CONCURRENT B-TREE ALGORITHM

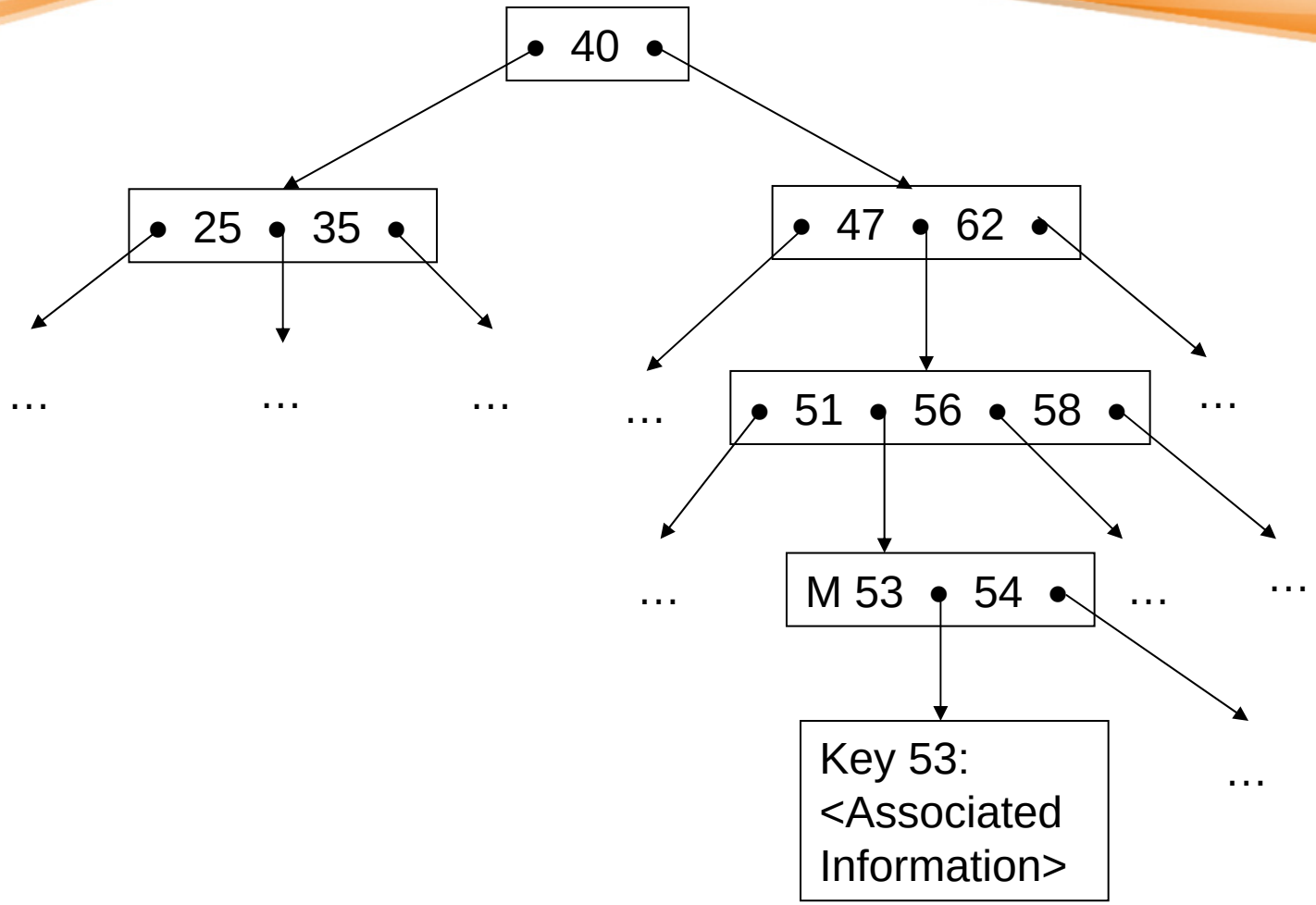
COSC6490A

Liping Han

Motivation

- B-tree is widely used for storing large files, especially on secondary storage devices
- Concurrent interest in database design

B-Tree Data Structure

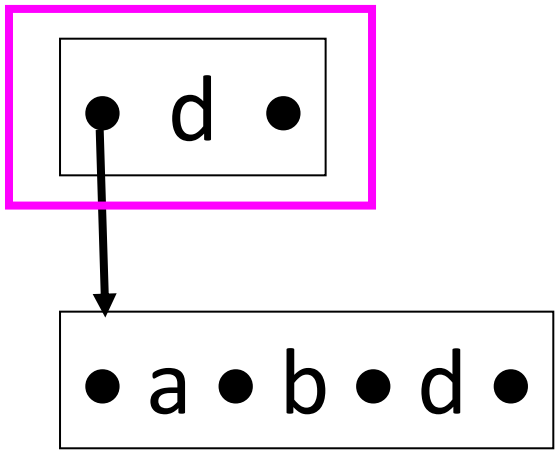


Sequential algorithm

- A descent through the tree to a leaf node
- An operation on the leaf node
  - Search: check for the existence of a key
  - Insert: add a key
  - Delete: remove a key
- An optional ascent to restructure the tree
  - Overflow
  - Underflow

