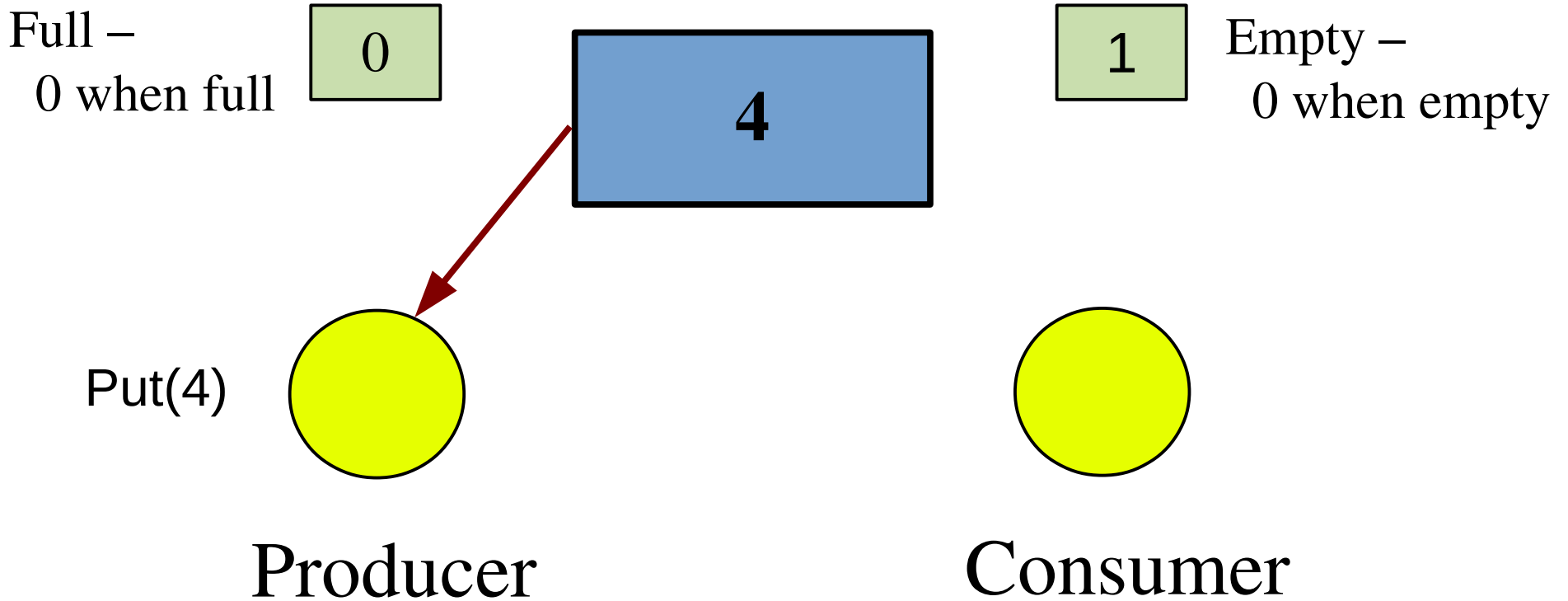
Monitor



Monitor's buffer now contains 4

Producer increments the Empty semaphore's count

Monitor



Monitor's buffer now contains 4
Producer returns and continues processing

Monitor

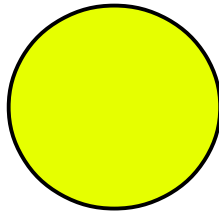
Full –
0 when full

0

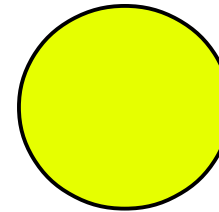
4

1

Empty –
0 when empty



Producer

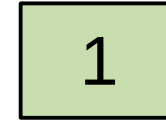
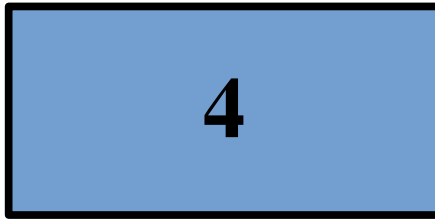
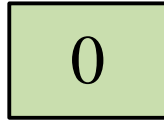


Consumer

Monitor's buffer now contains 4
The Full semaphore's count is 0
The Empty semaphore's count is 1

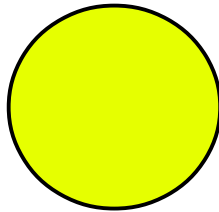
Monitor

Full –
0 when full

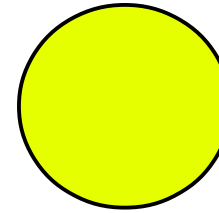


Empty –
0 when empty

Put(6)



