Linearizability
- A Quick Overview

Ashvin Goel

Electrical and Computer Engineering
University of Toronto

ECE1724

These slides are modified versions of slides from Michael Freedman & Wyatt Lloyd's course on Distributed Systems

Data Consistency Models

- Contract (or a set of guarantees) that a system provides to applications about expected behavior when data is accessed (read, written, updated, etc.)

Linearizability

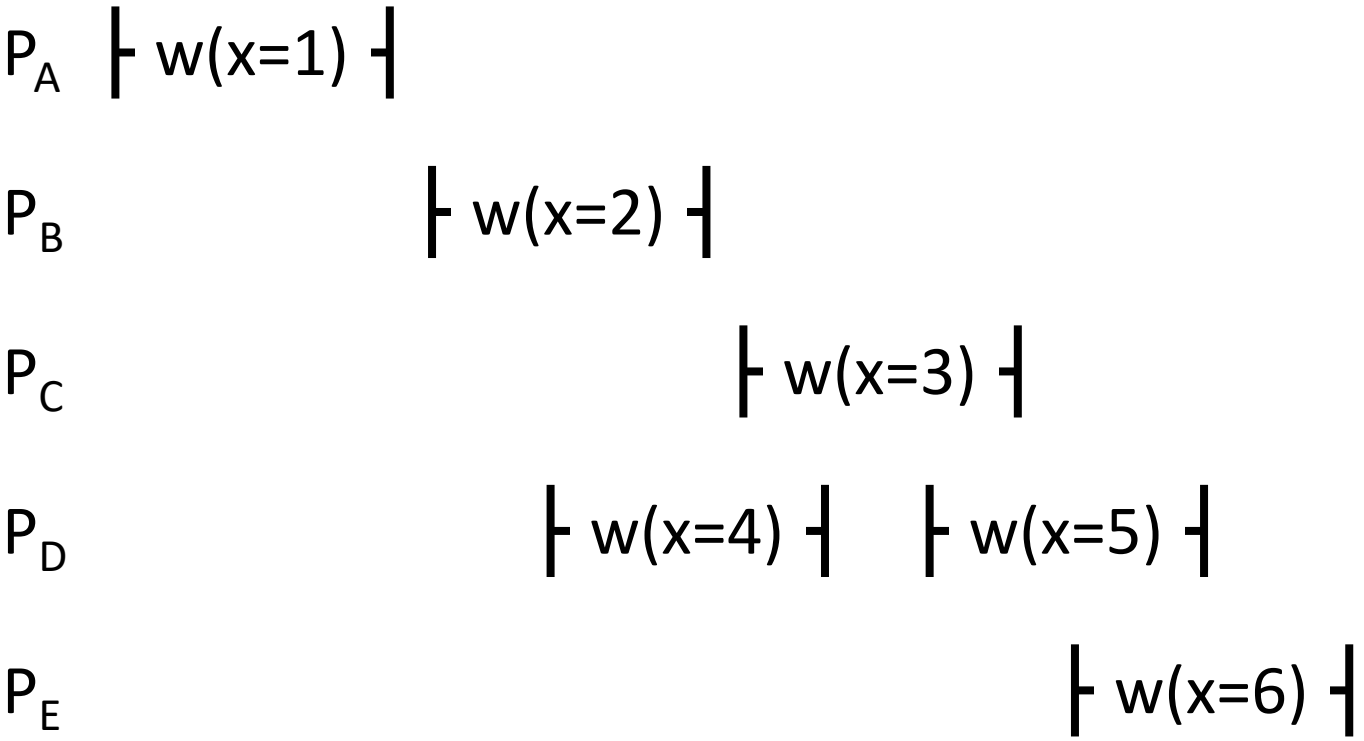
- Assumption: Each operation accesses one data item

- All operations are performed in some total order
- The total order preserves the real-time ordering between operations
  - If operation A completes before operation B begins in real-time, then A is ordered before B
  - If neither A nor B completes before the other begins, then there is no real-time order, but there must be some total order

