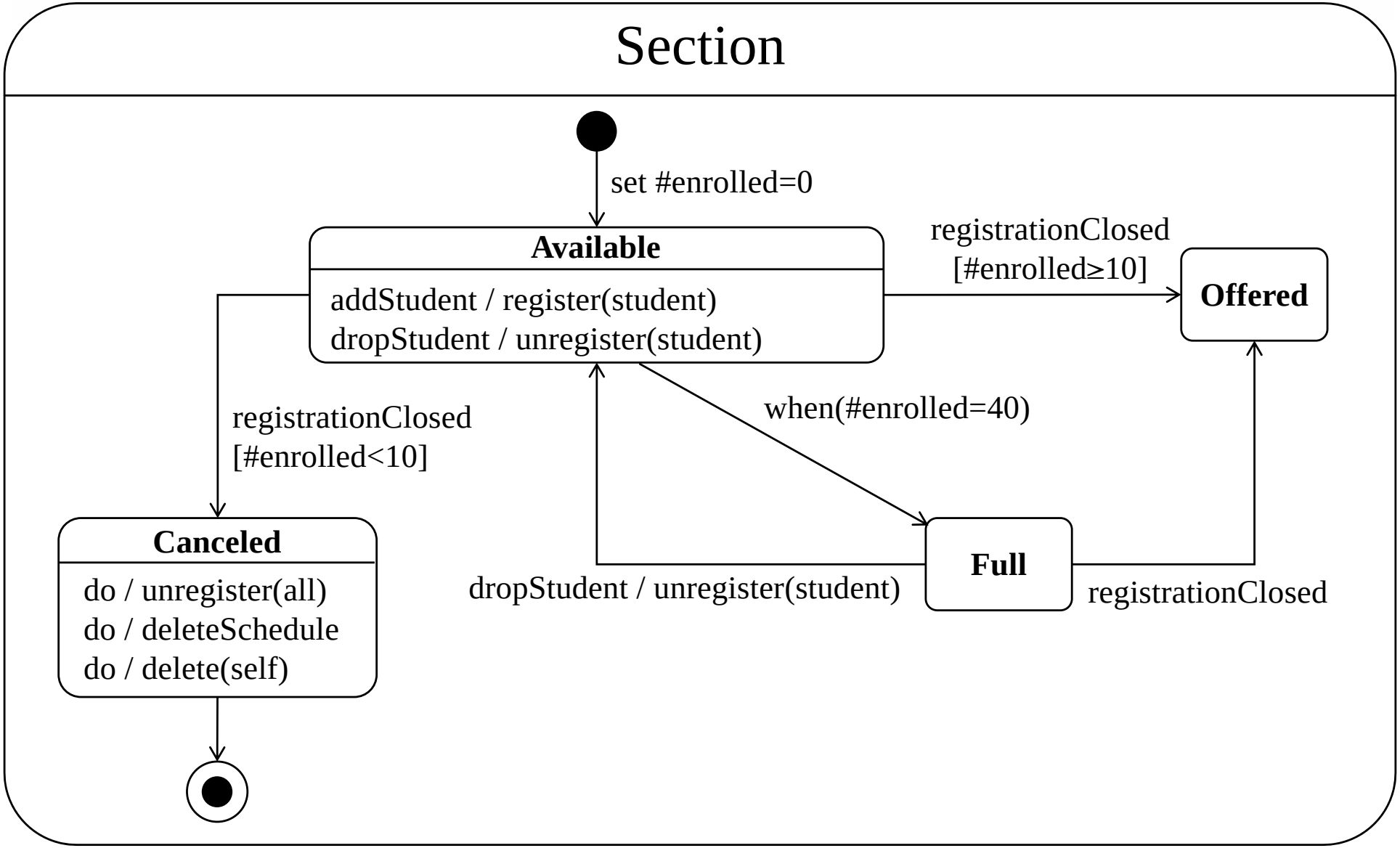
Software Engineering

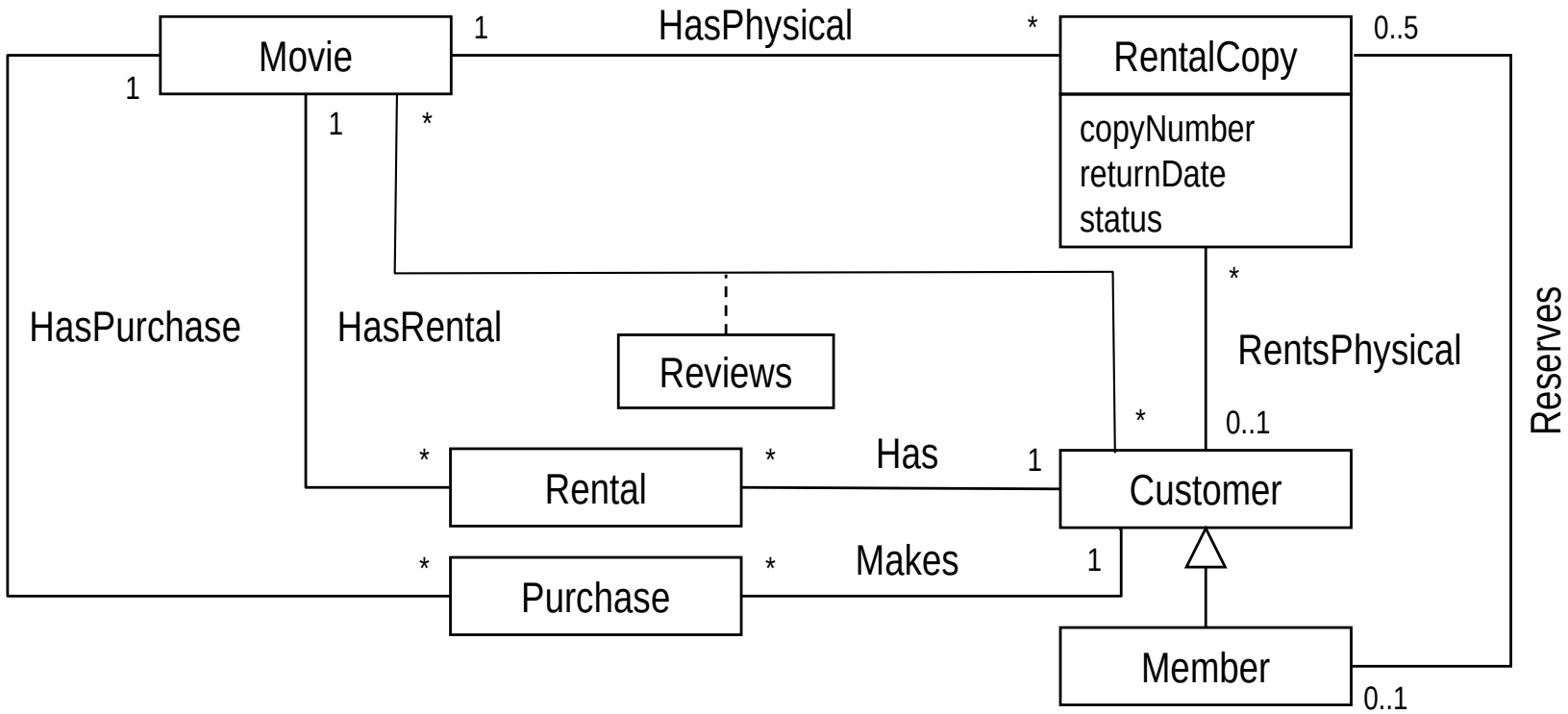
System Analysis and Design

ASU STATE MACHINE DIAGRAM: SECTION CLASS



EXERCISE: STATE MACHINE DIAGRAM

Part of the information kept in the RentalCopy class in the domain model shown below is the rental status of a movie. Construct a state machine diagram showing the states that an instance of the RentalCopy class can be in *with respect to its rental status*. Show only the states, transitions and the events and/or conditions, if any, that cause a transition to be taken. *Do not show the activities that can occur within a state.*



EXERCISE: STATE MACHINE DIAGRAM

The problem statement requirements that could be relevant to determining the states of a RentalCopy object:

- It must be able to record which movies are sold and rented and by whom.
- For sold movies, the quantity sold should be recorded; for physical movie rental, which copy is rented and when it is due back should be recorded.
- The system should keep track of overdue rentals of physical movies and send email notices to customers who have movies overdue.
- Members should be able to make reservations for physical movie rentals either in person at the shop, by telephone or via the Web.
- A member can reserve at most five physical movies at any one time, but there is no limit on how many physical movies a member or nonmember can rent at any one time.
- A sales clerk should be able to sell and rent physical movies and process the return of rented physical movies.

EXERCISE: STATE MACHINE DIAGRAM—ANALYSIS

- It must be able to record which movies are sold and rented and by whom.

  states: Available; Rented
  events: rent
  transitions: Available → rent → Rented

- For sold movies, the quantity sold should be recorded; for physical movie rental, which copy is rented and when it is due back should be recorded.

  No new state, event or transition information in this statement.

