

## 6.826 Problem Set 2 Clarifications

### Problem 1.

The intention of grouping "code" and "decode" functions in the module header is to state the invariant that they are inverse of one another. You can use the code and decode functions by simply writing `code(b)` and `decode(b)` where `b` is a Byte.

When determining whether `CypherFile` implements `File` and whether `File` implements `CypherFile`, consider only traces that contain `Read` and `WriteAtomic` operations, do not worry about the fact that `File` has some additional operations. (You still need to look at the implementation of `NewFile` because it is used to implement `WriteAtomic`.)

Recall that the action corresponding to a function call or atomic procedure call is labelled by both the parameters and the return result of the function or procedure.

When analyzing the effect of `NewFile` operation you may assume that for every call `WriteAtomic(pn,x,data)` the write does not start after the end of file: it is always the case that  $x \leq d(pn).size$

### Problem 2.

d) You should consider the set of states that can be generated by `Read` and `Append` operations in a file system with existing `EMPTY` files. The point of this requirement is that the state is nontrivial, but is still generated only by `Read` and `Append` operations. You assume in the d) part that the invariant in b) holds for the state.

### Problem 3.

You will need to make modifications to the state to maintain the additional information which will allow you to implement the cache replacement policies. Write all modifications that you make to the original implementation.