# Analysis of Simple Communication Protocol (1)

## - Exercises on Specification & Verification -

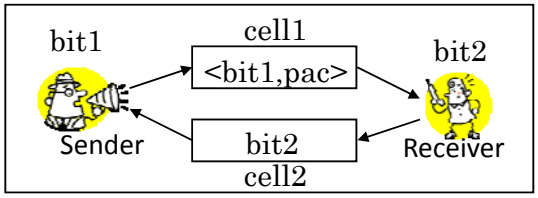
CafeOBJ Team of JAIST

# Roadmap

- Simple Communication Protocol (SCP)
- Modeling SCP
- Specification of Date Used
- Exercises on Specification & Verification

# Simple Communication Protocol (SCP)

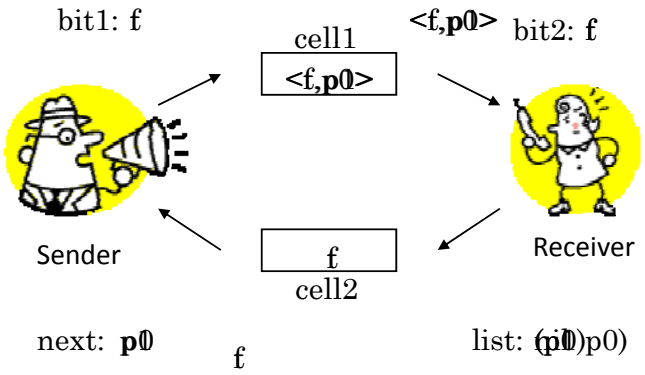
# Simple Communicationu Protocol




- Although ABP uses unreliable queues, SCP uses unreliable cells. Data in the cells can be lost.

  Initially, both cells are empty & both bits are the same.
- Sender & Receiver do the following:
  - Sender puts <bit1,pac> into cell1 repeatedly.
  - Receiver puts bit2 into cell2 repeatedly.
  - When Sender gets a bit $b$ from cell2, if $b$ does not equal bit1, Sender selects the next packet and alternates bit1.
  - When Receiver gets <$b$,$p$> from cell1, if $b$ equals bit2, Receiver receives $p$ and alternates bit2.

# Animation






# One Desirable Property

- When Receiver receives the $n$th packet,
  - Receiver has received the $n$+1 packets $\mathrm{p}0$, …, $\mathrm{p}n$ in this order,
  - each $\mathrm{p}i$ for $i = 0,\ldots,n$ has been received only once, and
  - no other packets have been received.
- The property is called
  the *reliable communication property*
  in this talk.

# Modeling SCP

# Observations

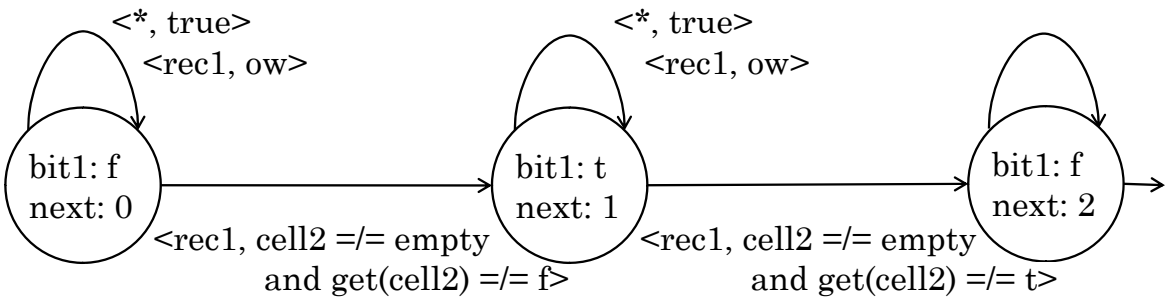
- Sender-to-Receiver channel
  ```
  bop cell1 : Sys -> PCell
  ```
- Receiver-to-Sender channel
  ```
  bop cell2 : Sys -> BCell
  ```
- Sender's bit
  ```
  bop bit1  : Sys -> Bool
  ```
- Receiver's bit
  ```
  bop bit2  : Sys -> Bool
  ```
- The ordinal of the packet sent next by Sender
  ```
  bop next  : Sys -> Nat
  ```
- The packets received by Receiver
  ```
  bop list  : Sys -> List
  ```

