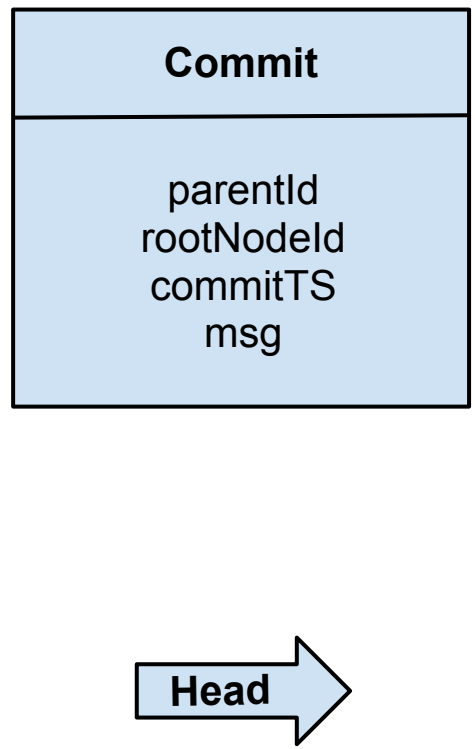
MicroKernel Revision Model

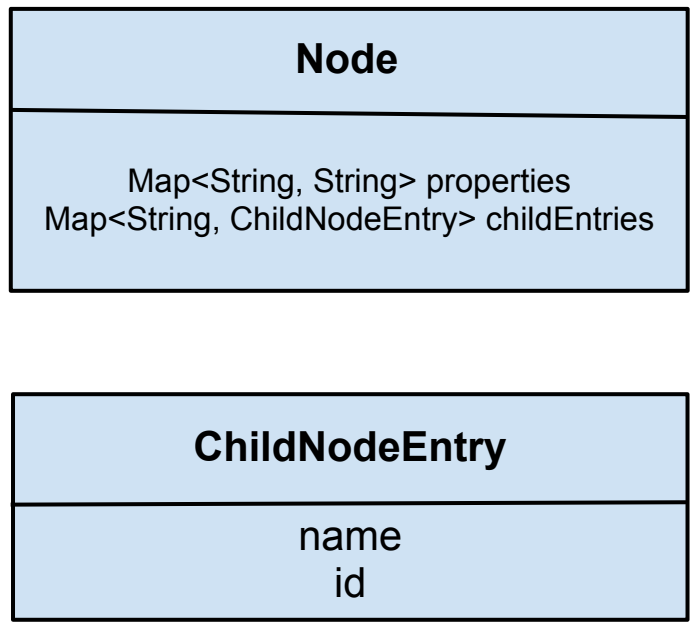
MVCC by Example

stefan@apache.org

Topics

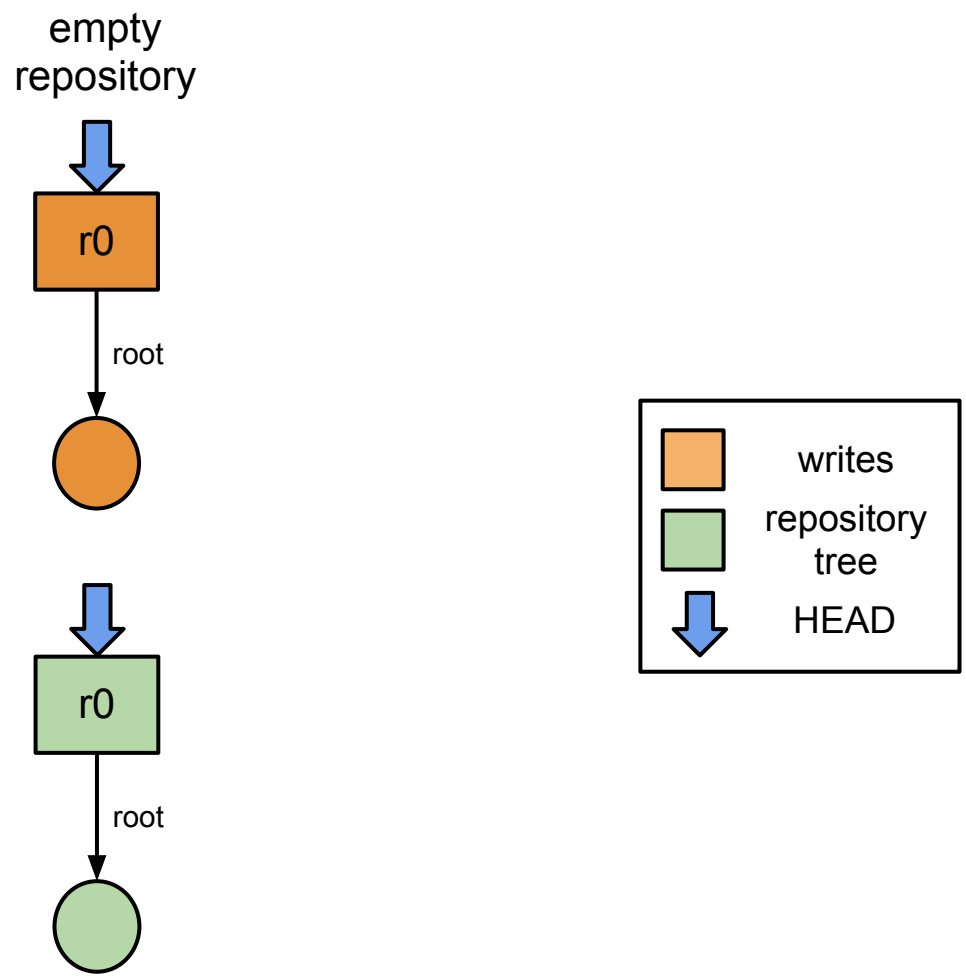
- MicroKernel Object Model
- MVCC in action
- Pros/Cons of MVCC approach