Previous Work

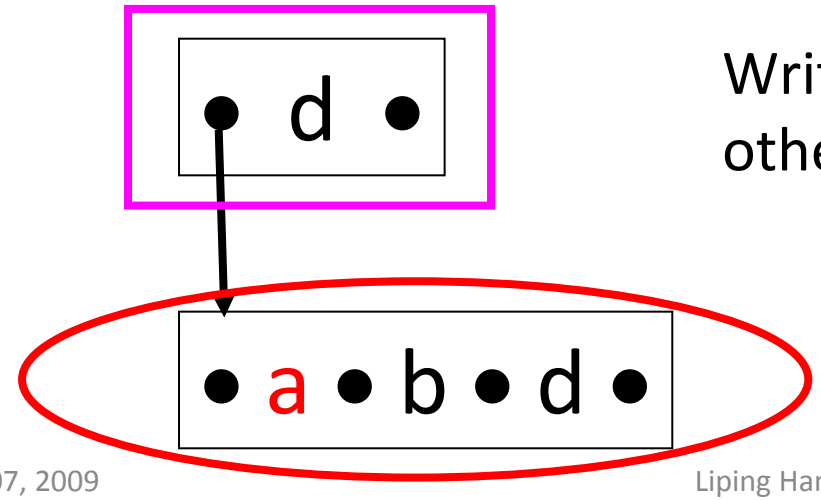
- Exclusively lock entire subtree of the highest affected node (use semaphore)
- Lock-coupling : acquire lock on child before releasing lock on parent (use write-exclusive lock, exclusive lock)



Writer-exclusion lock: lock out other writers

Previous Work

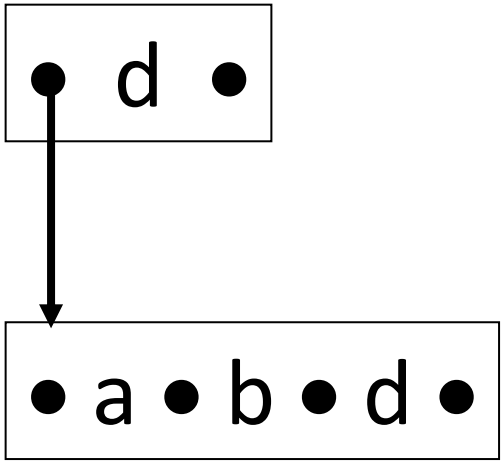
- Exclusively lock entire subtree of the highest affected node (use semaphore)
- Lock-coupling : acquire lock on child before releasing lock on parent (use write-exclusive lock, exclusive lock)



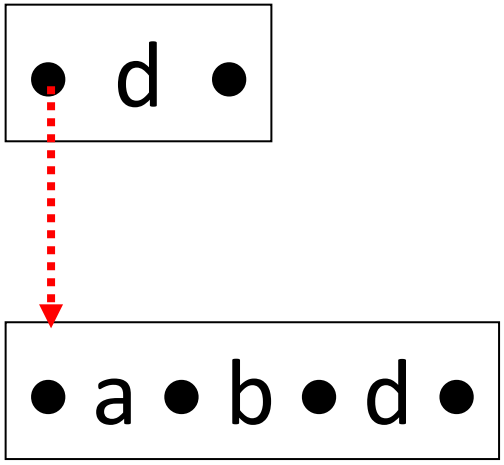
Writer-exclusion lock: lock out other writers

Exclusive lock: lock out all other readers and writers

Problem without lock-coupling

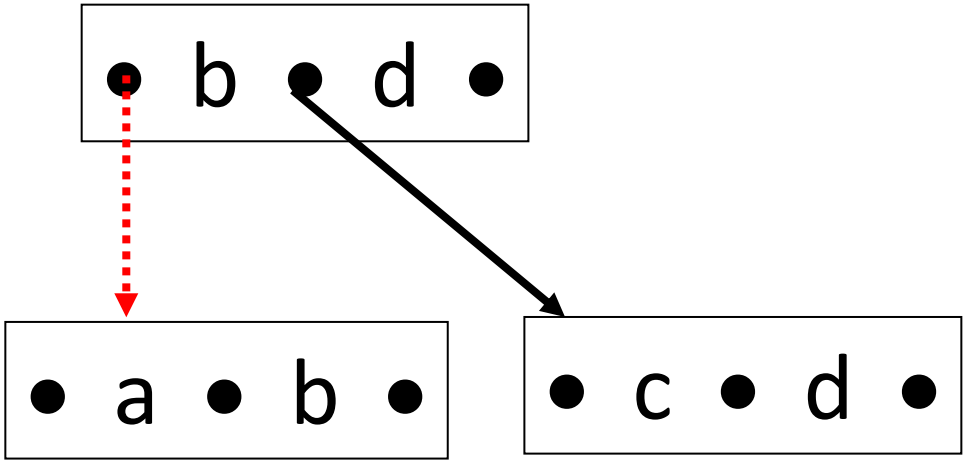


Problem without lock-coupling



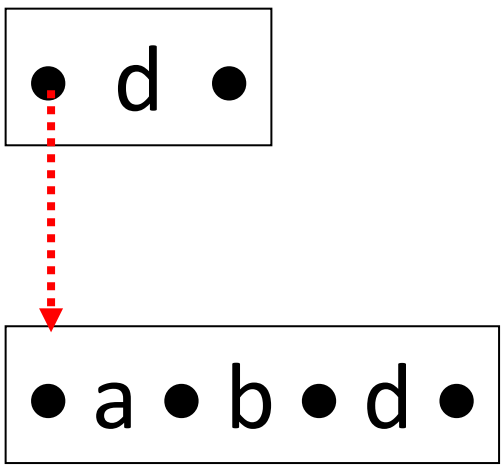
Search(d)