Producer

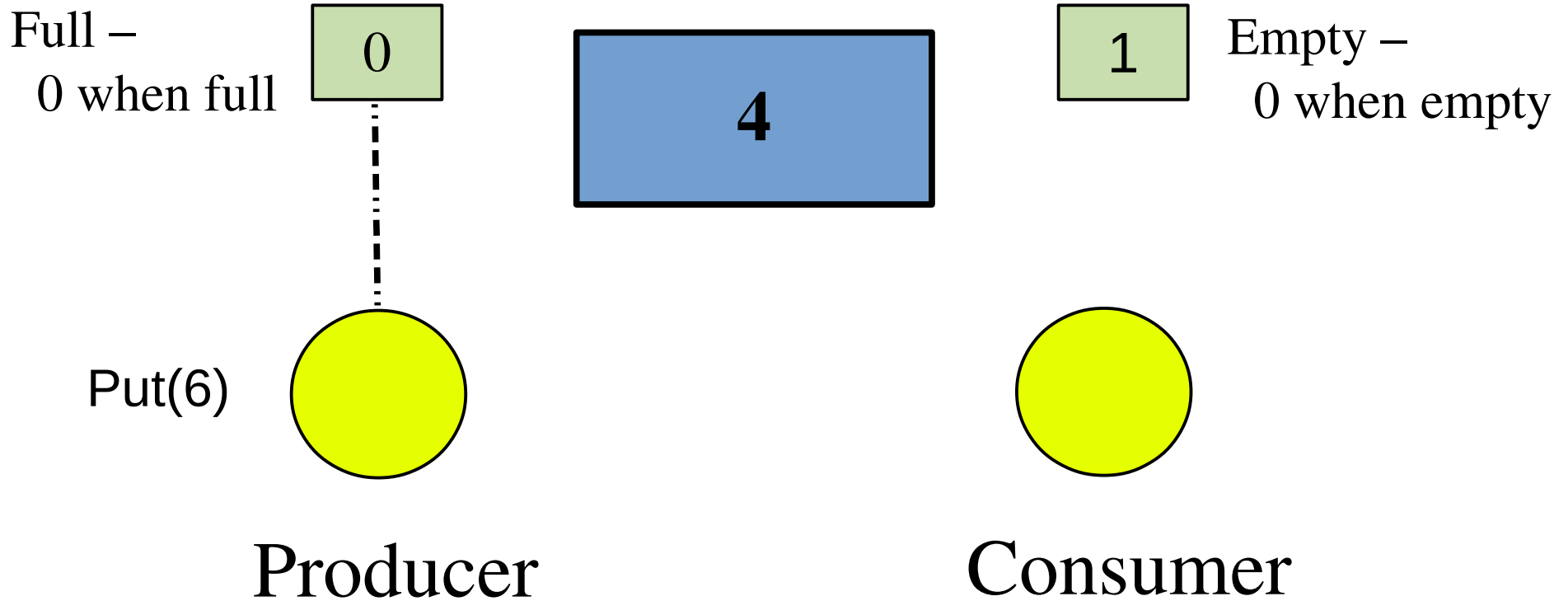


Consumer

Monitor's buffer now contains 4

The producer attempts to put a 6 into the monitor

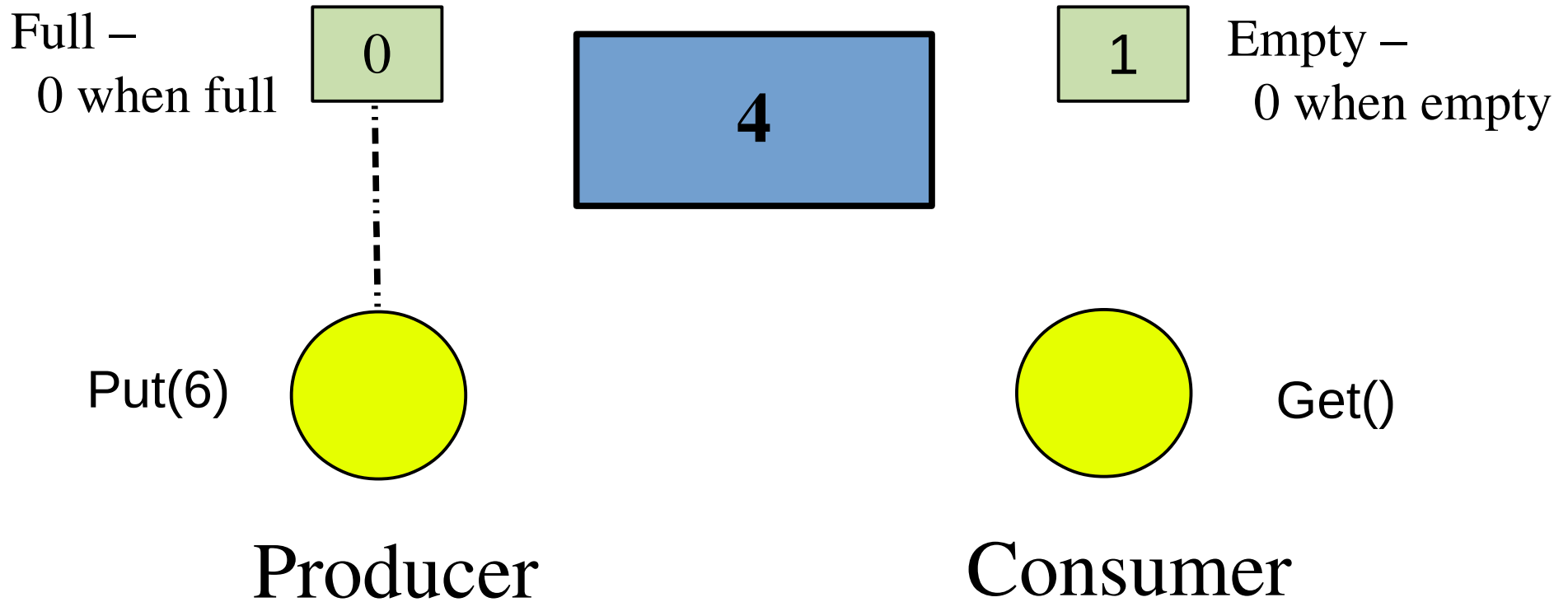
Monitor



Monitor's buffer now contains 4

The Full semaphore's count is checked but is 0, producer is blocked!

Monitor

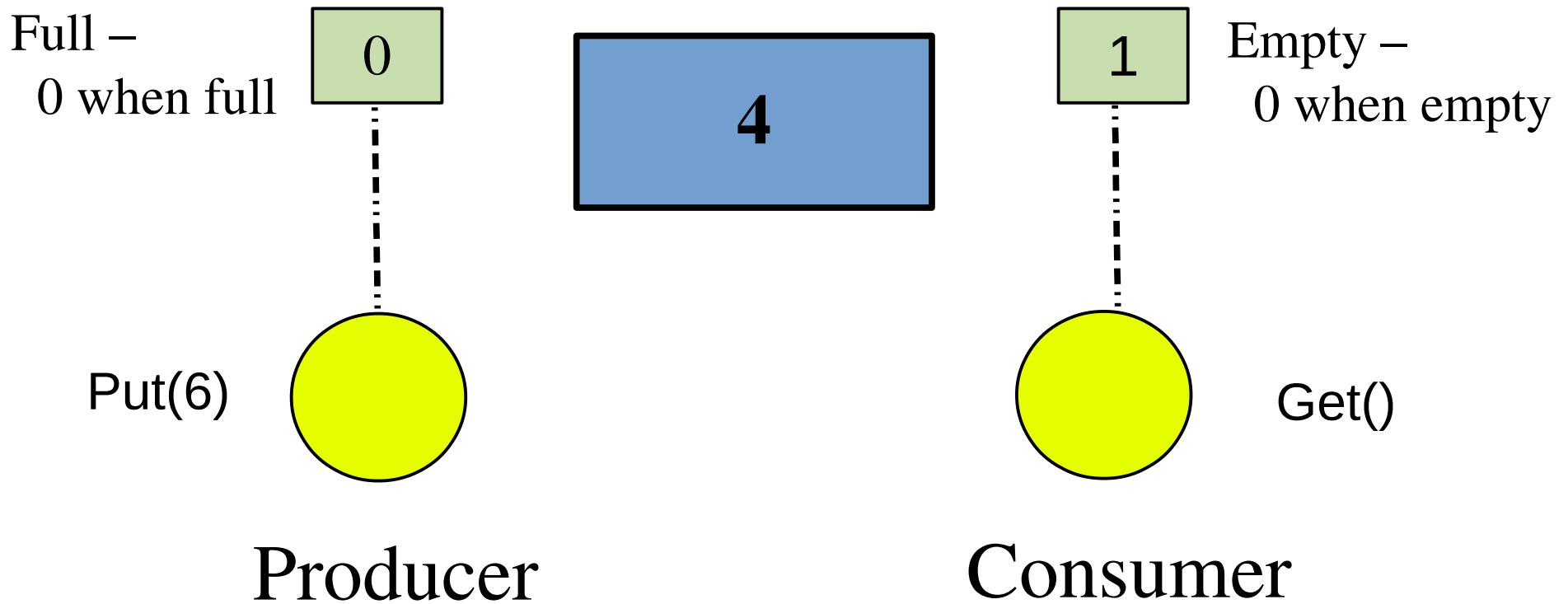


Monitor's buffer now contains 4

The Full semaphore's count is checked but is 0, producer is blocked!

Now the consumer executes a get()

Monitor

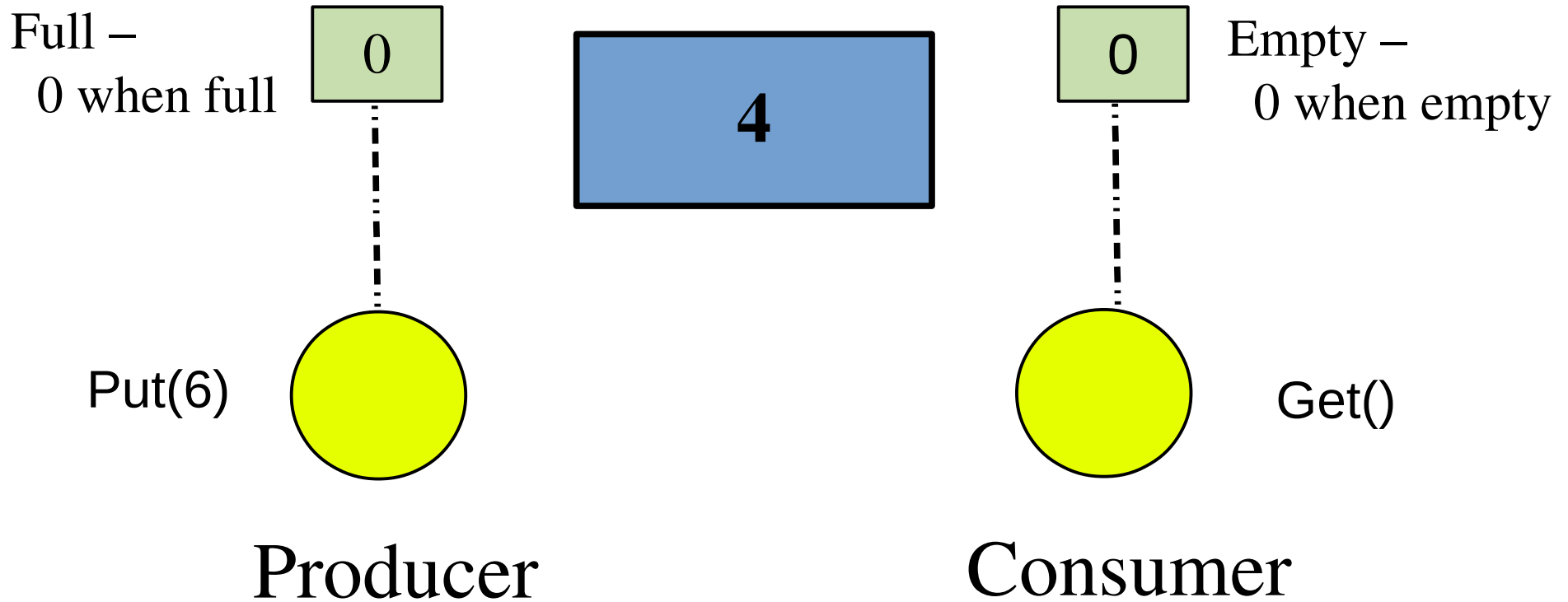


Monitor's buffer now contains 4

The Full semaphore's count is checked but is 0, producer is blocked!

