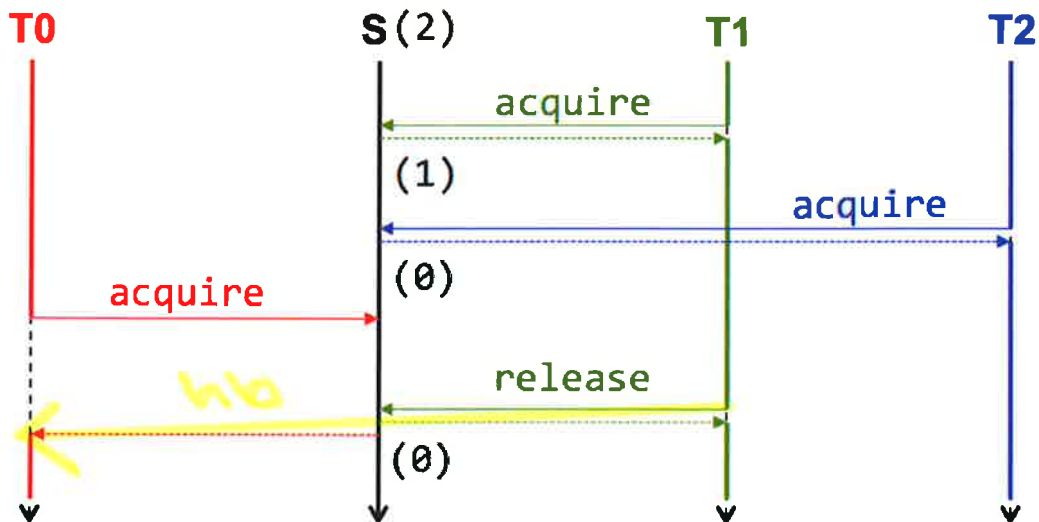


Visibilitätsgarantien der *java.util.concurrent* Synchronizer

Zeichnen Sie die happens-before Relation in die gegebenen Ablaufsequenzen ein.

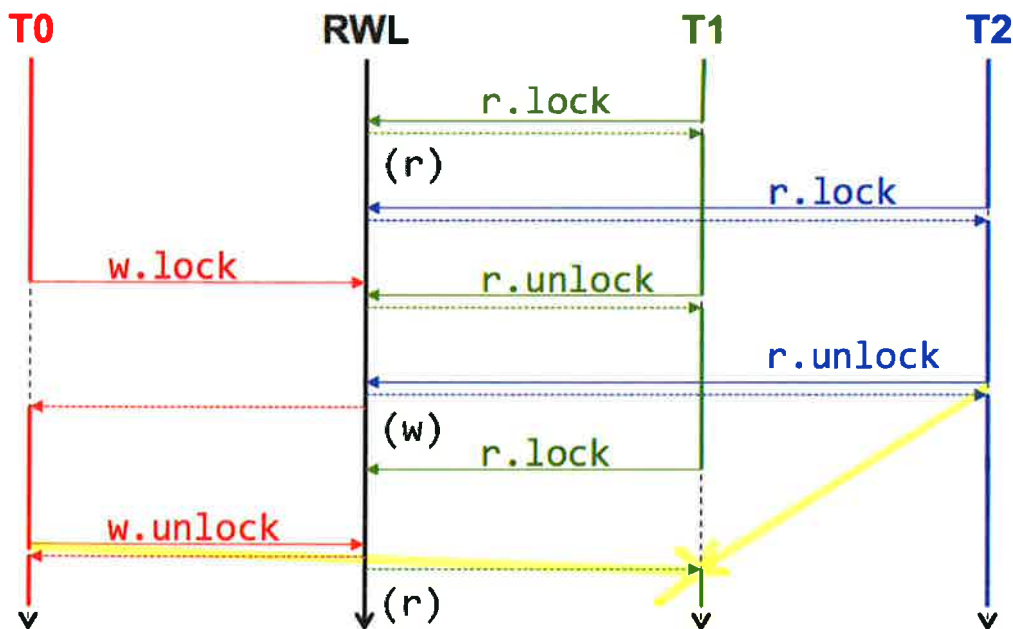
Semaphore:



JMM:

Actions in a thread prior to calling `release()` happen-before actions following a successful call to `acquire()` in another thread.