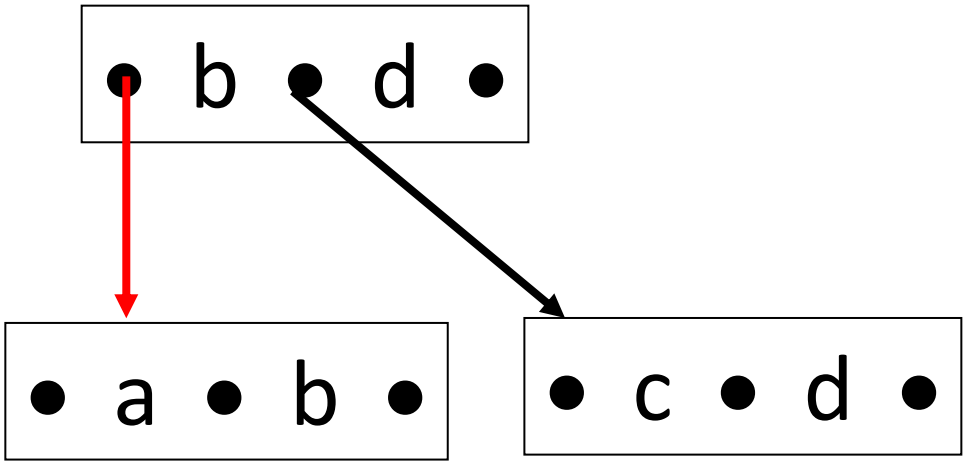
Problem without lock-coupling



Insert(c)

Problem without lock-coupling



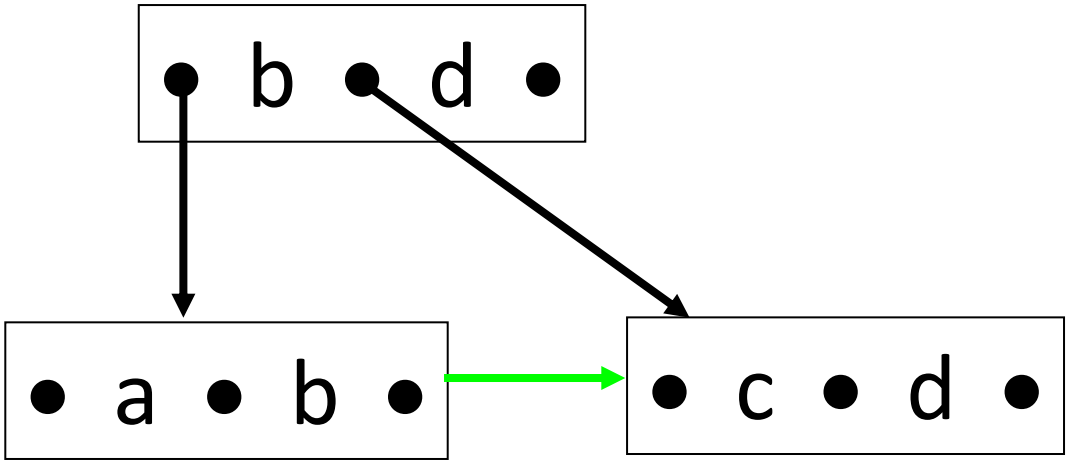


Search(d) fail

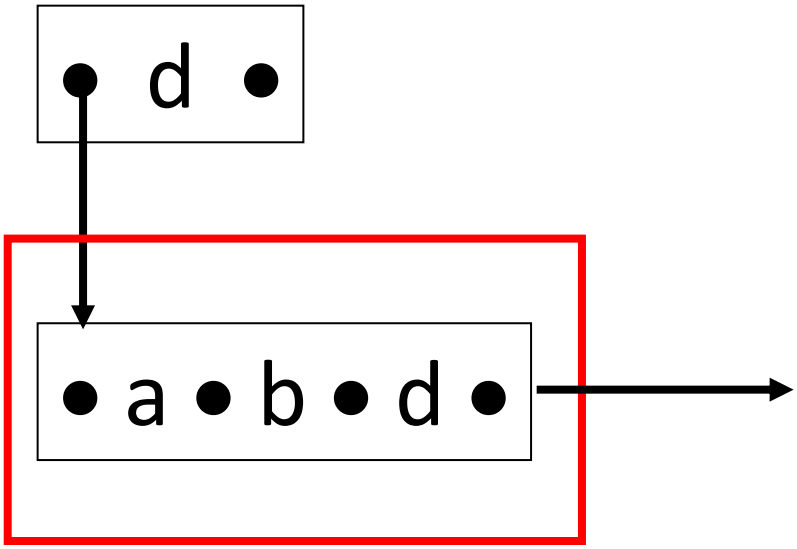
Insert(c)

