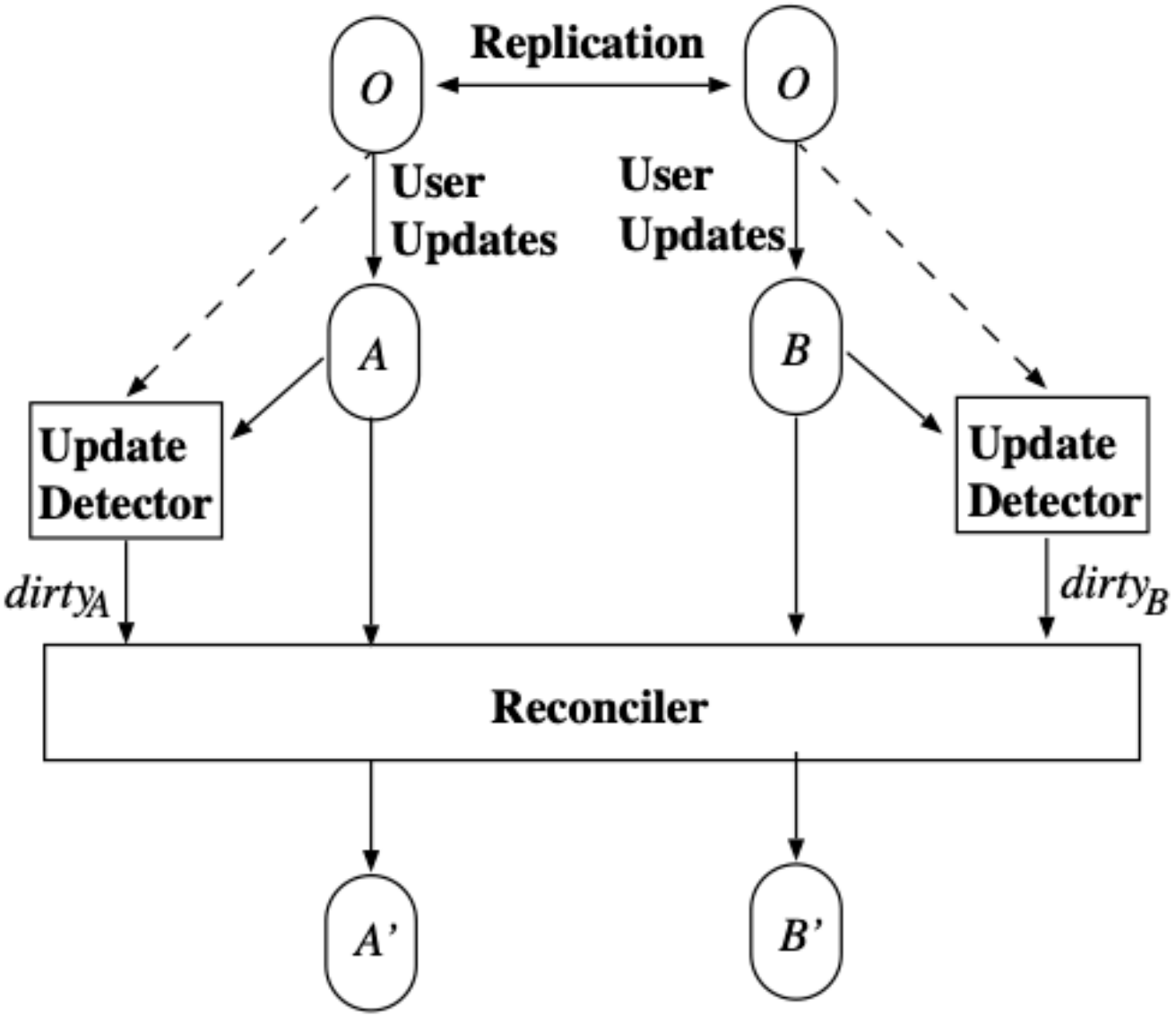
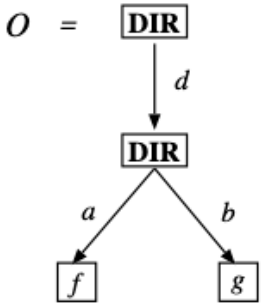
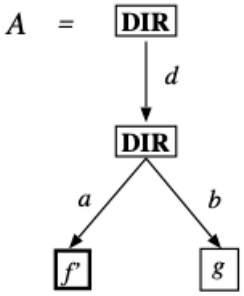
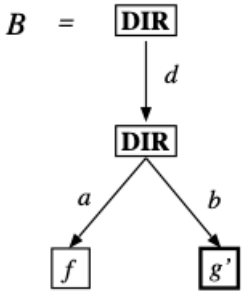
Exercise – File synchroniser