Commit
parentId
rootNodeId
commitTS
msg
Head
symbolic
reference to most
recent commit


Node
Map<String, String> properties
Map<String, ChildNodeEntry> childEntries
ChildNodeEntry
name
id

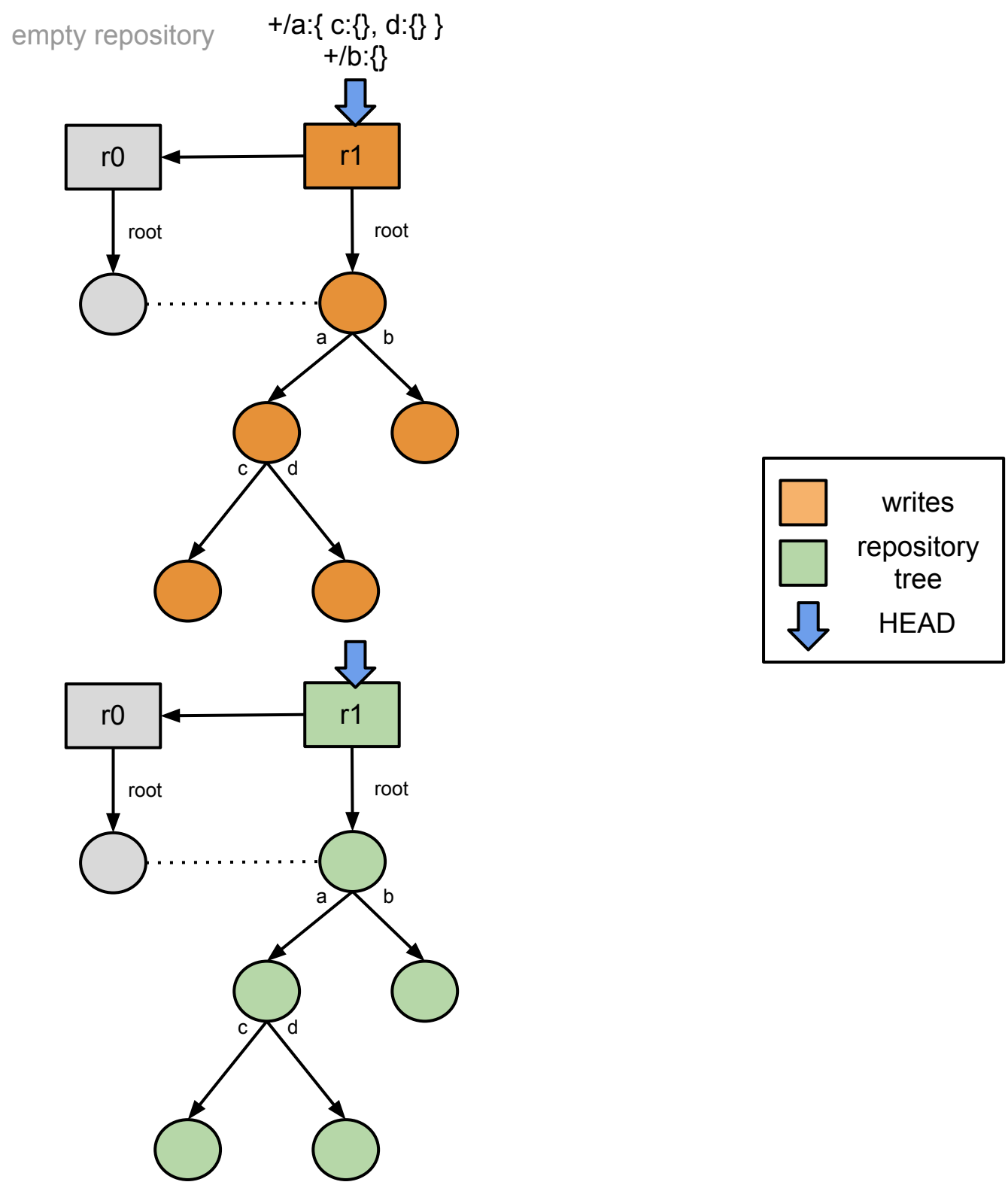
an example in 4 steps...