B-link Structure

- By Lehman and Yao
- Add a rightlink : lock-coupling is unnecessary

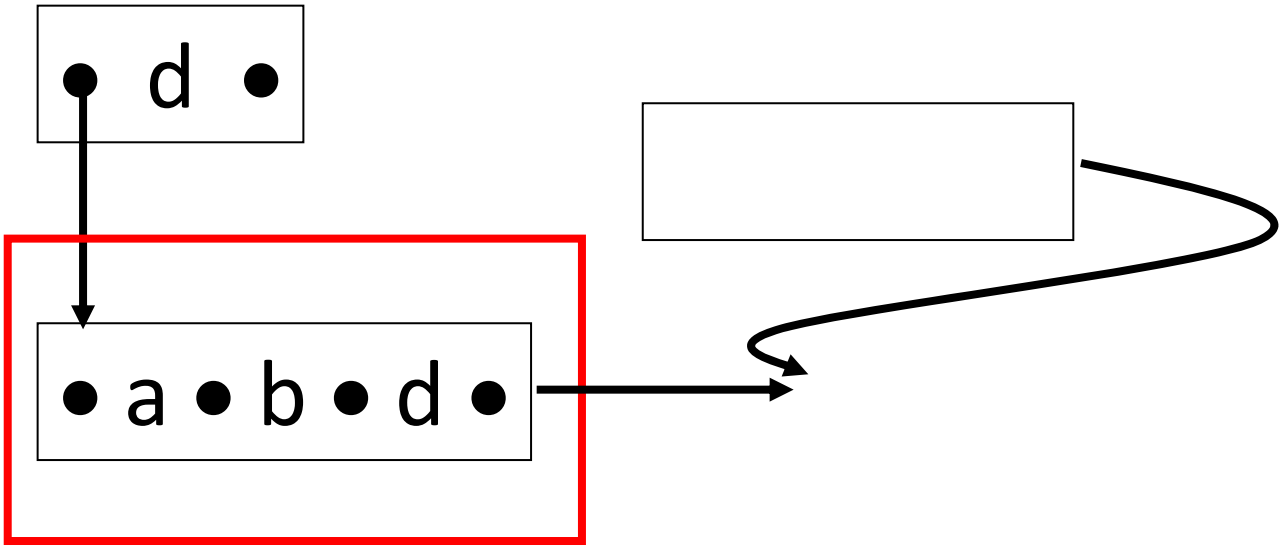


A Split in B-link Structure



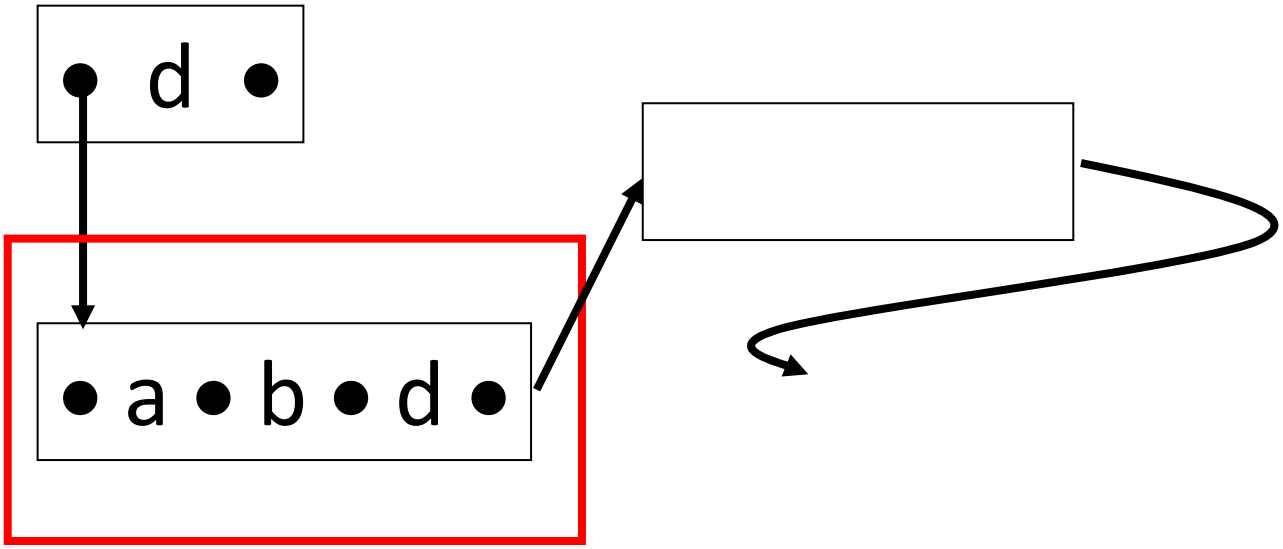
Insert(c): locate

A Split in B-link Structure



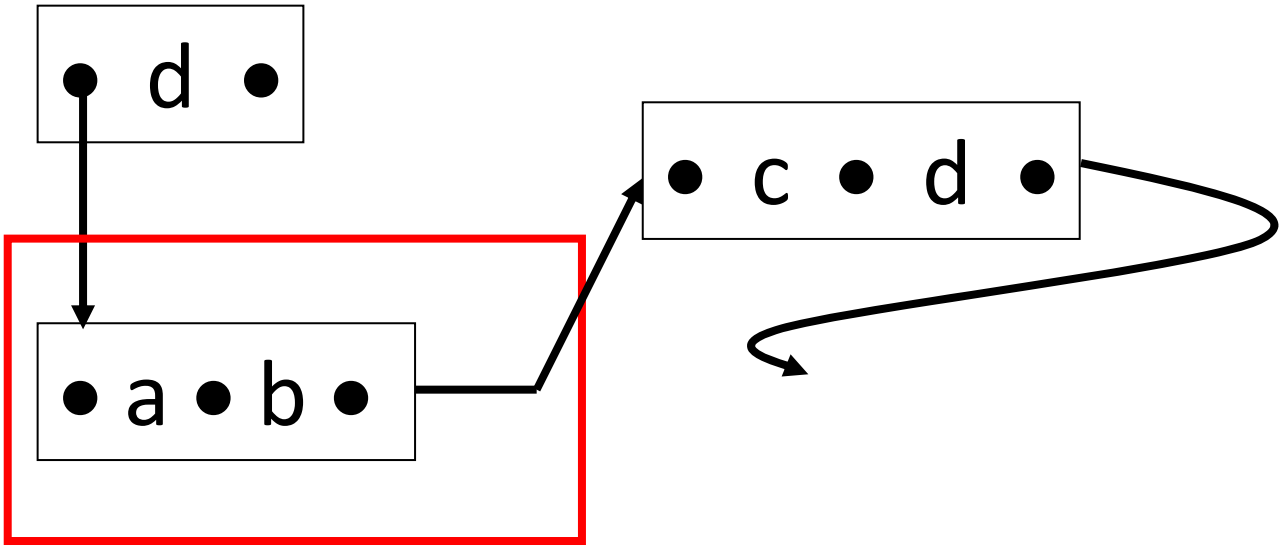