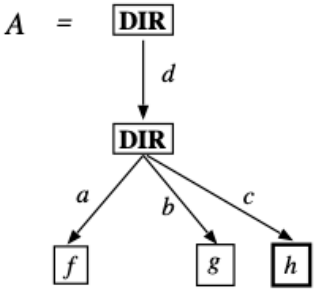


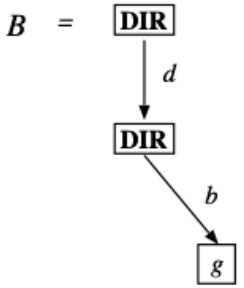
Exercise – File synchroniser

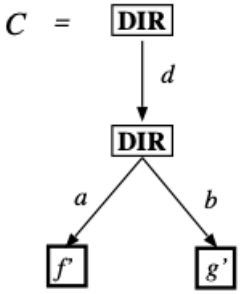


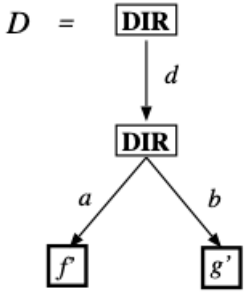


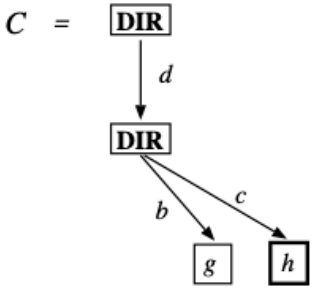


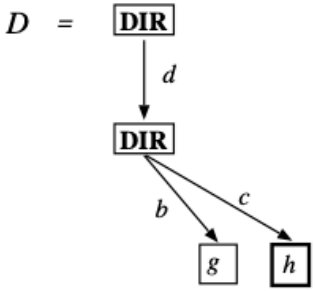




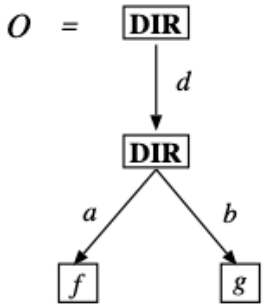


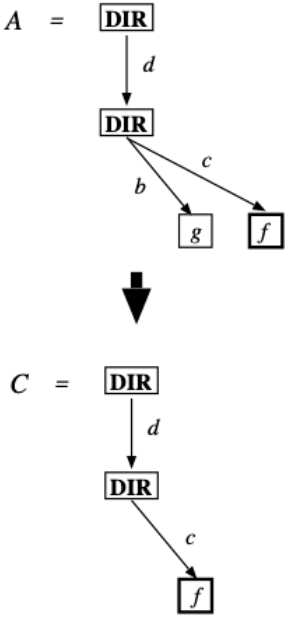


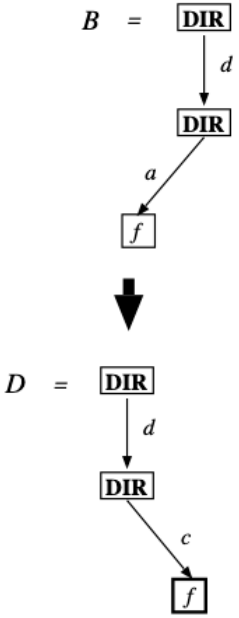


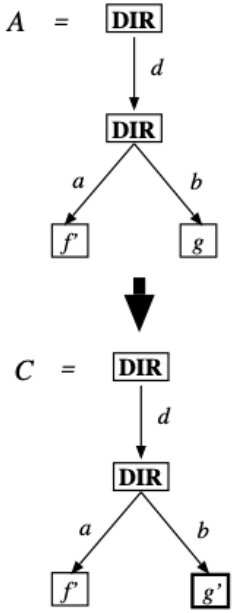


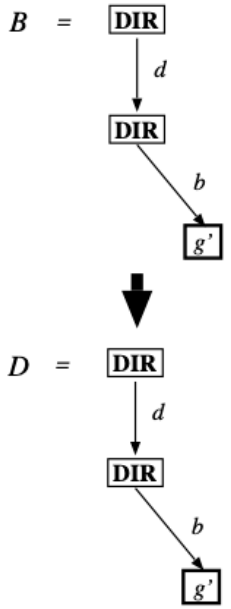
Exercise – File synchroniser











Exercise – File synchroniser

$$\neg dirty_A(p) \implies C(p) = D(p) = B(p)$$

$$\neg dirty_B(p) \implies C(p) = D(p) = A(p)$$

$$isdir_{A,B}(p) \implies isdir_{C,D}(p)$$

$$dirty_A(p) \wedge dirty_B(p) \wedge \neg isdir_{A,B}(p) \implies C(p) = A(p) \wedge D(p) = B(p)$$

Exercise – File synchroniser

$$
\begin{aligned}
&recon(A, B, p) \;=\\
&\quad 1)\;\; \text{if } \neg dirty_A(p) \wedge \neg dirty_B(p)\\
&\qquad\quad \text{then } (A, B)\\
&\quad 2)\;\; \text{else if } isdir_{A,B}(p)\\
&\qquad\quad \text{then let } \{p_1, p_2, \ldots, p_n\} = children_{A,B}(p)\\
&\qquad\qquad\qquad\quad (\text{in lexicographic order})\\
&\qquad\quad \text{in let } (A_0, B_0) \;=\; (A, B)\\
&\qquad\qquad\quad \text{let } (A_{i+1}, B_{i+1}) \;=\; recon(A_i, B_i, p_{i+1})\\
&\qquad\qquad\qquad\quad \text{for } 0 \leq i < n\\
&\qquad\qquad\quad \text{in } (A_n, B_n)\\
&\quad 3)\;\; \text{else if } \neg dirty_A(p)\\
&\qquad\quad \text{then } (A \overset{p}{\leftarrow} B, B)\\
&\quad 4)\;\; \text{else if } \neg dirty_B(p)\\