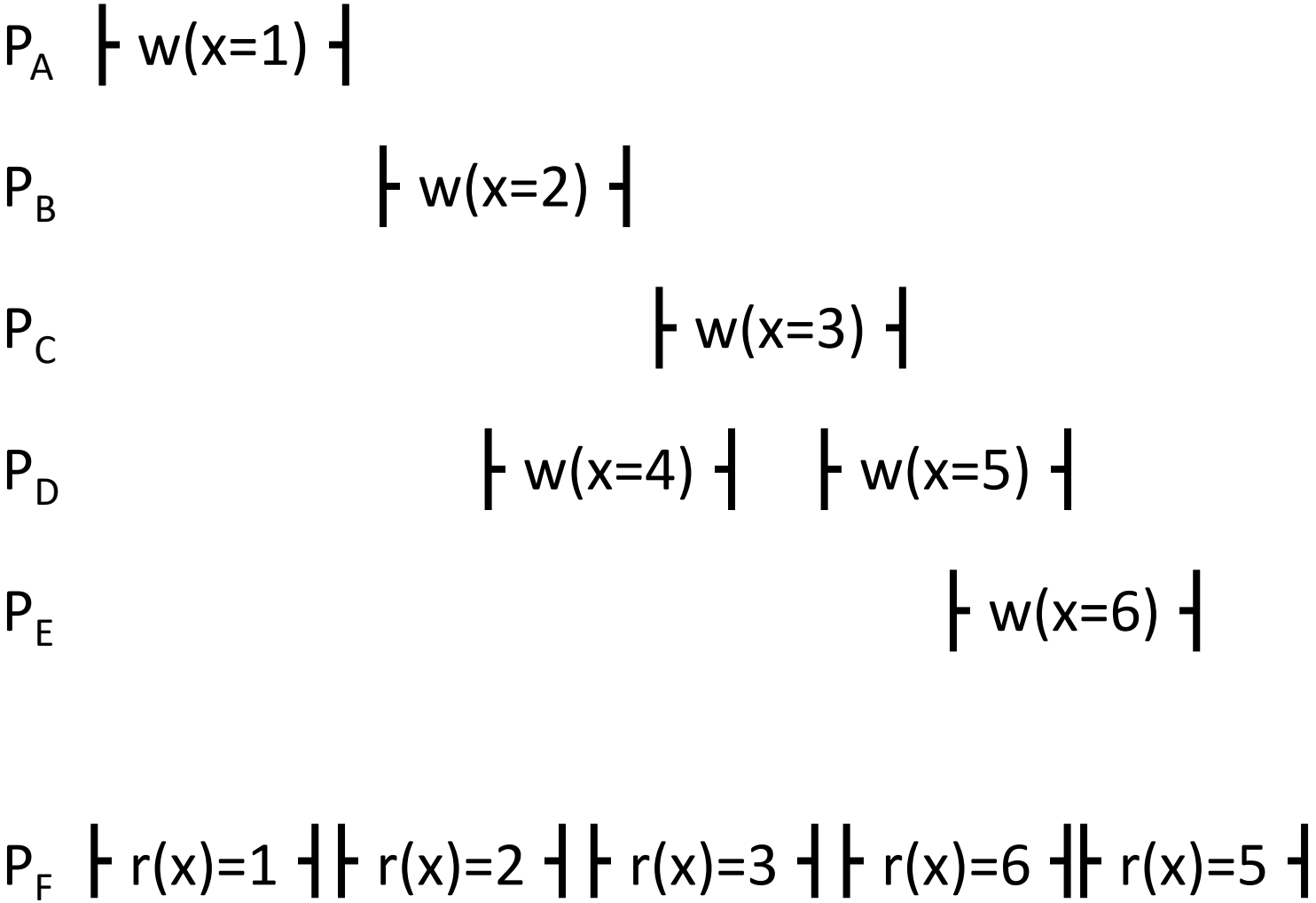
Understanding Linearizability

- Writes are ordered
  - Writes appear to occur instantaneously
- Reads read latest data
  - After a write completes, a later read (in real-time order) returns the value of the write (or later write)
  - Once a read returns a value, all later reads return that value or the value of a later write