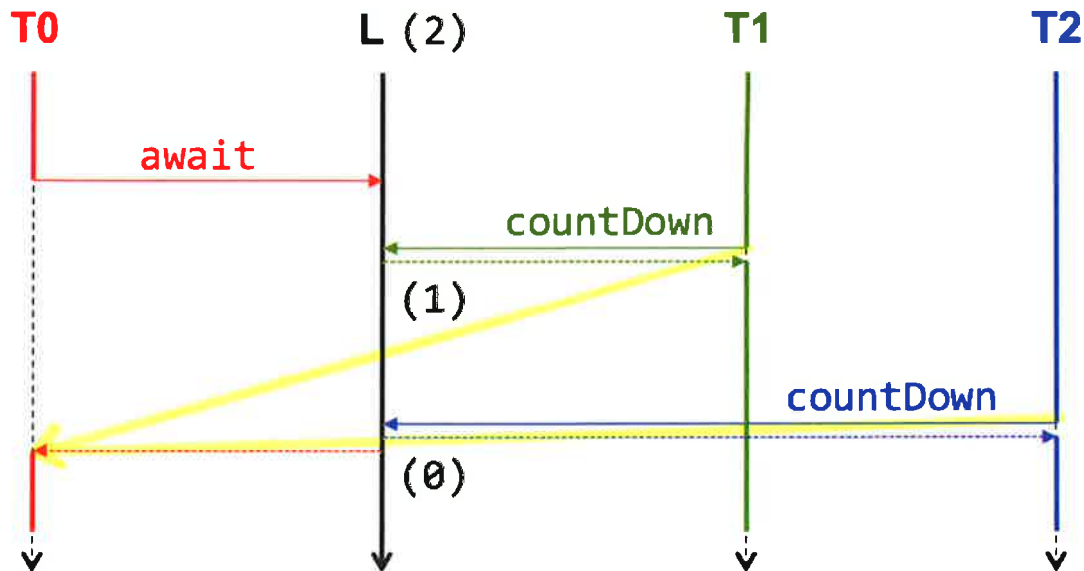
ReadWriteLock:



JMM:

The read-Lock and the write-Lock both individually conform to the specification of the Lock interface. Additionally: An unlock on the write-Lock happens-before the lock on the read-Lock.

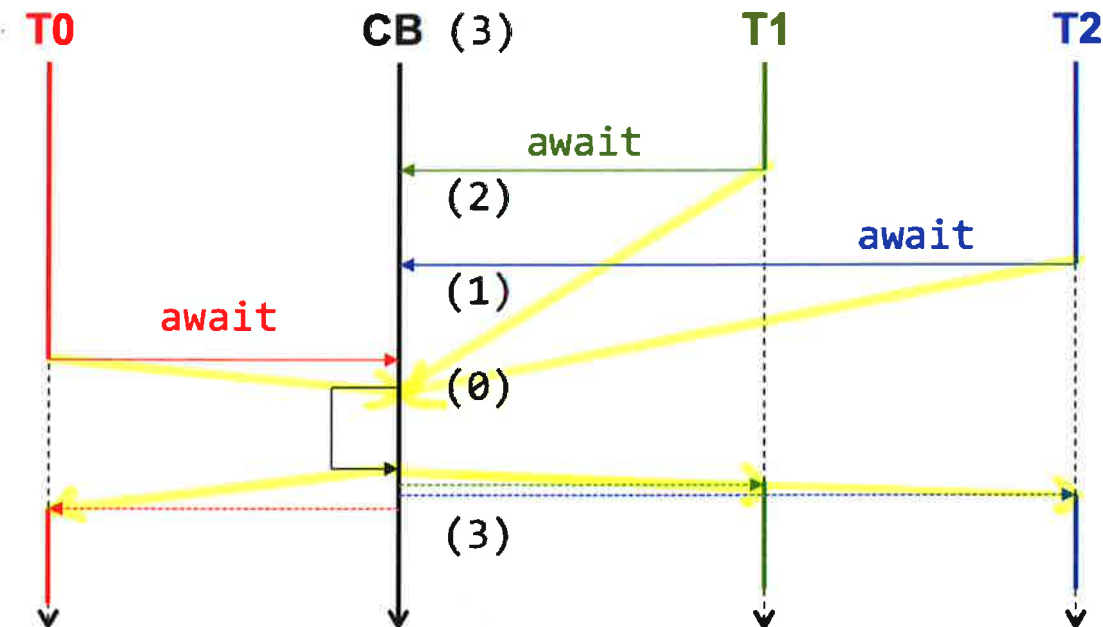
CountDownLatch:



JMM:

Actions in a thread prior to calling `countDown()` happen-before actions following a successful return from a corresponding `await()` in another thread on the same latch instance.