The Empty semaphore's count is checked to be > 0

Monitor



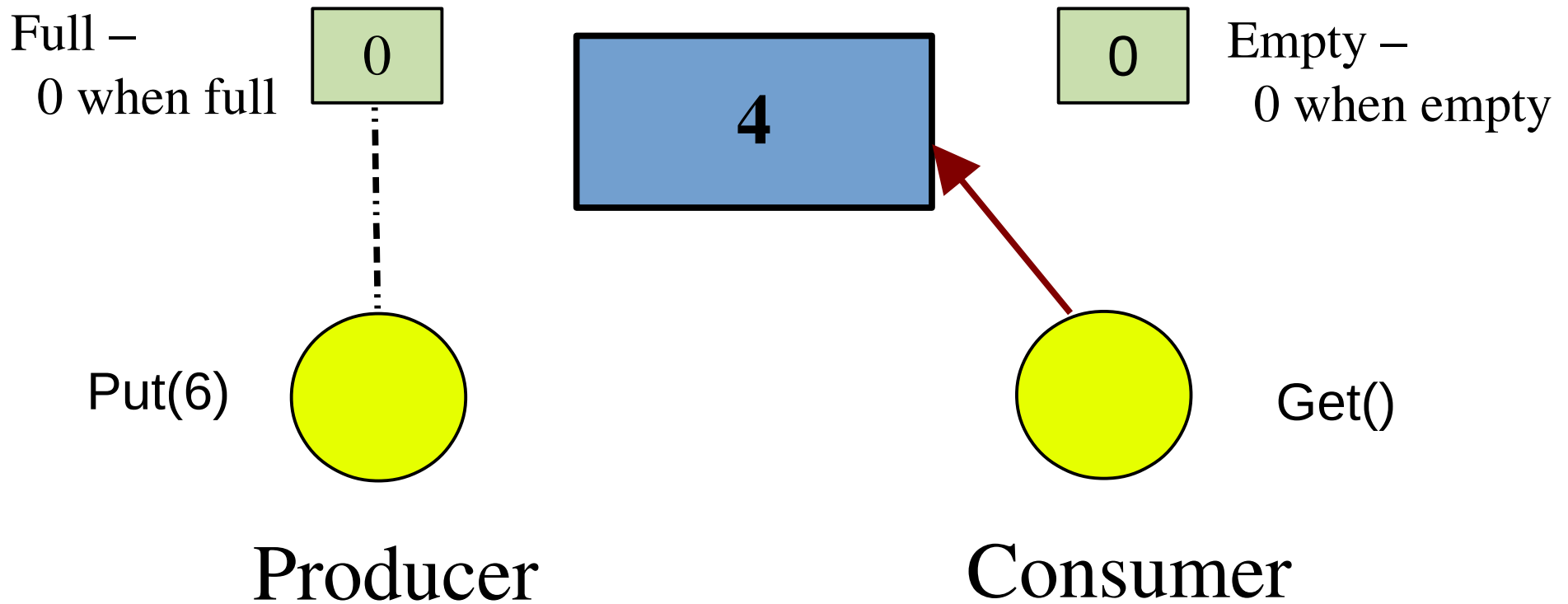
Monitor's buffer now contains 4

The Full semaphore's count is checked but is 0, producer is blocked!

The Empty semaphore's count is checked to be > 0

It is! So consumer passes through and decrements Empty's count

Monitor

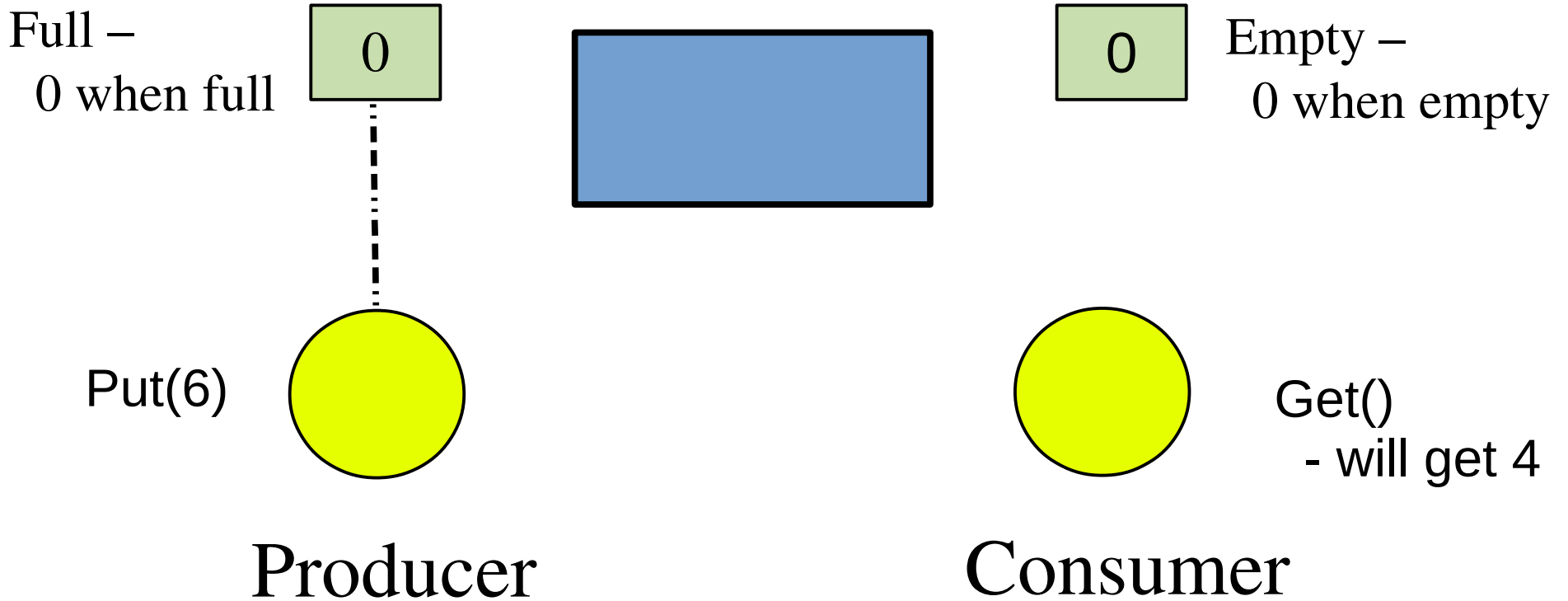


Monitor's buffer now contains 4

The Full semaphore's count is checked but is 0, producer is blocked!

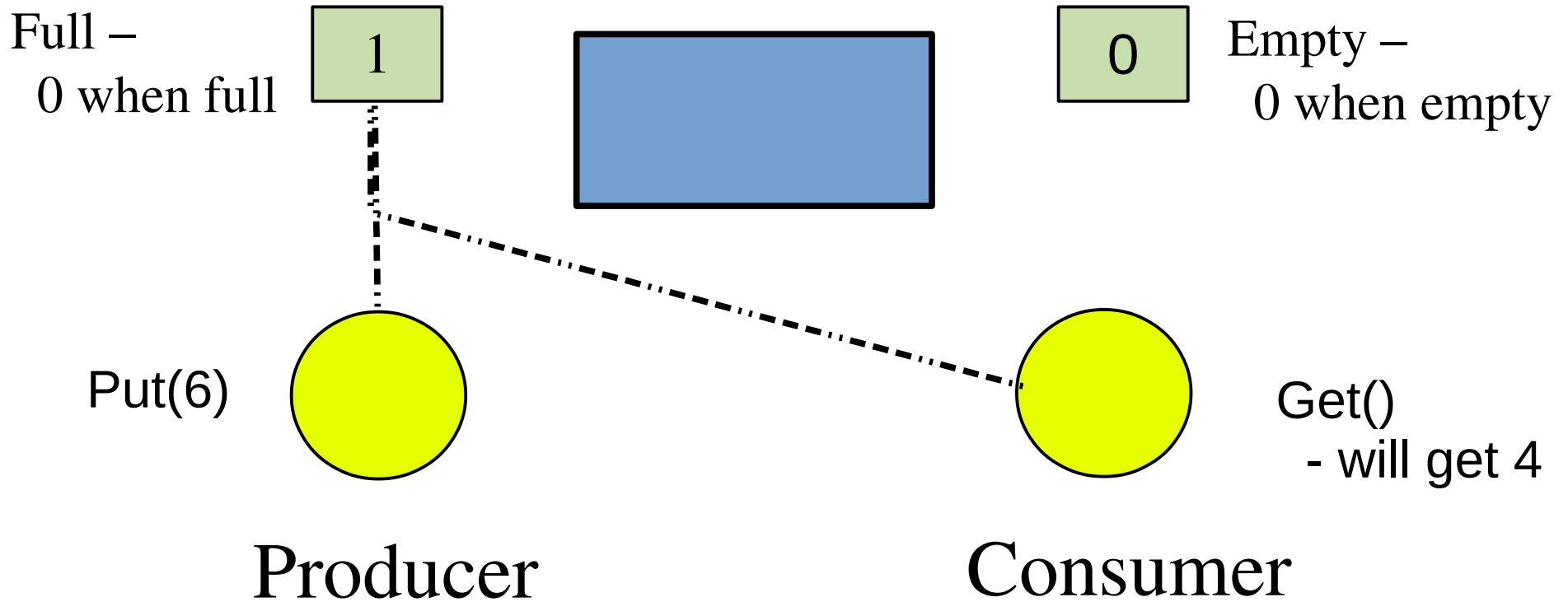
Consumer requests a token from the Monitor