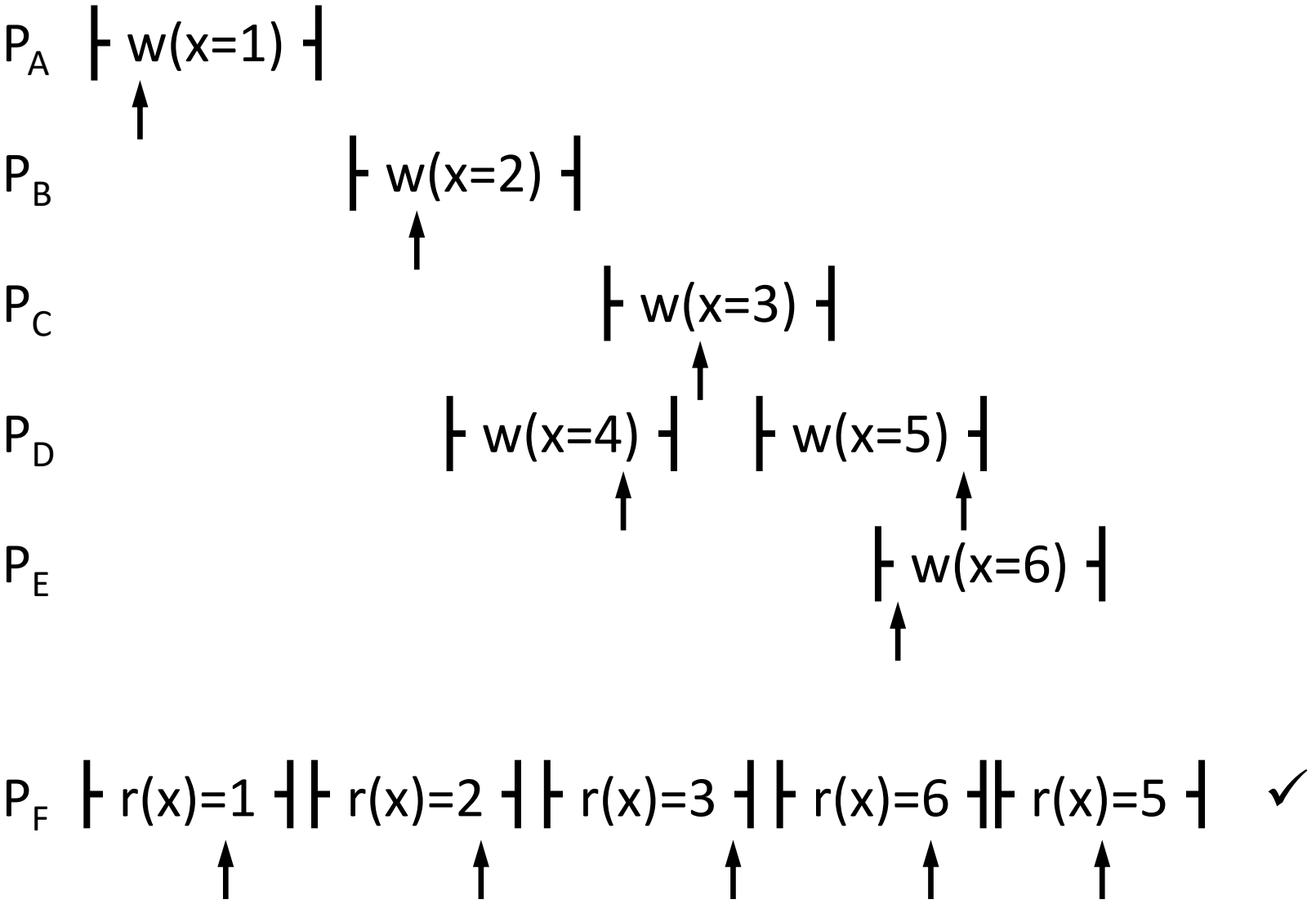
Real-Time Ordering Examples



