

input: a `RepositoryInfo` *rep* representing the repository being analyzed and three sets of `CommitInfo` *db.commits*, *previousSnapshot* and *currentSnapshot*.

```

1: begin
2: commitsToDelete  $\leftarrow$  previousSnapshot \ currentSnapshot
3: delete commitsToDelete from database
4:
5: newCommits  $\leftarrow$  currentSnapshot \ previousSnapshot
6: commitsToInsert  $\leftarrow$  newCommits \ db.commits
7: commitsToUpdate  $\leftarrow$   $\{c \mid c \in db.commits \wedge$ 
8:    $((rep.pullsFrom \cup rep.pushesTo) \setminus c.foundIn) \neq \emptyset\}$ 
9: for each c  $\in$  commitsToInsert do
10:   UPDATEFOUNDIN(c, rep, currentSnapshot)
11: end for
12:
13: for each c  $\in$  commitsToUpdate do
14:   UPDATEFOUNDIN(c, rep, currentSnapshot)
15: end for
16:
17: insert commitsToInsert into database
18: update commitsToUpdate in database
19: orphanedCommits  $\leftarrow$   $\{c \mid c \in db.commits \wedge c.foundIn =$ 
20:    $\emptyset\}$ 
21: delete orphanedCommits from database
22: end
23:
24: procedure UPDATEFOUNDIN(c: CommitInfo, rep: RepositoryInfo,
25:   currentSnapshot: Set of CommitInfo)
26:   isAhead  $\leftarrow$  c  $\in$  currentSnapshot  $\wedge$ 
27:      $\exists r (r \in rep.pushesTo \wedge c \notin r.commits)$ 
28:   isBehind  $\leftarrow$  c  $\notin$  currentSnapshot  $\wedge$ 
29:      $\exists r (r \in rep.pullsFrom \wedge c \in r.commits)$ 
30:   if isBehind then
31:     c.foundIn  $\leftarrow$  c.foundIn  $\cup \{r \mid r \in rep.pullsFrom \wedge$ 
32:       c  $\in$  r.commits  $\}$ 
33:   end if
34:   if isAhead then
35:     c.foundIn  $\leftarrow$  c.foundIn  $\cup rep \cup$ 
36:        $\{r \mid r \in rep.pushesTo \wedge c \in r.commits\}$ 
37:   end if
38:   if  $(\neg(isBehind) \wedge \neg(isAhead))$  then
39:     if c  $\notin$  currentSnapshot then
40:       c.foundIn  $\leftarrow$   $((c.foundIn \setminus rep) \setminus rep.pushesTo) \setminus$ 
41:         rep.pullsFrom
42:     else
43:       isTracked  $\leftarrow$   $\exists b (b \in rep.branches \wedge$ 
44:         c  $\in b.reachableCommits)$ 
45:       if isTracked then
46:         c.foundIn  $\leftarrow$  c.foundIn  $\cup rep \cup$ 
47:           rep.pushesTo  $\cup rep.pullsFrom$ 
48:       else
49:         c.foundIn  $\leftarrow$  rep
50:       end if
51:     end if
52:   end if
53: end procedure

```