Monitor



Monitor removes 4 from buffer, holds onto it
Monitor now holds no tokens

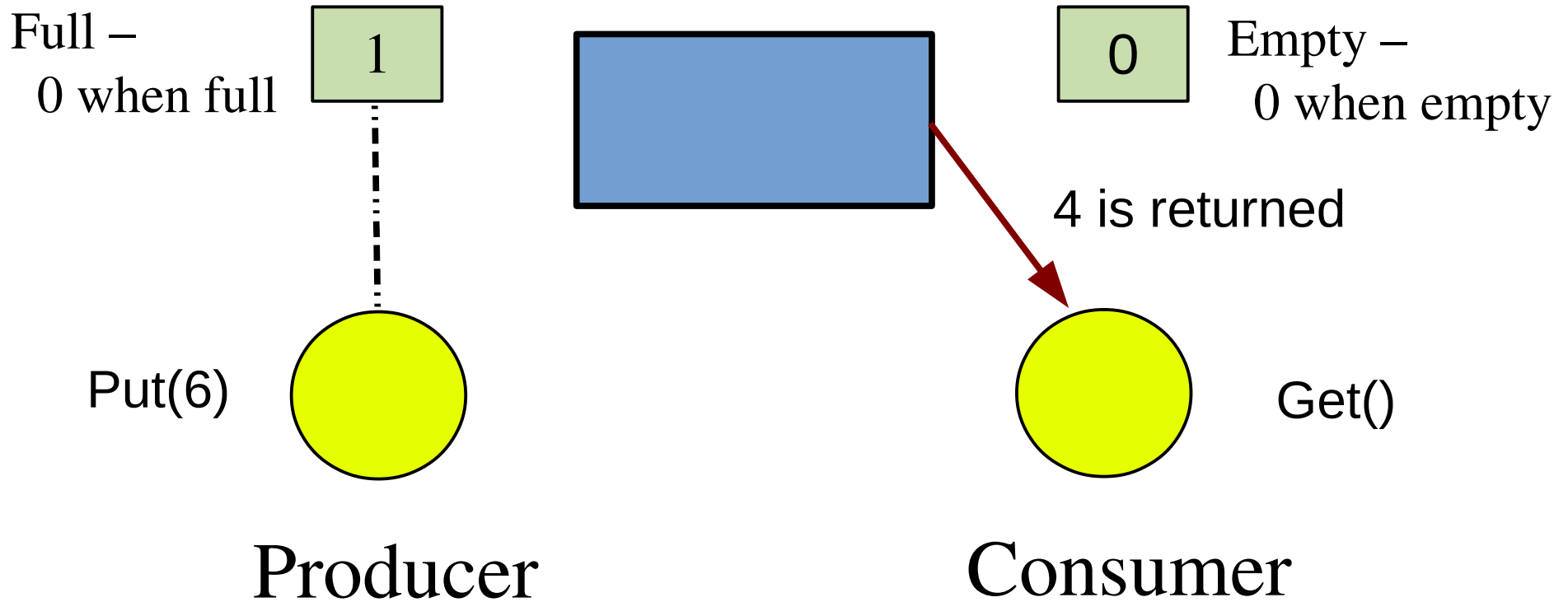
Monitor



Monitor now holds no tokens

The Full semaphore's count is incremented by the consumer

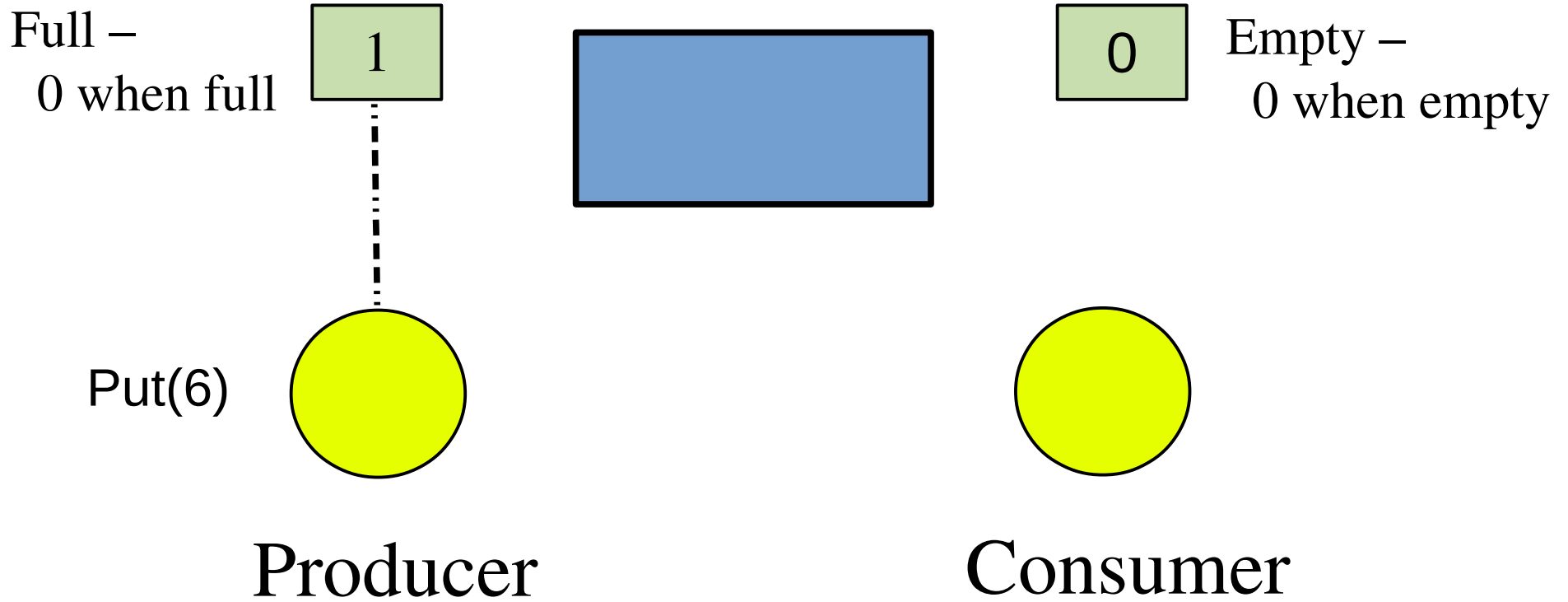
Monitor



Monitor now holds no tokens

Monitor sends 4 to the consumer, consumer continues processing

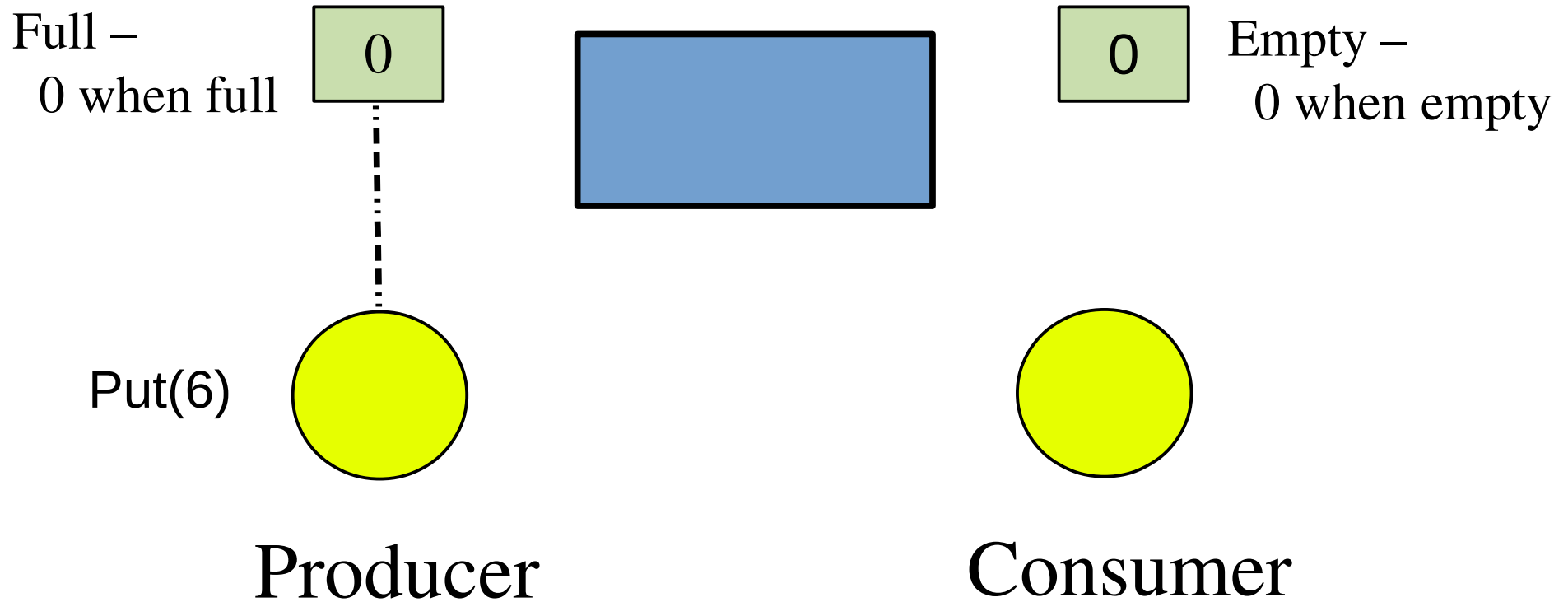
Monitor



Monitor now holds no tokens

Producer is no longer blocked, continues

Monitor



Monitor now holds no tokens

Producer is no longer blocked, continues

The Full semaphore's count is decremented

Monitor

Full –
0 when full

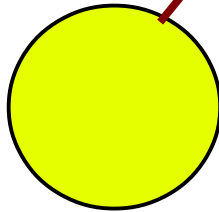
0

6

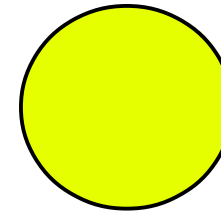
0

Empty –
0 when empty

Put(6)