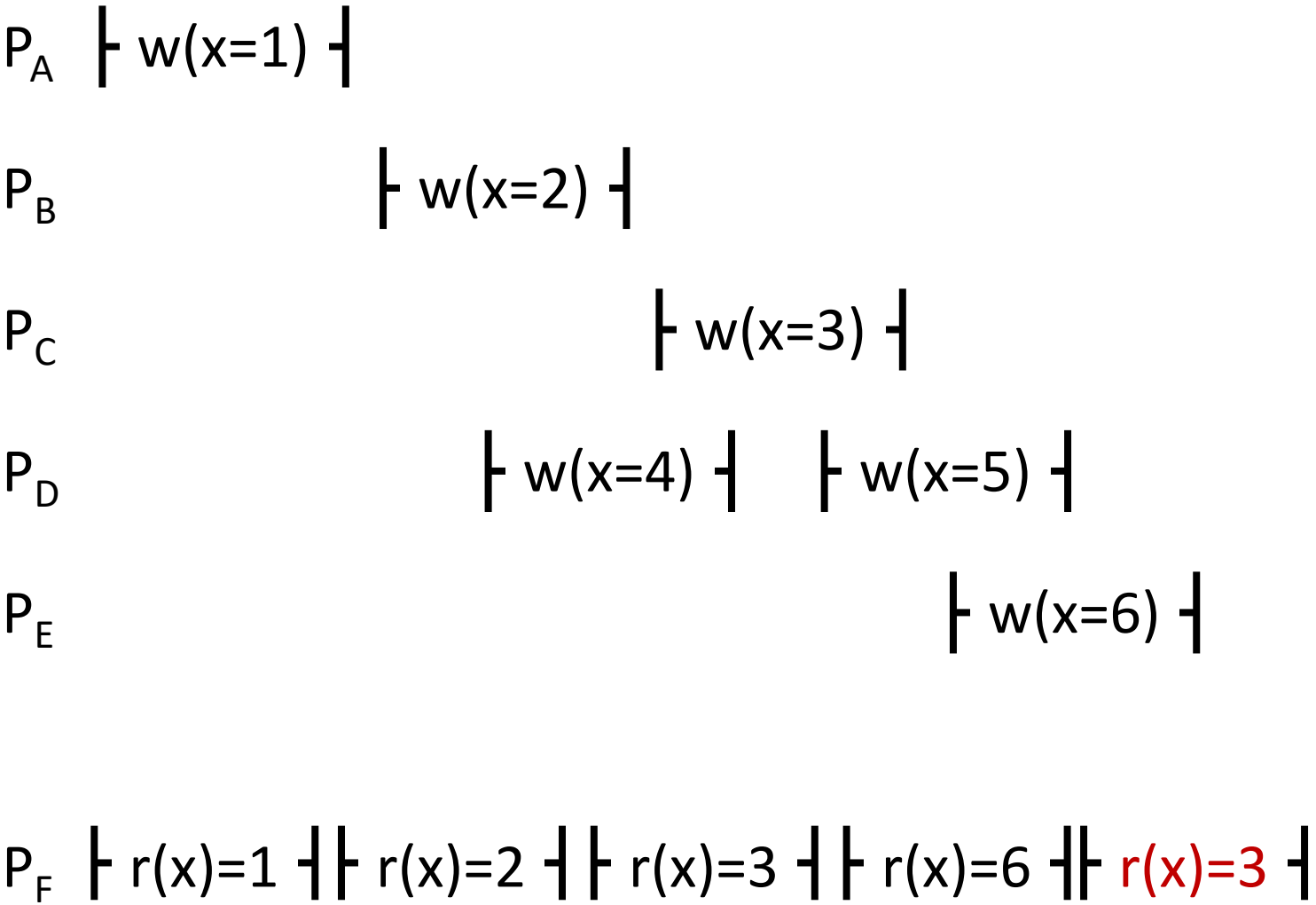
Linearizable?