Insert(c): half-split

A Split in B-link Structure



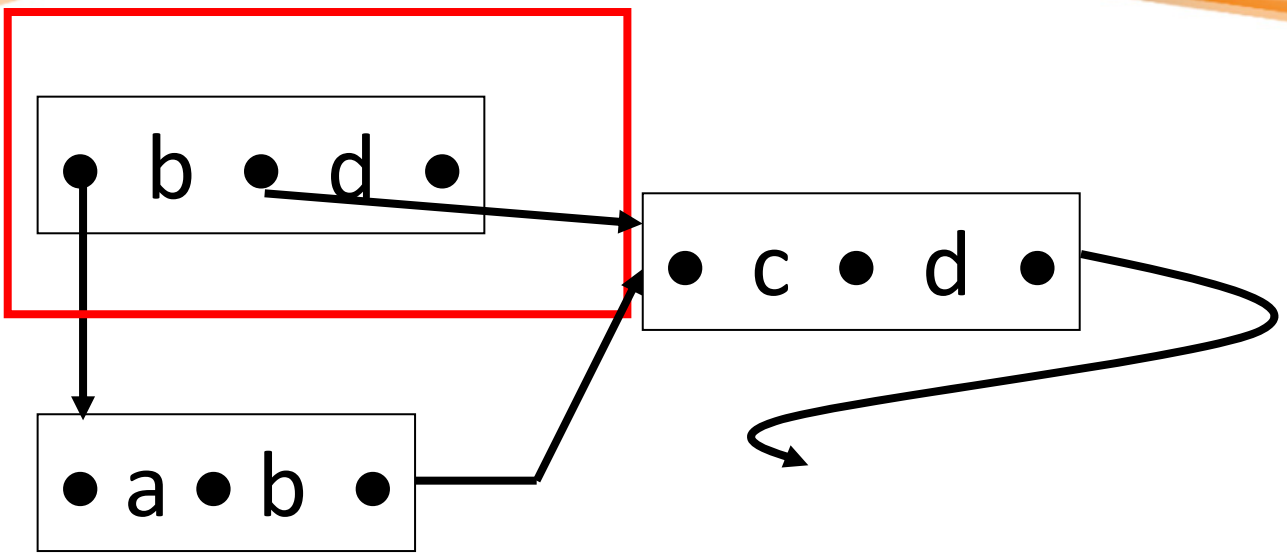
Insert(c): half-split

A Split in B-link Structure



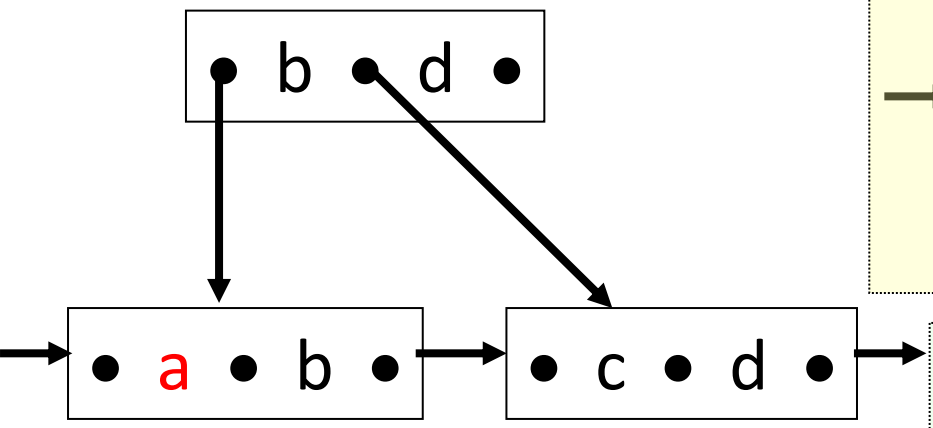
Insert(c): half-split

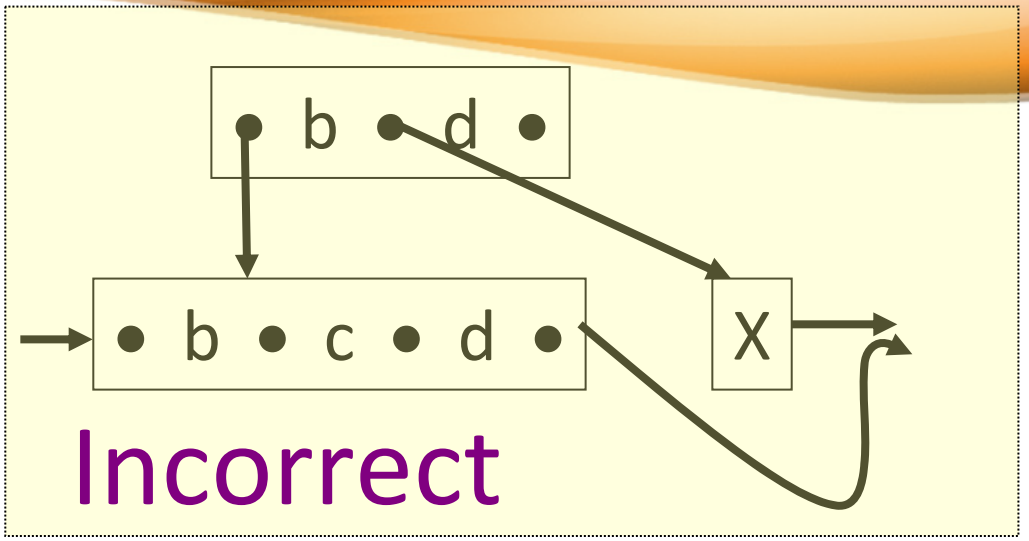
A Split in B-link Structure