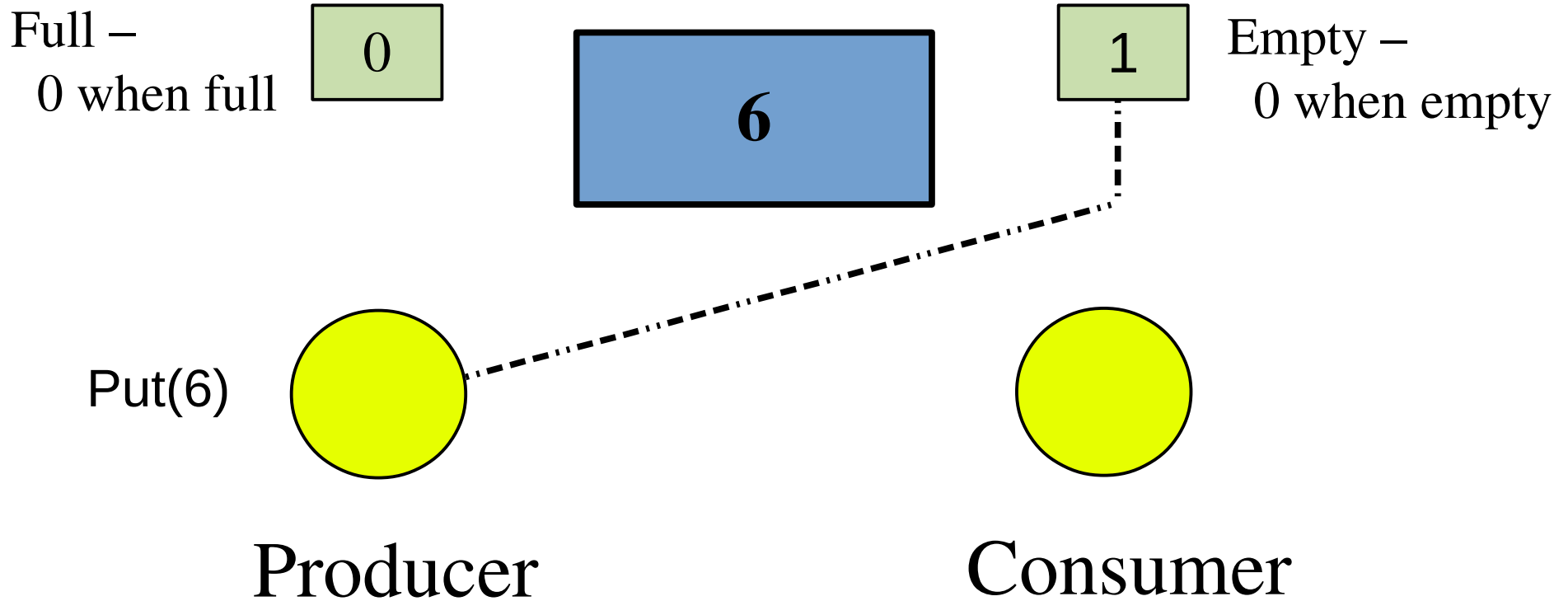
Producer



Consumer

Producer sends 6 to the Monitor's buffer
Monitor's buffer now contains 6

Monitor



Monitor's buffer now contains 6

The Empty semaphore's count is incremented

Monitor

Full –
0 when full

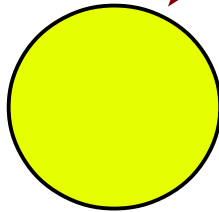
0

6

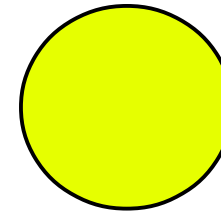
1

Empty –
0 when empty

Put(9)



Producer



Consumer

Monitor's buffer now contains 6
Producer continues processing

Monitor

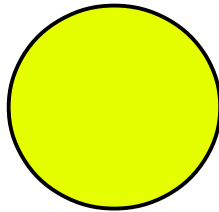
Full –
0 when full

0

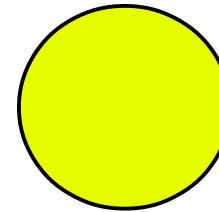
6

1

Empty –
0 when empty



Producer



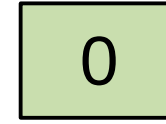
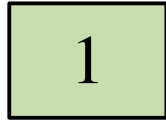
Consumer

Monitor's buffer now contains 6
The Full semaphore's count is 0
The Empty semaphore's count is 1

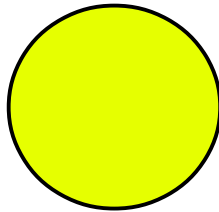
Consumer gets the token from Monitor

Monitor

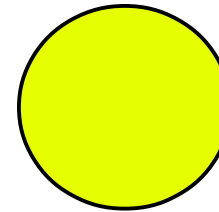
Full –
0 when full



Empty –
0 when empty



Producer

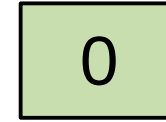
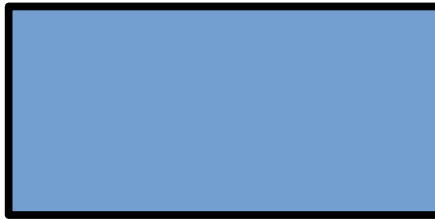
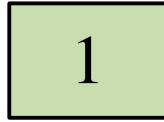


Consumer

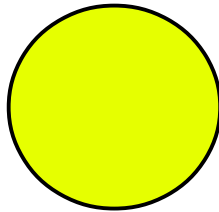
Monitor buffer is empty

Monitor

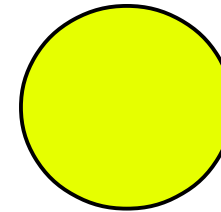
Full –
0 when full



Empty –
0 when empty



Producer

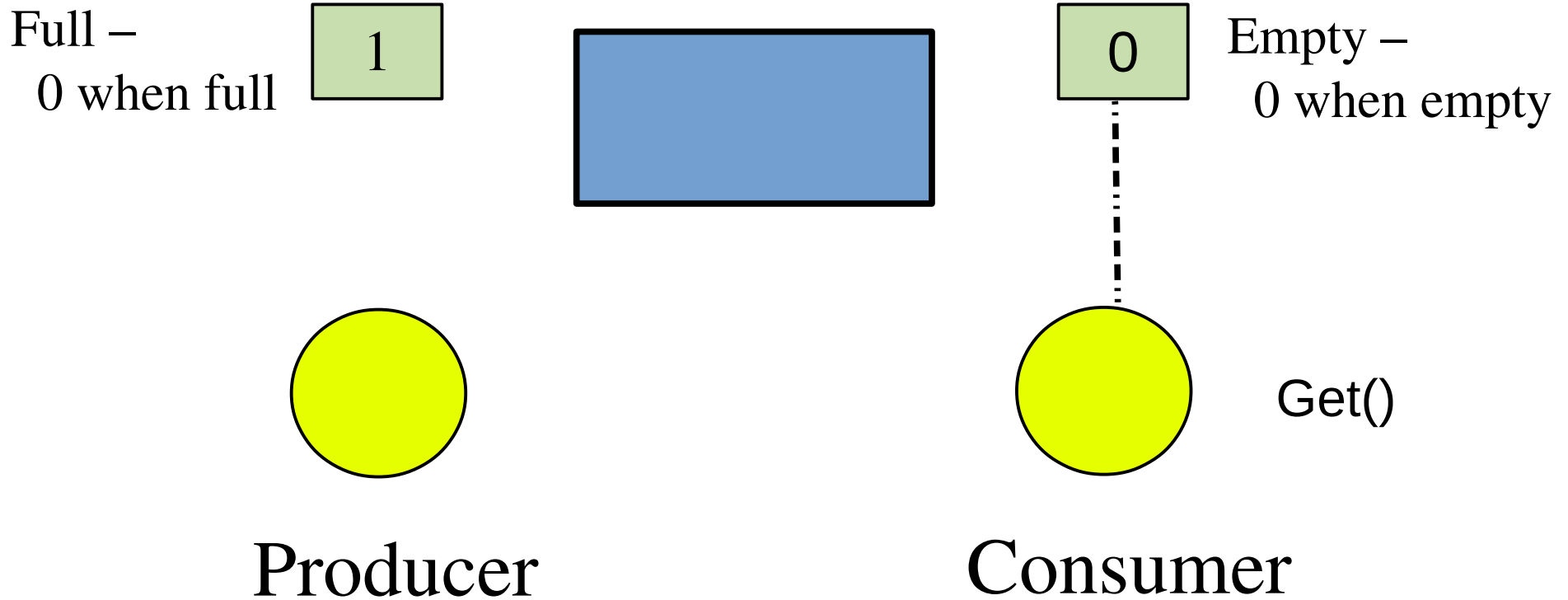


Get()

Consumer

Monitor buffer is empty
The consumer executes a get()