&\qquad\quad \text{then } (A, B \overset{p}{\leftarrow} A)\\
&\quad 5)\;\; \text{else}\\
&\qquad\quad (A, B).
\end{aligned}
$$

Exercise – File synchroniser

```
public interface FileSystem {
        public String getRoot();
        public String getParent(String path);
        public List<String> getChildren(String path);
        public List<String> getAncestors(String path);
        public String getAbsolutePath(String relativePath);
        public String getRelativePath(String absolutePath);
        public void replace(String absolutePathTargetFS, FileSystem fsSource,  String absolutePathSourceFS);
        public FileSystem getReference();
        public File createDirectory(String path);
        public void fileCopy(File input, File output) throws Exception;
        ….
}

public class LocalFileSystem implements FileSystem {…}
```

Exercise – File synchroniser

```
public class Synchronizer {
    public void synchronize(FileSystem fs1, FileSystem fs2) {
        FileSystem refCopy1 = fs1.getReference();
        FileSystem refCopy2 = fs2.getReference();

        List<String> dirtyPaths1 = computeDirty(refCopy1, fs1, "");
        List<String> dirtyPaths2 = computeDirty(refCopy2, fs2, "");
        reconcile(fs1, dirtyPaths1, fs2, dirtyPaths2, "");
    }
    public void reconcile(FileSystem fs1, List<String> dirtyPaths1, FileSystem fs2, List<String> dirtyPaths2,
String currentRelativePath) {…}

    public List<String> computeDirty(FileSystem lastSync, FileSystem fs, String currentRelativePath) {…}
…
}
```