step 0: create an empty root node


empty
repository
r0
root
r0
root
writes
repository
tree
HEAD

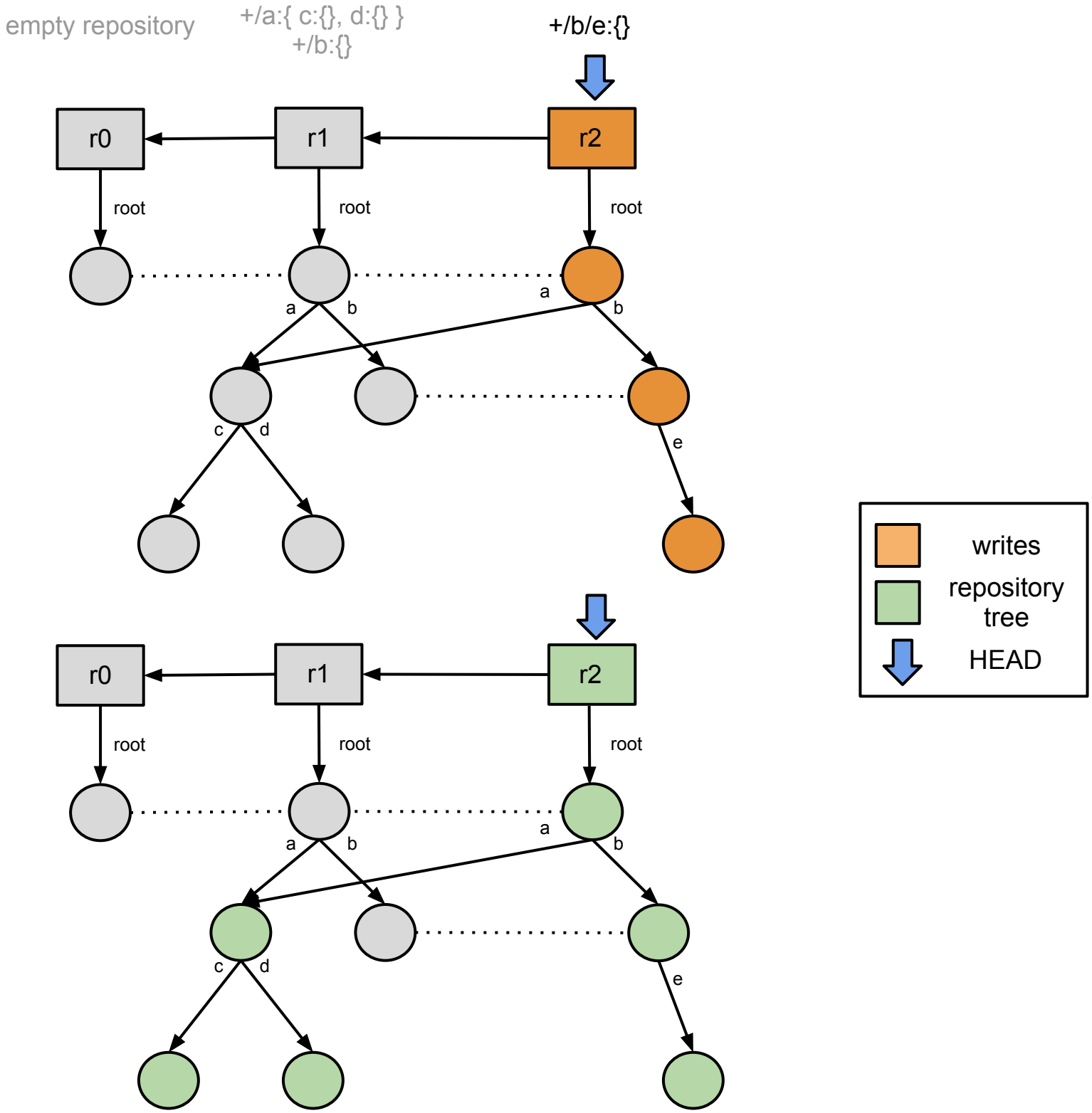
step 1: add nodes /a, /a/c, a/d and /b

+ /a : { c:{}, d:{} }
+ /b : {}


empty repository
+/a:{ c:{}, d:{} }
+/b:{}
r0
r1
root
root
a
b
c
d
r0
r1
root
root
a
b
c
d
writes
repository tree
HEAD