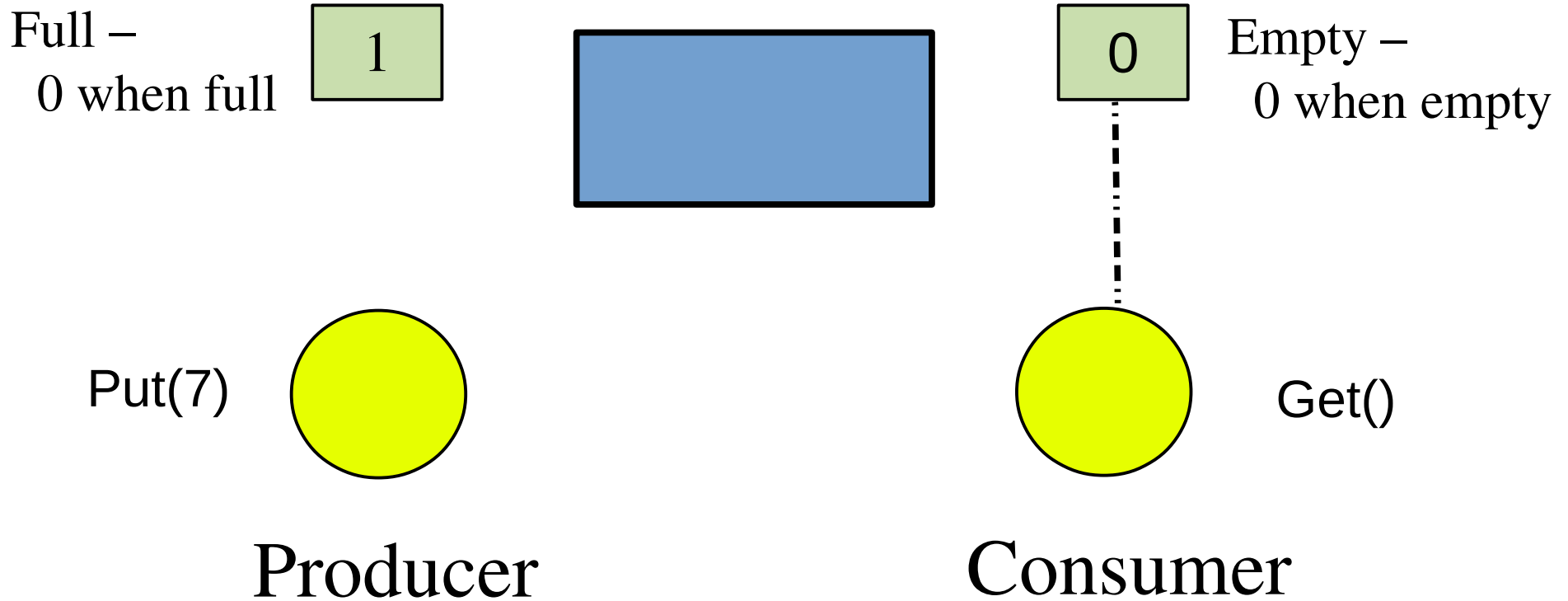
Monitor



Monitor buffer is empty

The Empty semaphore's count is checked but is 0, consumer blocked!

Monitor

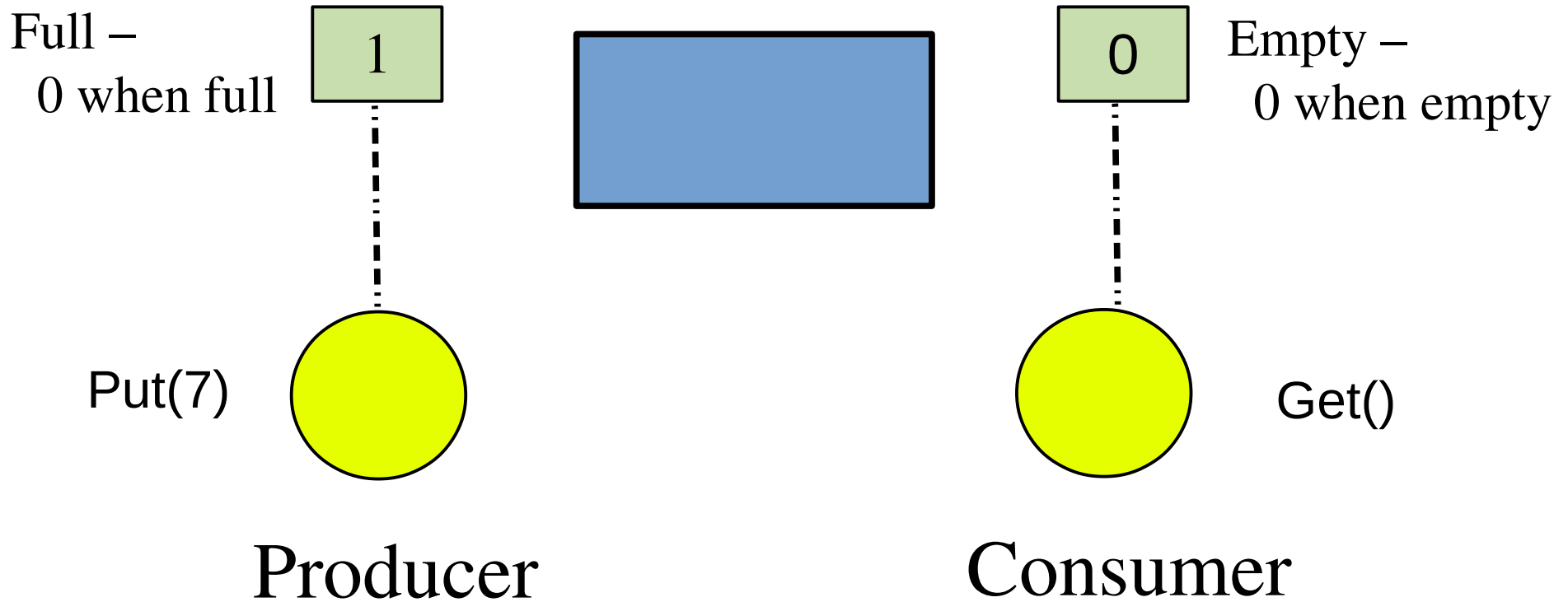


Monitor buffer is empty

The Empty semaphore's count is checked but is 0, consumer blocked!

The producer attempts to put 7 into the monitor's buffer

Monitor

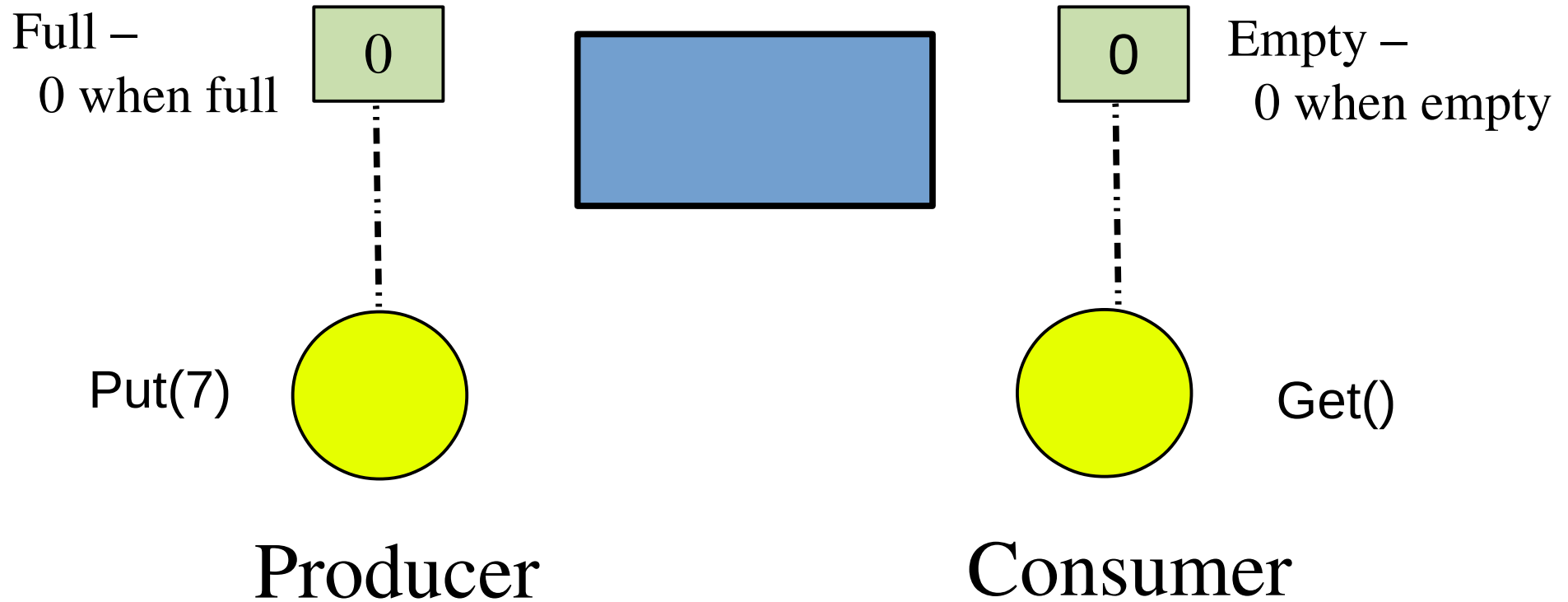


Monitor buffer is empty

The Empty semaphore's count is checked but is 0, consumer blocked!