- The system should keep track of overdue rentals of physical movies and send email notices to customers who have movies overdue.

  states: Overdue
  events: when(date>returnDate)
  transitions: Rented → when(date>returnDate) → Overdue
  actions: send overdue notice

EXERCISE: STATE MACHINE DIAGRAM—ANALYSIS

- Members should be able to make reservations for physical movie rentals either in person at the shop, by telephone or via the Web.

  states: Reserved
  events: reserve
  transitions: Available → reserve → Reserved
  Reserved → rent → Rented

- A member can reserve at most five physical movies at any one time, but there is no limit on how many physical movies a member or nonmember can rent at any one time.

  No state, event or transition information in this statement.

- A sales clerk should be able to sell and rent physical movies and process the return of rented physical movies.

  event: return
  transitions: Rented → return → Available
  Overdue → return → Available

EXERCISE: STATE MACHINE DIAGRAM—ANALYSIS

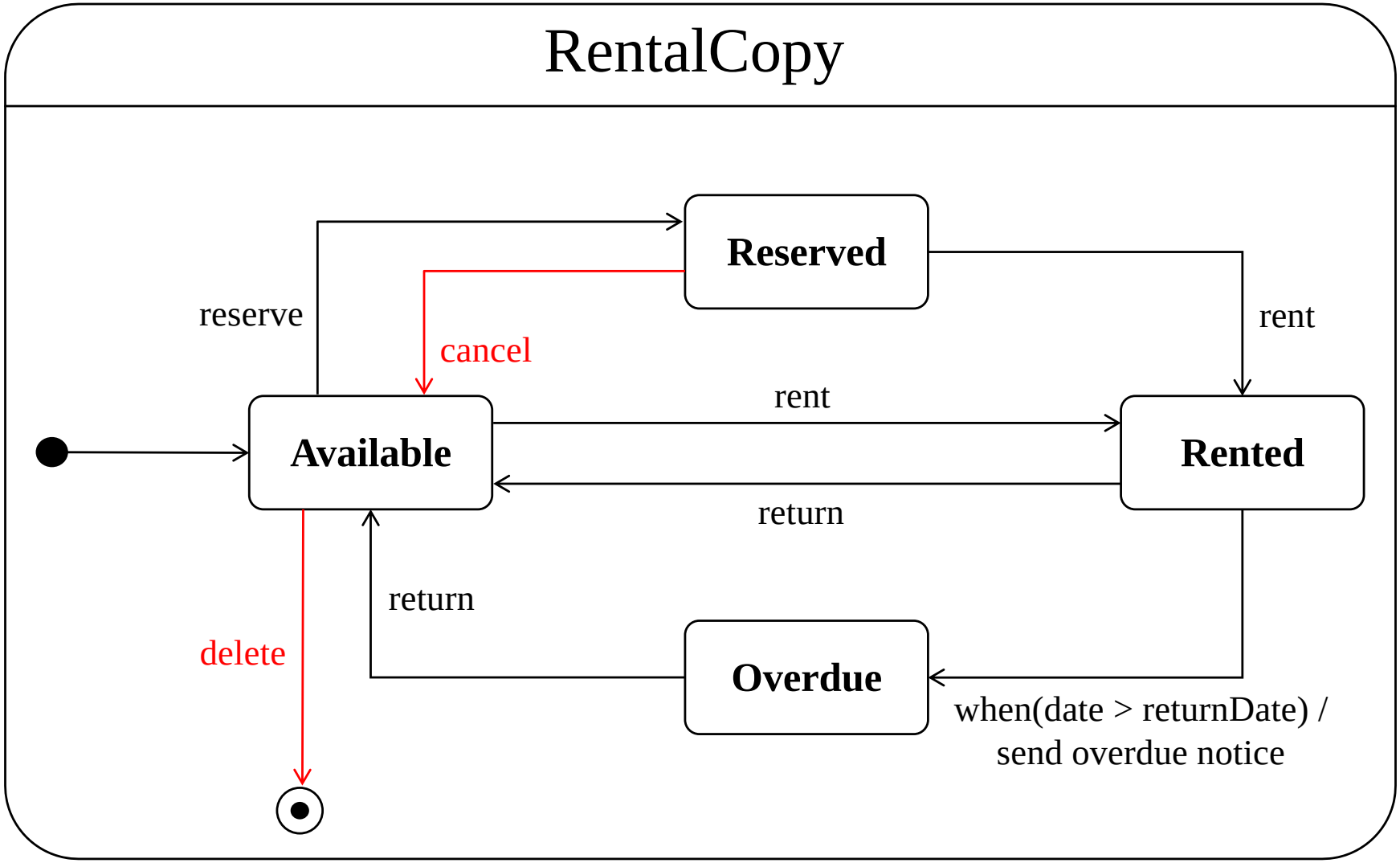
Other reasonable events and transitions (but not explicitly stated)

