28.7.4 Sequentially Consistent Atomics

For a candidate execution *execution*, memory-order is a strict total order of all events in EventSet(*execution*) that satisfies the following.

- For each pair (*E*, *D*) in *execution*.[[HappensBefore]], (*E*, *D*) is in memory-order.
- For each pair (*R*, *W*) in *execution*.[[ReadsFrom]], there is no WriteSharedMemory or ReadModifyWriteSharedMemory event *V* in SharedDataBlockEventSet(*execution*) such that *V*.[[Order]] is SeqCst, the pairs (*W*, *V*) and (*V*, *R*) are in memory-order, and any of the following conditions are true.
  - The pair (*W*, *R*) is in *execution*.[[SynchronizesWith]], and *V* and *R* have equal ranges.
  - The pairs (*W*, *R*) and (*V*, *R*) are in *execution*.[[HappensBefore]], *W*.[[Order]] is SeqCst, and *W* and *V* have equal ranges.
  - The pairs (*W*, *R*) and (*W*, *V*) are in *execution*.[[HappensBefore]], *R*.[[Order]] is SeqCst, and *V* and *R* have equal ranges.

  > NOTE 1 This clause additionally constrains SeqCst events on equal ranges.

- For each WriteSharedMemory or ReadModifyWriteSharedMemory event *W* in SharedDataBlockEventSet(*execution*), if *W*.[[Order]] is SeqCst, then it is not the case that there is an infinite number of ReadSharedMemory or ReadModifyWriteSharedMemory events in SharedDataBlockEventSet(*execution*) with equal range that is memory-order before *W*.

  > NOTE 2 This clause together with the forward progress guarantee on agents ensure the liveness condition that SeqCst writes become visible to SeqCst reads with equal range in finite time.

A candidate execution has sequentially consistent atomics if a memory-order exists.