The Full semaphore's count is checked to be > 0

Monitor



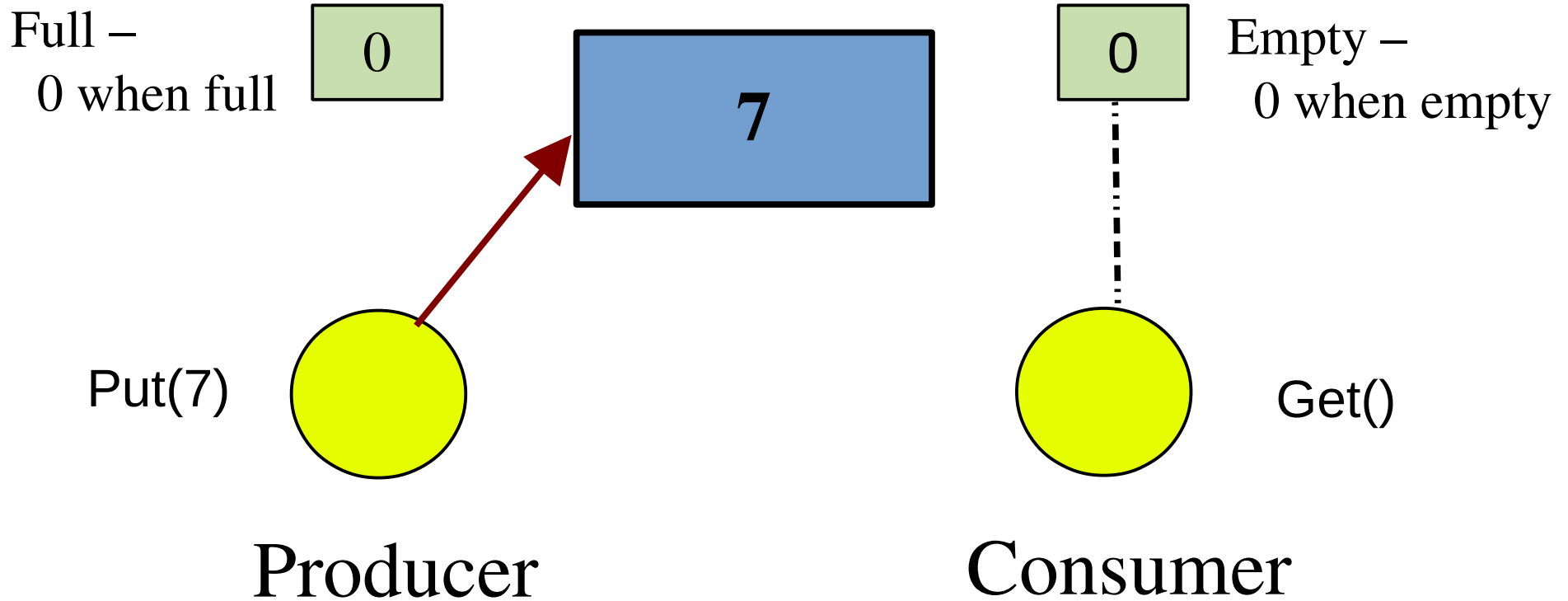
Monitor buffer is empty

The Empty semaphore's count is checked but is 0, consumer blocked!

The Full semaphore's count is checked to be > 0

It is! So producer passes through and decrements Full's count

Monitor

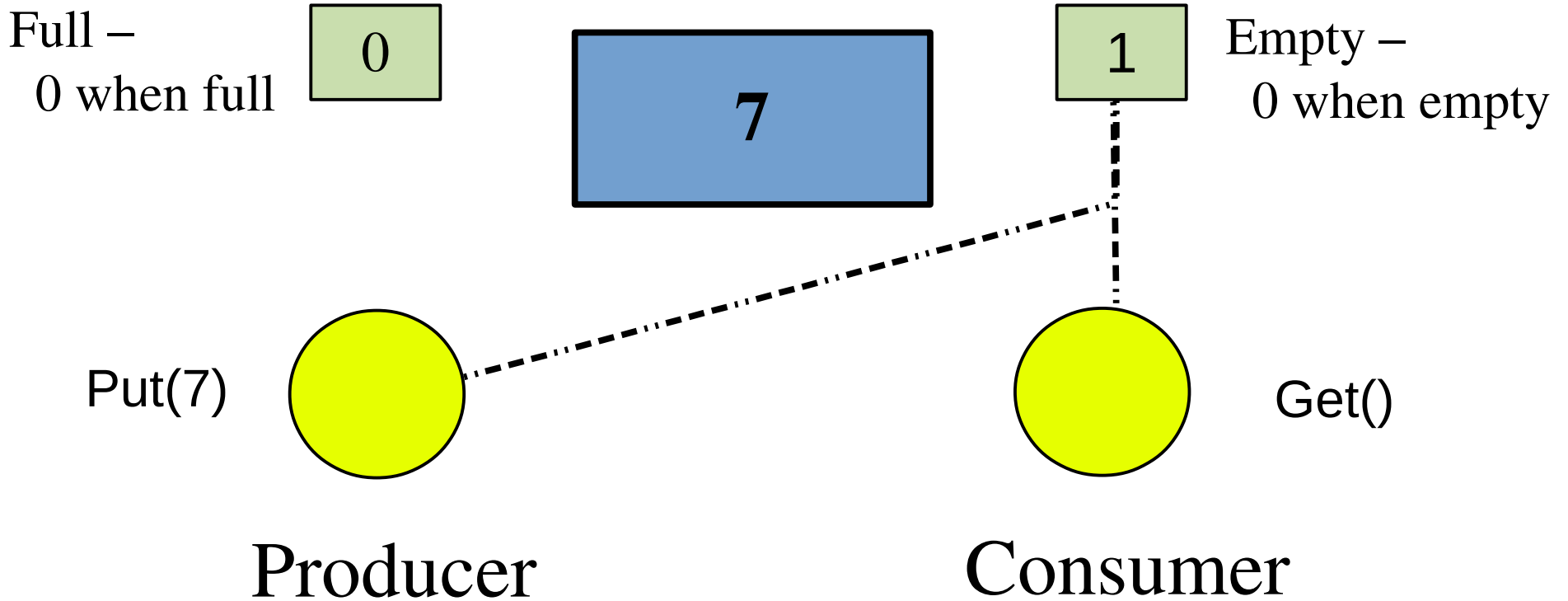


Producer sends 7 to the Monitor's buffer

The Empty semaphore's count is checked but is 0, consumer blocked!

Monitor's buffer now contains 7

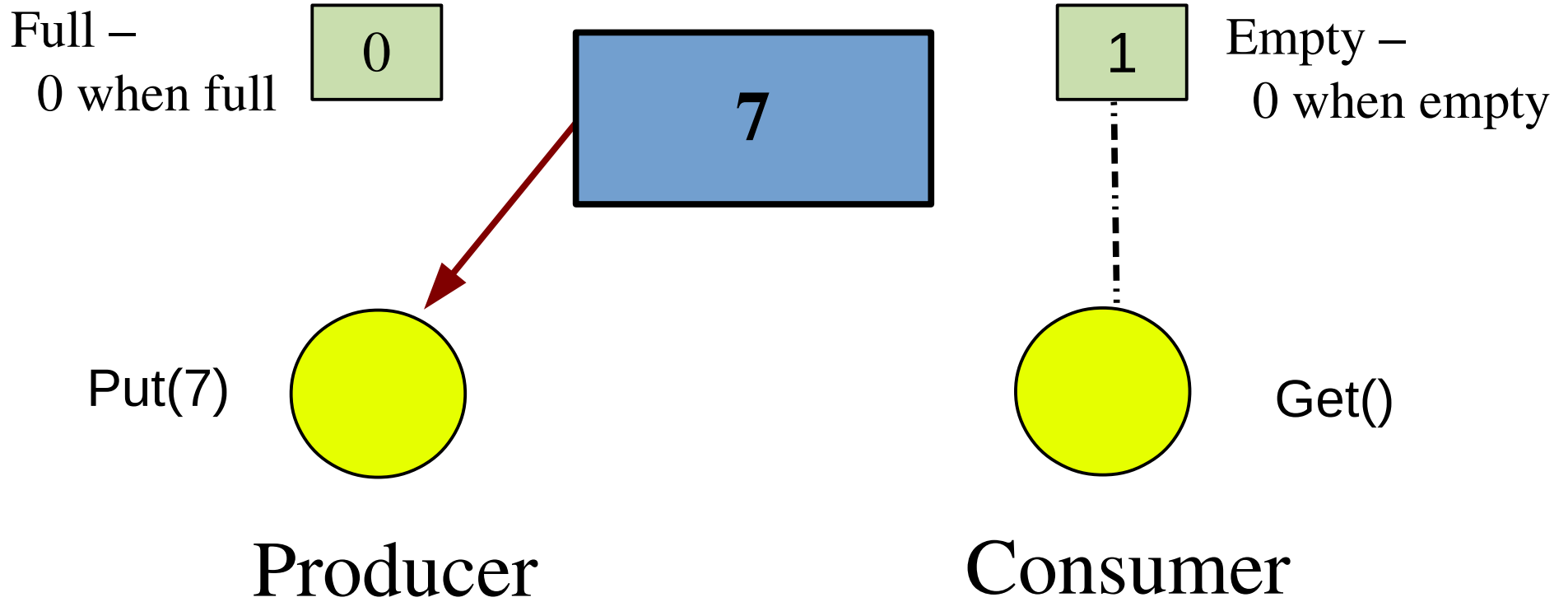
Monitor



Monitor's buffer now contains 7

The Empty semaphore's count is incremented

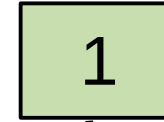
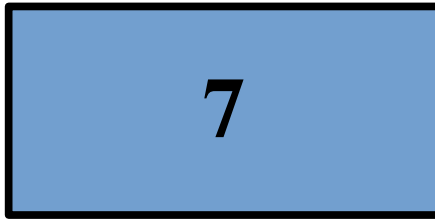
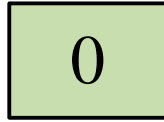
Monitor



