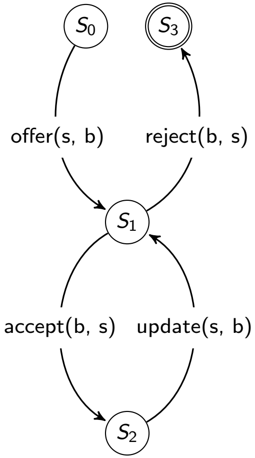
Example Finite State Machine Representation

Part of a purchase protocol that deals with making offers

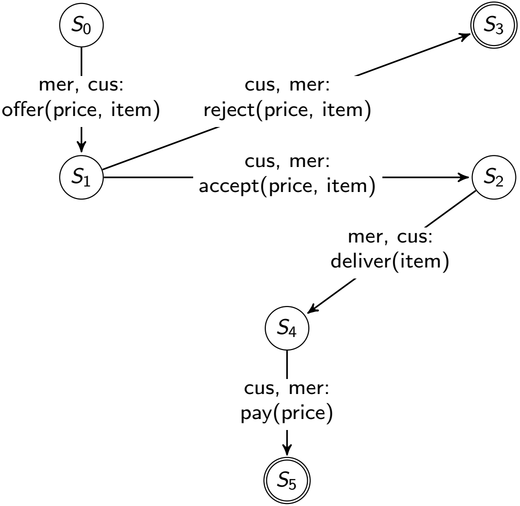
- Roles: buyer (b) and seller (s)
- Transitions labeled with messages
  - Specify legal message flows



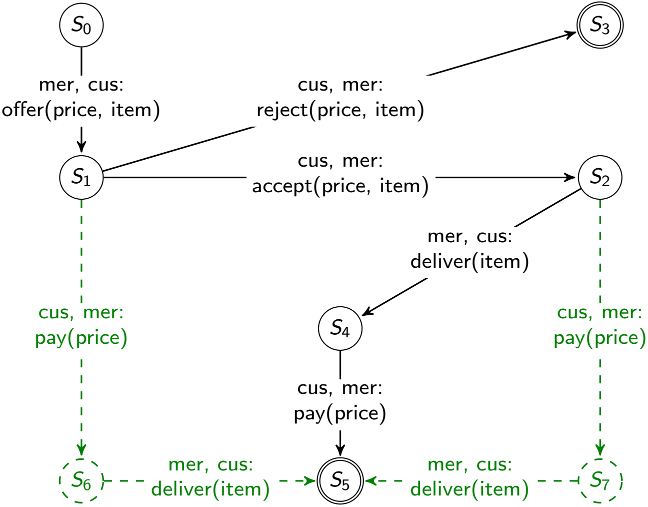
Finite State Machine (NetBill Protocol)

Legitimate protocol: specifies interactions, not internal decision making

- Roles: merchant (mer) and customer (cus)
- Transitions: messages sender, receiver
- Enactment: *reject*
- Enactment: *accept*, *deliver*, *pay*
- Correctness: purely operational terms (sequences of messages, not meanings)
  - Excludes legitimate enactments (next picture)



State Machine Example: Generalized



Produce Ever-Larger FSMs with Additional Enactments?

Can we not use FSMs to capture all reasonable paths?

- Complicates implementation
- Not runtime but hardwired flexibility
- Presupposes an arbitrary selection of paths: which path is reasonable, which is not?
- The same criticism applies to the low-level conception, even if specified declaratively in logic

Evaluation of the FSM Representation

Does not account for meanings of messages

- Flexibility: limited by over-specifying message order and occurrence
- Compliance checking: easy since the protocol is explicit about message order and occurrence
    - Failure to comply may not indicate an application-level problem
- Implicit meanings: loss of interoperability due to inconsistent interpretations of messages
- Designers agree offline regarding the meanings, thereby limiting heterogeneity

State Diagrams

Formalized in UML 2.0 from Harel's statecharts

Generalize over finite state machines

- Condition or guard on a transition
- Superstate (OR-state): being in a substate entails being in the superstate
  - Natural for summarizing states that bear similar meanings and support similar transitions
- Parallel states indicate being in the each of the states at the same time (AND-state)
  - Cartesian product of the individual states
  - Natural for expressing mutually independent components of the state

Exercise: Diagram the Purchase Protocol

First as we specified and second with concurrent Pay and Ship subprotocols

Exercise: Diagram the Purchase Protocol with Return and Refund

Exercise: Diagram Precedence, Occurrence, Exclusion

Across two messages, m_1 and m_2

Applying State Diagrams in Our Setting

Behavior descriptions, but of social behavior

- In general, sequence diagrams should describe interactions whereas state diagrams should describe internal behaviors
  - Traditional sequence diagrams often step into internal details
  - Traditional state diagrams are low-level, just as traditional sequence diagrams are, only more so
- Our state diagrams apply to a *social* state, which can be affected through messages described by sequence diagrams
- Consider state diagrams as describing the progression of the social state of a service engagement
  - We can express this from an outside, i.e., a public or an institutional, as opposed to an implementation perspective
  - A research challenge is to ensure the social state remains sufficiently aligned across the interacting parties
  - For a properly designed service engagement, its social state ought to progress consistently