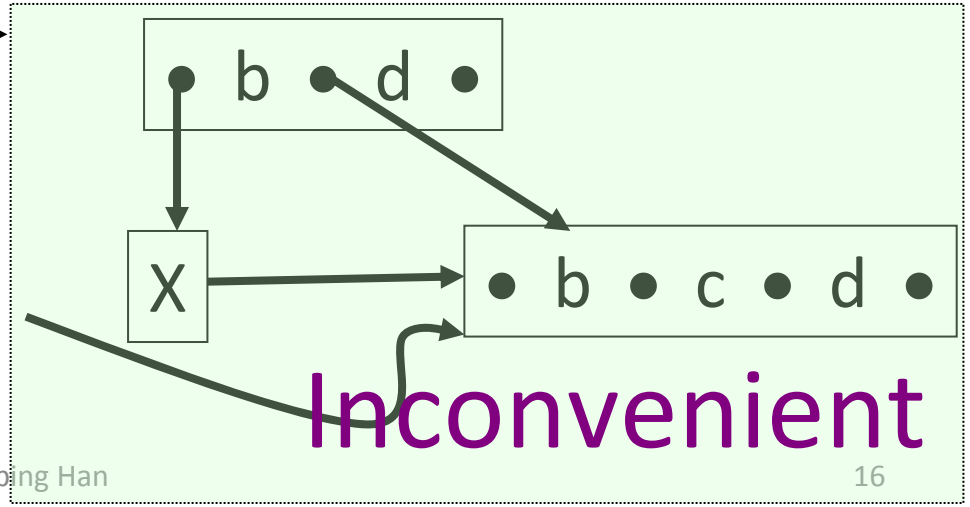


Insert(c): add-link

What about Merge

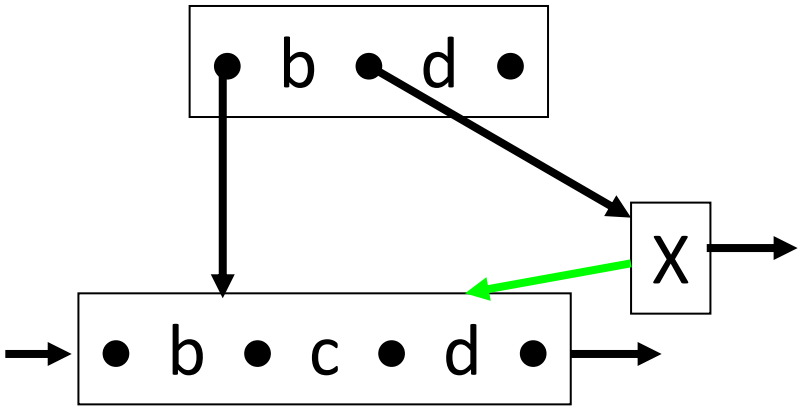




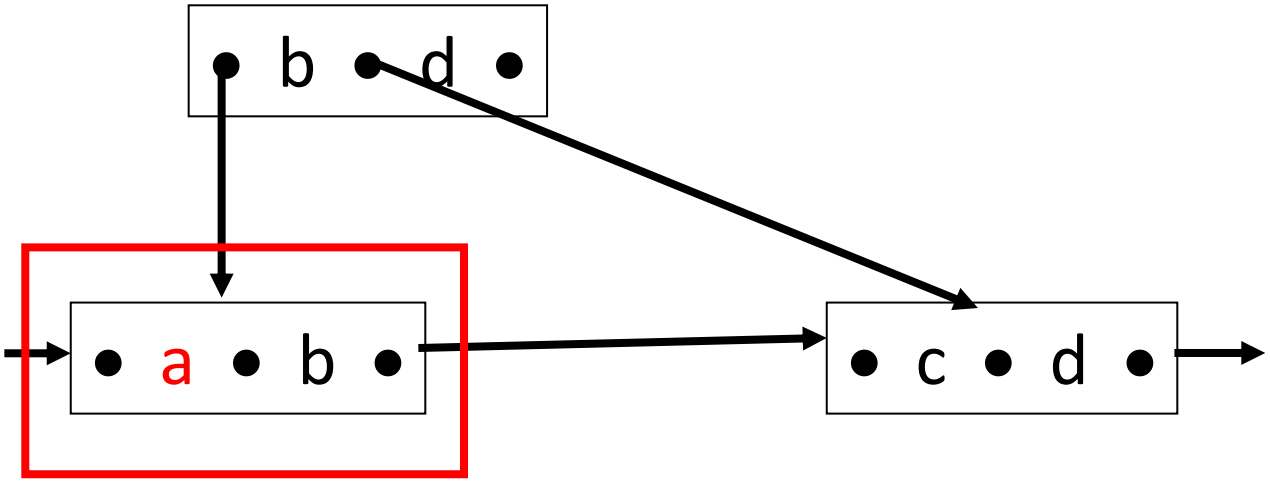


Further Modification to B-link

- By Lanin and Shasha
- Empty node add an outlink to the node where its content is merged to