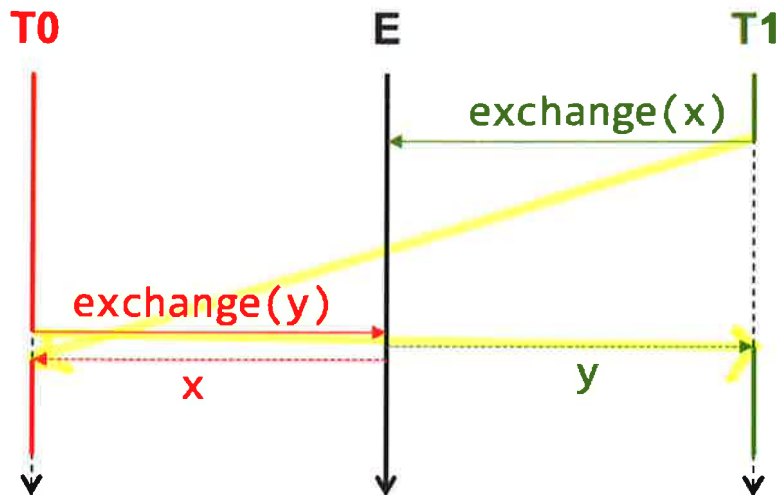
CyclicBarrier:



JMM:

Actions in a thread prior to calling `await()` happen-before actions that are part of the barrier action, which in turn happen-before actions following a successful return from the corresponding `await()` in other threads on the same barrier instance.

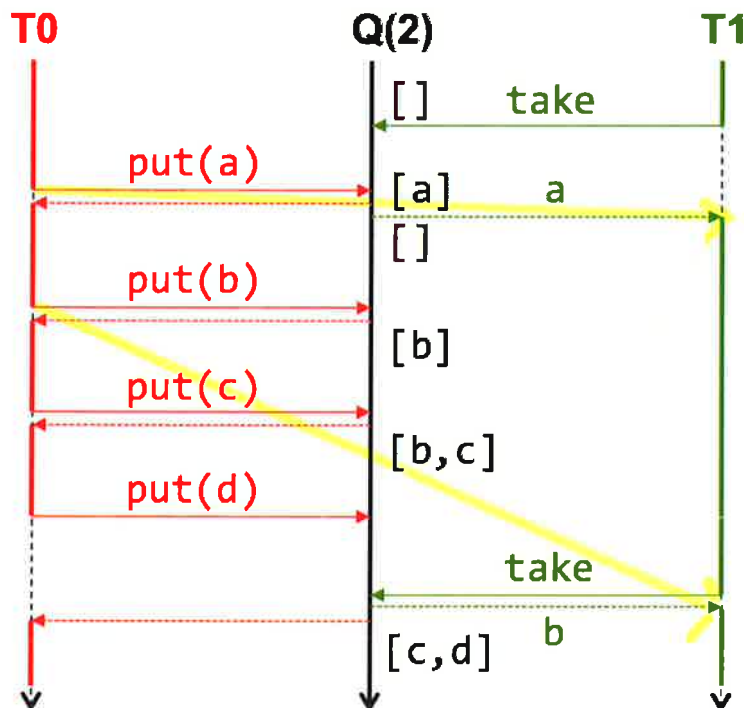
Exchanger:



JMM:

For each pair of threads that successfully exchange objects via an Exchanger, actions prior to the `exchange()` in each thread **happen-before** those subsequent to a **return from the corresponding `exchange()`** in the other thread on the same exchanger instance.

BlockingQueue:



JMM:

As with other concurrent collections, actions in a thread prior to placing an object into a BlockingQueue **happen-before** actions subsequent to the **access or removal of that element** from the BlockingQueue in another thread on the same queue instance.

