```
let (int2,_2) = P2::open::<R<P2>>(...)?;
 2:
 3:
       Chaperon::session("foo", || {
 4:
             let b = P2::transaction(|j| {
 5:
                    let mut int2 = int2.borrow_mut(j);
 6:
                     let out = *int2;
 7:
                    *int2 = 0;
 8:
                     out
 9:
               });
10:
             P1::transaction(|j| {
                     let mut int1 = int1.borrow_mut(j);
11:
12:
                    *int1 = b;
               });
13:
       }).expect("Unsuccessful")
14:
                                      Postponed commits
                                          13
                                               14
                                                                      14 14 14
                                      10
                                                              14
                                                       14
                              P2::transaction is
                                           P1::transaction is
                                                                      Delete Chaperon file
                                                                           Delete
                                                                               Delete P2::journal
           Chaperoned
                   Create Chaperon file
                       check it in
                          Create P2::journal and
                                      Create P1::journal and
                                                Chaperoned session
                                                    and checks out
                                                       P1::transaction commits
                                                           and checks out
                                                               P2::transaction
                                   check it in
                                                                          P1::journal
           session
                                                               commits
                                                ends
                               done
                                           done
                                                       Rollback 🗘
                                                                               Forward
```

type R<P: MemPool> = LogRefCell<i32, P>;

let (int1,\_1) = P1::open::<R<P1>>(...)?;

1: