Combining pre-/post-conditions

- Standard interpretation
  - joining pre- and post-conditions conjunctively

context $C::op()$
pre: P_1 **post:** Q_1

context $C::op()$
pre: P_2 **post:** Q_2

$\rightsquigarrow$

context $C::op()$
pre: P_1 **and** P_2
post: Q_1 **and** Q_2

- Alternative interpretation
  - **case distinction** (like in protocol state machines)
  - only useful for pre-/post-condition pairs

context $C::op()$
pre: P_1 **post:** Q_1

context $C::op()$
pre: P_2 **post:** Q_2

$\rightsquigarrow$

context $C::op()$
pre: P_1 **or** P_2
post: (P_1@pre **implies** Q_1)
and (P_2@pre **implies** Q_2)

Messages

```
context Subject::hasChanged()
post: observer^update(self)
```

in calls on hasChanged, some update message with argument `self` will have been sent to observer

```
context Subject::hasChanged()
post: observer^update(? : Subject)
```

the actual argument does not matter

```
context Subject::hasChanged()
post: let messages : Set(OclMessage) =
            observer^^update(? : Subject)
      in messages->notEmpty() and
         messages->forAll(m |
            m.result().oclIsUndefined() and
            m.hasReturned() and
            m.subject = self)
```

all those messages

result of message call

whether it has finished

its actual parameter value

Initial values and derived properties

- Initial values
  - fix the initial value of a property of a classifier

```
package Booking                          -- which package
  context Passenger::status              -- which property
  init: Status::Swallow                  -- initial value
endpackage
```

- { derived } properties
  - define how the value of a property is derived from other information

```
context Passenger::currentFlights : Sequence(Flight)
derive: self->collect(booking)
            ->select(date = today()).flight->asSequence()
```

Query bodies and model features

- Bodies of { query } operations
  - define the value returned by a query operation
  - can be combined with a precondition

```
context TravelHandling::delay() : Minutes
body: tsh.delay->sum()
```

- Definition of additional model features
  - defined for the context classifier

```
context TravelStageHandling
def: isEarly() : Boolean = self.delay < 0

context TravelHandling
def: someEarly() : Boolean = tsh->exists(isEarly())
```