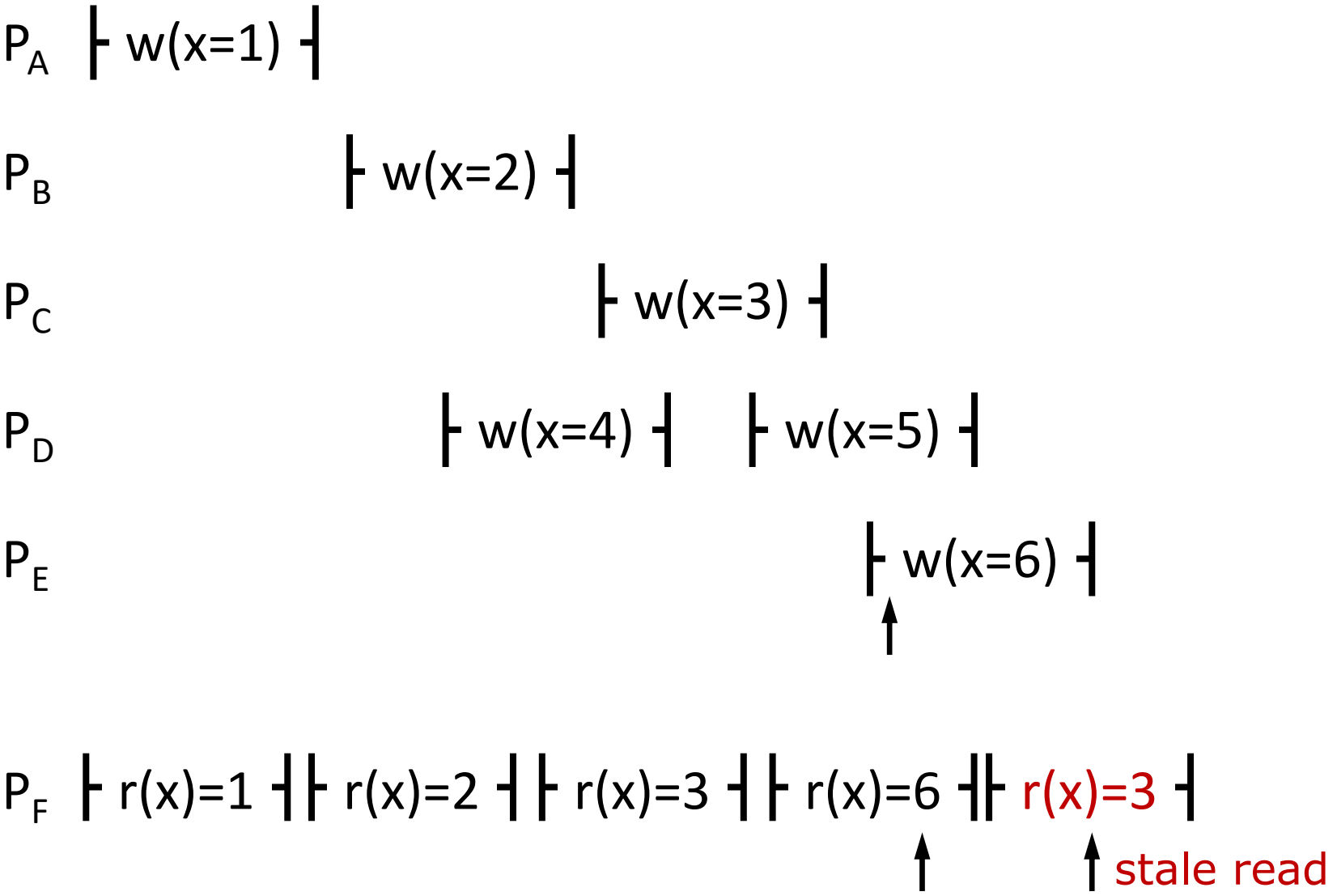
Linearizable: Yes



Linearizable?



Linearizable: No



Why Linearizability?

- Behavior is like single machine processing one request at a time
  - Hides the complexity of distributed and replicated systems from applications
  - Hides complexity associated with failures
  - Easier to write correct applications
- Atomic broadcast (Zab protocol used by ZooKeeper), RAFT, PAXOS, etc., provide linearizability for replicated data stores
- However, linearizability is a strong consistency guarantee that can limit performance
  - We will discuss this issue today