



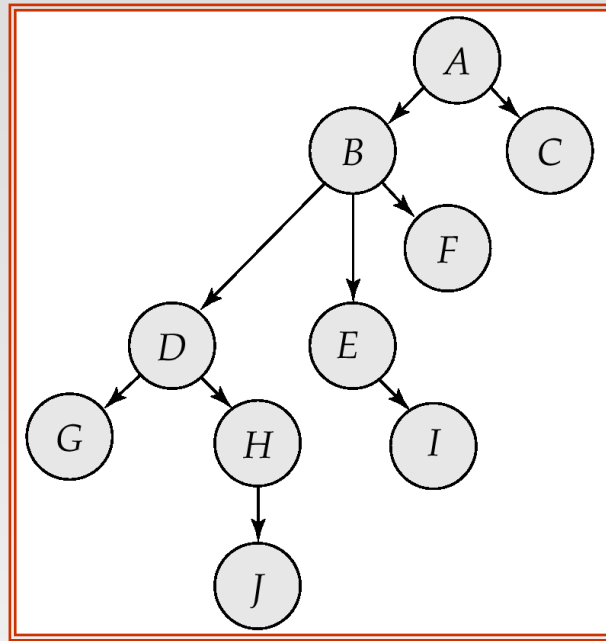
Graph-Based Protocols

- Graph-based protocols are an alternative to two-phase locking
- Impose a partial ordering \rightarrow on the set $\mathbf{D} = \{d_1, d_2, \dots, d_h\}$ of all data items.
 - If $d_i \rightarrow d_j$ then any transaction accessing both d_i and d_j must access d_i before accessing d_j .
 - Implies that the set \mathbf{D} may now be viewed as a directed acyclic graph, called a *database graph*.
- The *tree-protocol* is a simple kind of graph protocol.





Tree Protocol



1. Only exclusive locks are allowed.
2. The first lock by T_i may be on any data item. Subsequently, a data Q can be locked by T_i only if the parent of Q is currently locked by T_i .
3. Data items may be unlocked at any time.
4. A data item that has been locked and unlocked by T_i cannot subsequently be relocked by T_i .





Graph-Based Protocols (Cont.)

- The tree protocol ensures conflict serializability as well as freedom from deadlock.
- Unlocking may occur earlier in the tree-locking protocol than in the two-phase locking protocol.
 - shorter waiting times, and increase in concurrency
 - protocol is deadlock-free, no rollbacks are required
- Drawbacks
 - Protocol does not guarantee recoverability or cascade freedom
 - ▶ Need to introduce commit dependencies to ensure recoverability
 - Transactions may have to lock data items that they do not access.
 - ▶ increased locking overhead, and additional waiting time
 - ▶ potential decrease in concurrency
- Schedules not possible under two-phase locking are possible under tree protocol, and vice versa.