events: cancel, delete

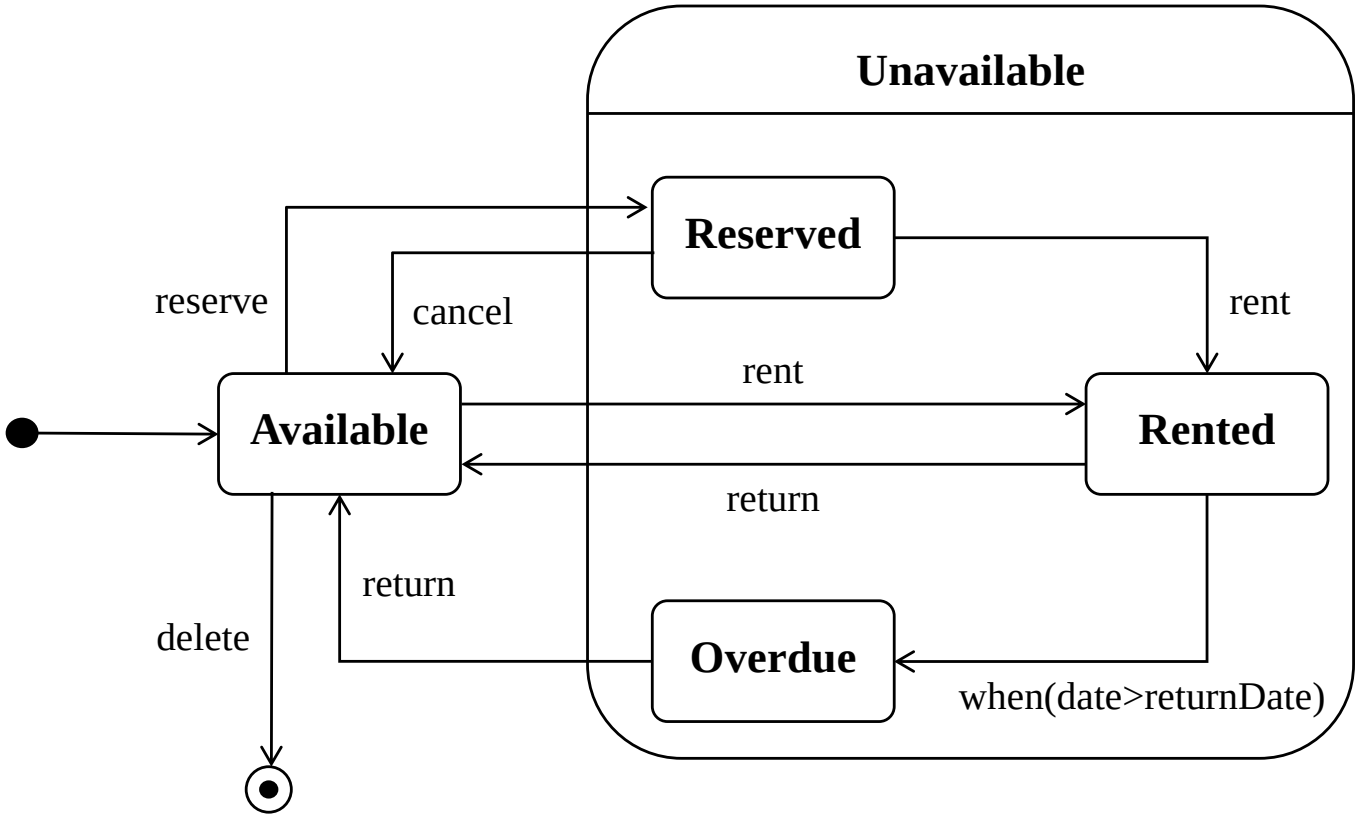
transitions: Reserved → cancel → Available

Available → delete → Final state

EXERCISE: STATE MACHINE DIAGRAM—SOLUTION



EXERCISE: STATE MACHINE DIAGRAM—SOLUTION



We could create a composite state Unavailable.

EXERCISE: STATE MACHINE DIAGRAM—COMMON ERRORS

- Showing states/transitions/events not applicable to the object under consideration.
  e.g., reservedCopy < 5 applies to Member objects, not to RentalCopy objects
  e.g., the Buy state applies to Video objects that are for sale, not to RentalCopy objects.

- Having transitions with no events or with several events.

- Missing/incorrect transitions.
  e.g., an overdue video is not destroyed! It can be returned.

- Having states with no outgoing transition.
  e.g., Overdue, Reserved

EXERCISE: STATE MACHINE DIAGRAM—COMMON ERRORS

- Using attributes not in the object.
  e.g., #copies

- Using states not in the problem statement.
  e.g., stolen, sold, lost

- Using incorrect states.
  e.g., Customer, Sales clerk, VideoCopy