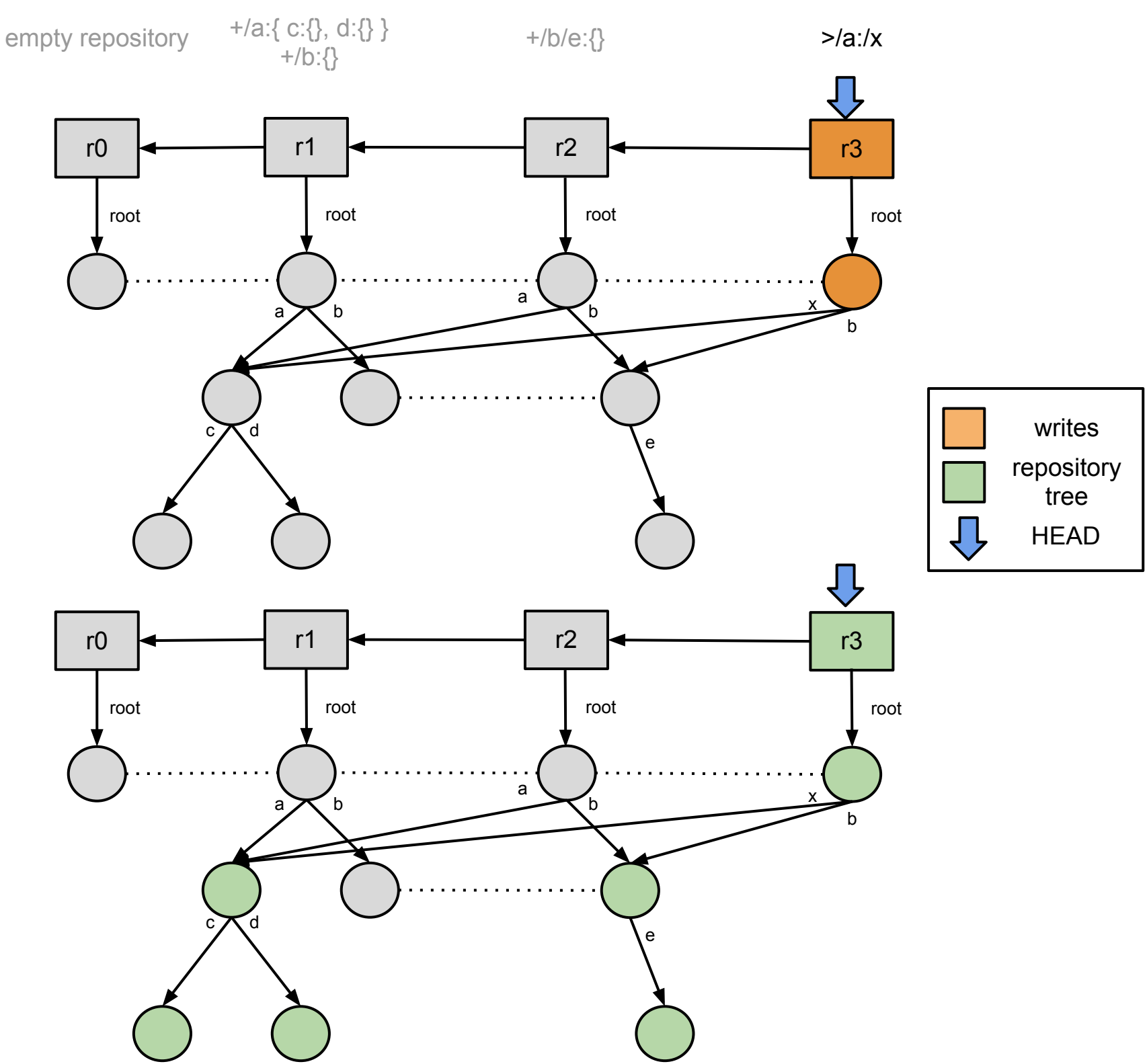
step 2: add node /b/e

+ /b/e {}


empty repository
+/a:{ c:{}, d:{} }
+/b:{}
+/b/e:{}
r0
r1
r2
root
root
root
a
b
a
b
c
d
e
writes
repository tree
HEAD
r0
r1
r2
root
root
root
a
b
a
b
c
d
e

step 3: move /a to /x

> /a : /x


empty repository
+/a:{ c:{}, d:{} }
+/b:{}
+/b/e:{}
>/a:/x
r0
r1
r2
r3
root
a
b
x
c
d
e
writes
repository tree
HEAD

MVCC pros

- writers don't interfere with readers and vice versa
- snapshot isolation (repeatable reads)
- improved concurrency
- minimal and very narrow point of synchronization on commit
- cheap copy & move

MVCC cons

- *heavy* on resources
- no stable identifiers across revisions (except for path),
  i.e. jcr-style references require an index
- non-trivial garbage collection/revision compacting problem...