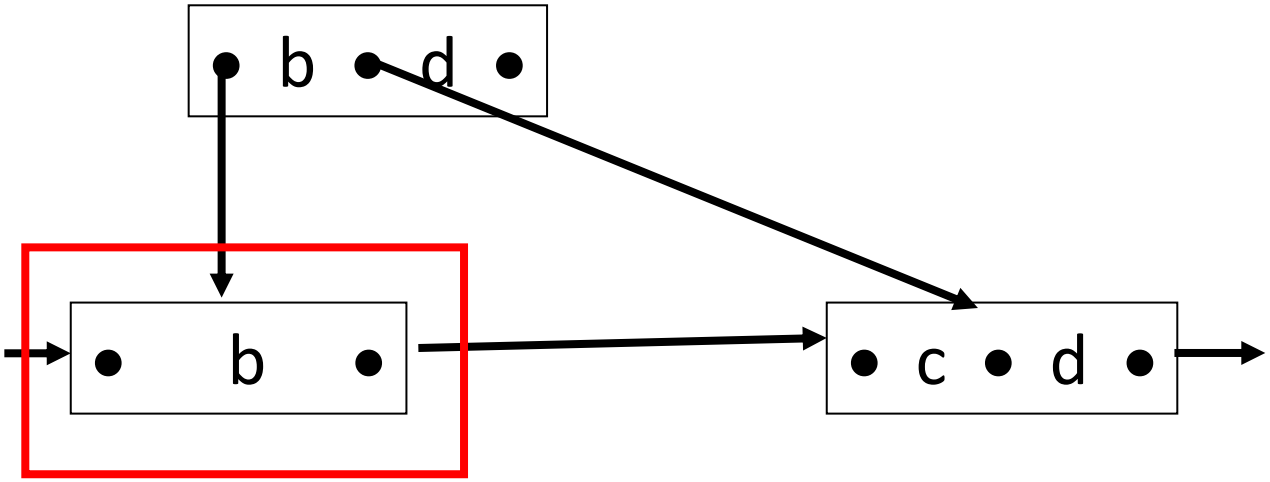


Merge in B-link structure



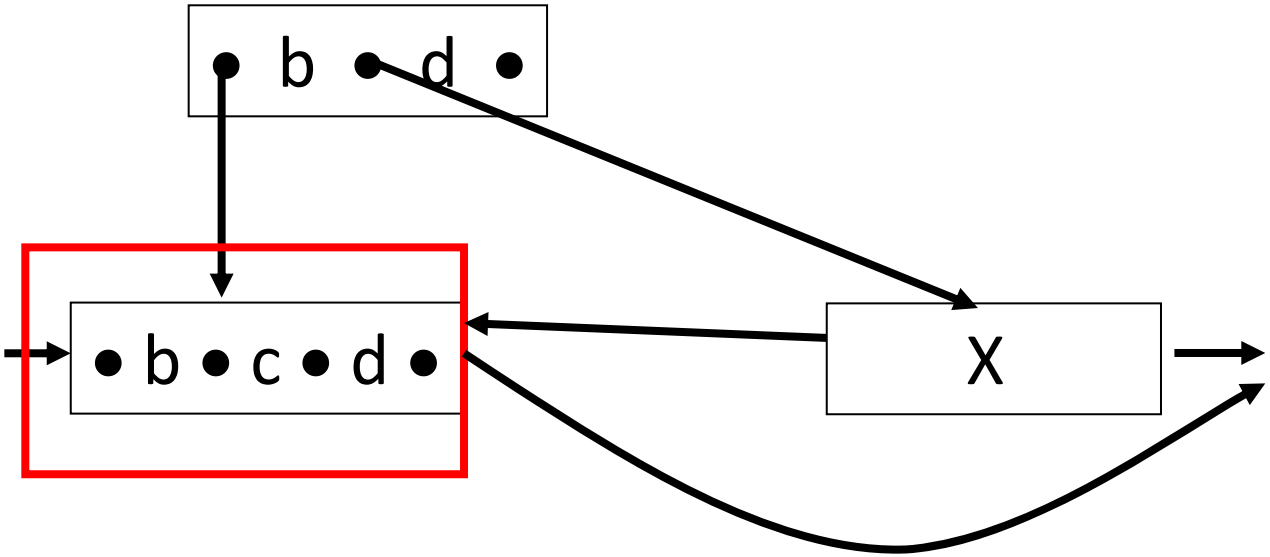
delete(a): locate

Merge in B-link structure



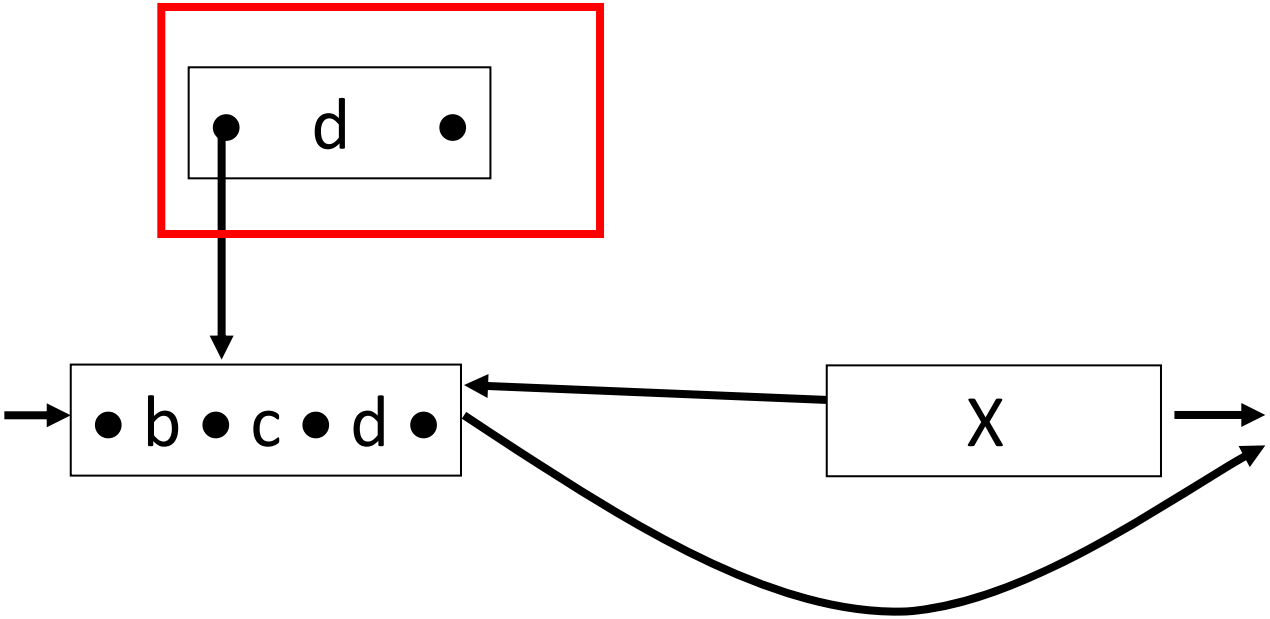
delete(a): locate

Merge in B-link structure



delete(a): half-merge

Merge in B-link structure



delete(a): remove link

Papers

- Lehman, P. L. and Yao, s. B. 1981. Efficient locking for concurrent operations on B-trees. *ACM Trans. Database Syst.* 6, 4 (Dec. 1981), 650-670. DOI= http://doi.acm.org.ezproxy.library.yorku.ca/10.1145/319628.319663
- Lanin, V. and Shasha, D. 1986. A symmetric concurrent B-tree algorithm. In *Proceedings of 1986 ACM Fall Joint Computer Conference* (Dallas, Texas, United States). IEEE Computer Society Press, Los Alamitos, CA, 380-389.