Assume for Simplicity:

- At most one book can be borrowed at any time. In addition, at most one hold can be placed at any time.
- BOD = Book-overdue notice and BHA = Book-oon-hold-available notice. (Assume people are non-mischievous and thus no restrictionis is put on the number of BOD and BHA notices.)
A SIMPLIFIED MODEL USING GUARDS

Following state-pairs are merged:

- \{hasBorrowed, hasBOD\}, \{borrowedAndHasHold, hasBODandHold\}, and \{borrowedAndHasBHA, hasBODandBHA\}.
- The part "?BOD" in a state-name means BOD-notice may or may not have been received. The guard "[¬BOD]" means BOD-notice not received.

Question:

- Show the state-diagram if we merge the states has-borrowed-hasHold-?BOD and has-borrowed-hasBHA-?BOD, (Hint: some more transitions will now have new guards.) Should we also merge has-hold and has-BHA at the same time?
- Modify the model by adding "search" operation for books.
IMPLEMENTATION OF THE STATES

State = (BOD, BHA):

- **BOD** = −1 (initial value), BOD = 0 (borrowed but no BOD-notice received), and BOD ≥ 1 (otherwise).
- **BHA** = −1 (initial value), BHA = 1 (has hold but no BHA-notice received), and BHA ≥ 1 (otherwise).
- Shown next to each state is the properties of these variables, and next to each action is shown the update of these variables.

- How will you handle the case of multiple books borrowed and holds on multiple books?
3.4 DOMINATION-TREE BASED DEVELOPMENT PLAN

Development (and Test Plan):

1. startMembership, stopMembership (test them together)
2. borrow, return (test borrow + return; recvBOD, renew (test them together, and also test them with borrow and return)
3. putHold, cancelHold (test putHold + cancelHold; also test them with borrow + … + return)
4. recvBHA (test recvBHA + cancelHold; also test return + recvBHA + cancelHold), etc.