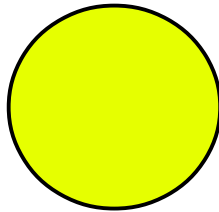
Monitor's buffer now contains 7
Producer returns from put

Monitor

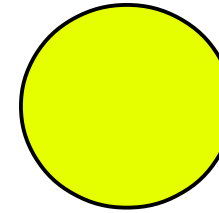
Full –
0 when full



Empty –
0 when empty



Producer

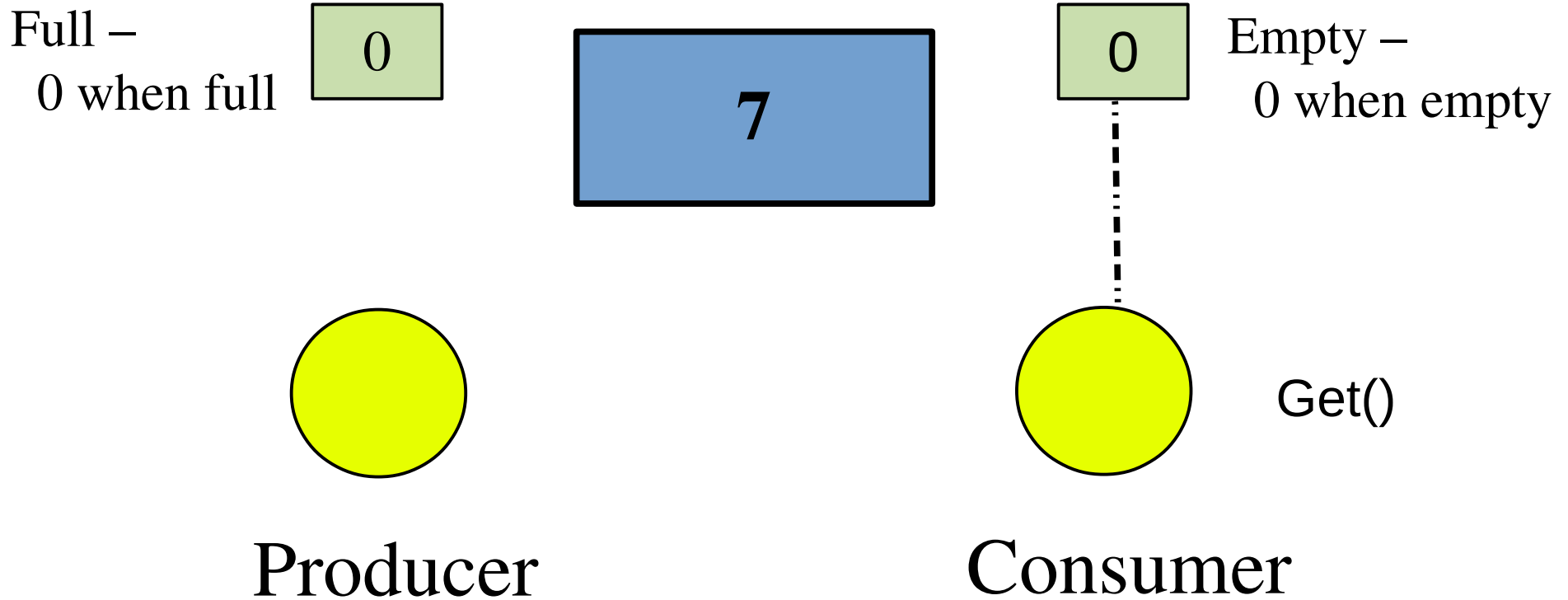


Consumer

Get()

Monitor's buffer now contains 7

Monitor



Monitor's buffer now contains 7

Consumer awakens, decrements the Empty semaphore's count

Monitor

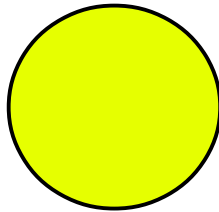
Full –
0 when full

0

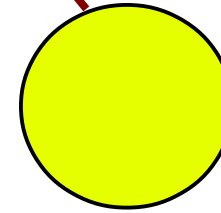
7

0

Empty –
0 when empty



Producer



Get()

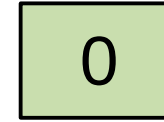
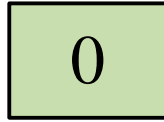
Consumer

Monitor's buffer now contains 7

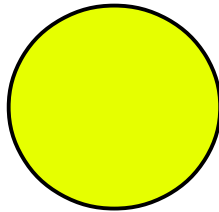
Consumer requests a token from the Monitor

Monitor

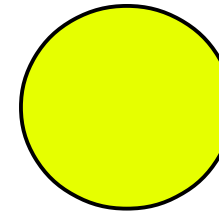
Full –
0 when full



Empty –
0 when empty



Producer



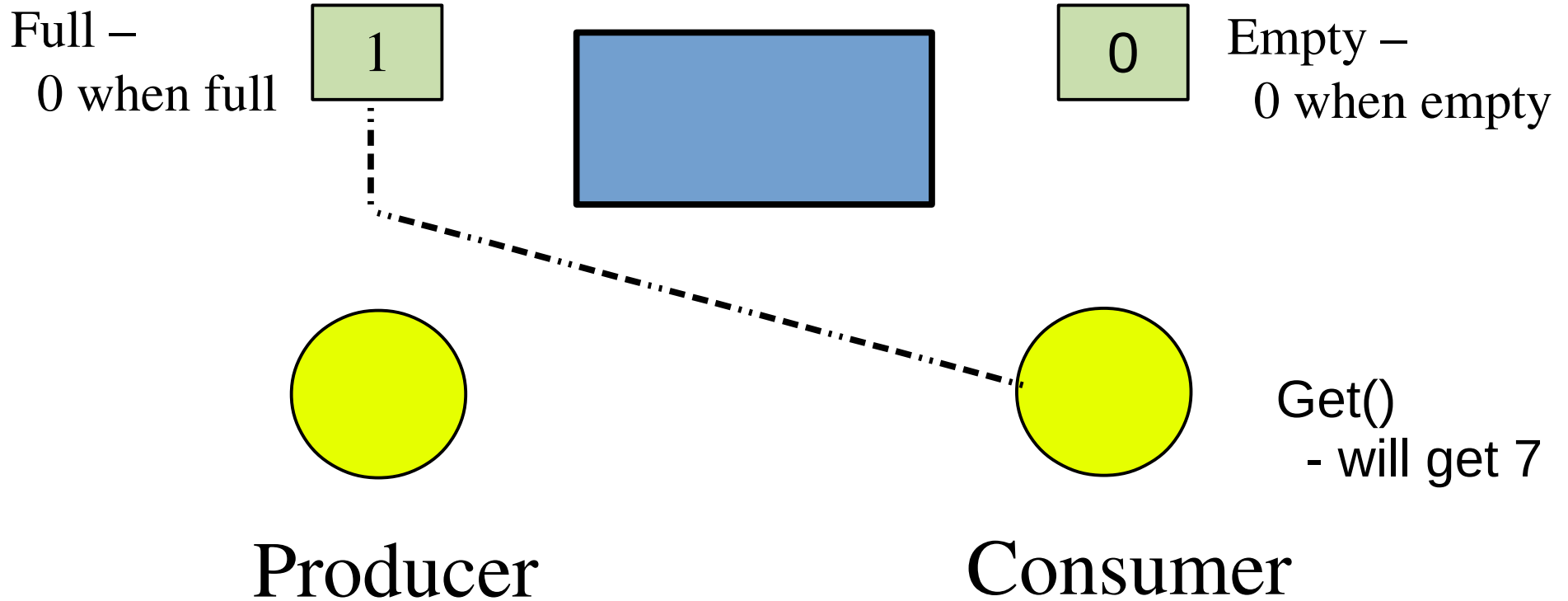
Get()
- will get 7

Consumer

Monitor is empty

Monitor removes 7 from buffer, holds onto it

Monitor



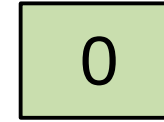
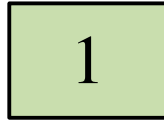
Monitor is empty

Monitor removes 7 from buffer, holds onto it

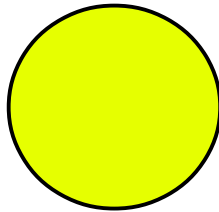
The Full semaphore's count is incremented

Monitor

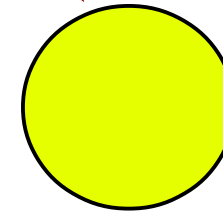
Full –
0 when full



Empty –
0 when empty



Producer



Consumer

7 is returned

Get()

Monitor is empty

Monitor sends 7 to the consumer