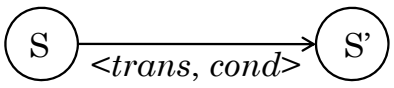
# Transitions

- Sender's sending pairs of bits & packets
  ```
  bop send1 : Sys -> Sys
  ```
- Sender's receiving bits
  ```
  bop rec1  : Sys -> Sys
  ```
- Receiver's sending bits
  ```
  bop send2 : Sys -> Sys
  ```
- Receiver's receiving pairs of bits & packets
  ```
  bop rec2  : Sys -> Sys
  ```
- Dropping the content of cell1
  ```
  bop drop1 : Sys -> Sys
  ```
- Dropping the content of cell2
  ```
  bop drop2 : Sys -> Sys
  ```



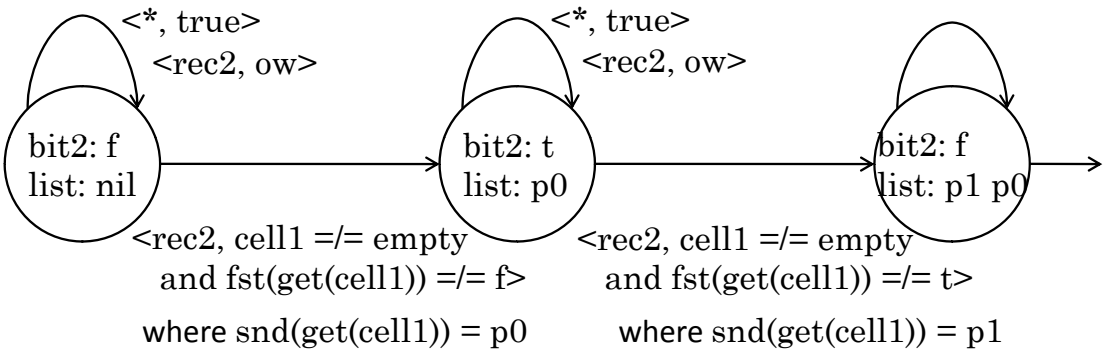
# Transition Diagram of Sender




- If the condition *cond* holds in the state S, then the transition trans can change S to S'.
- <*trans*,ow> means that if any other conditions for *trans* do not hold, *trans* can change S to S'.
- * represents any transition except those explicitly stated.

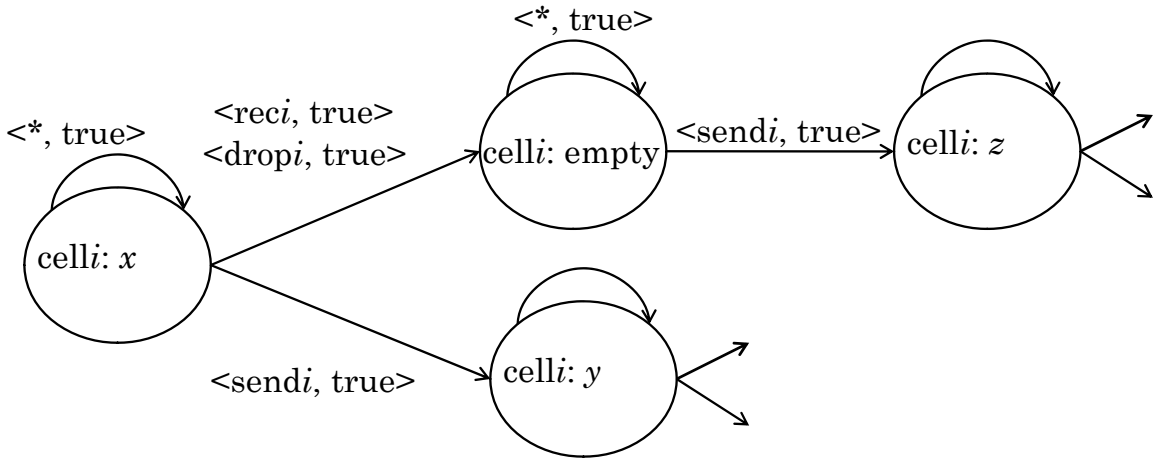
# Transition Diagram of Receiver




fst(< *e1*,*e2*>) = *e1*
snd(<*e1*,*e2*>) = *e2*



# Transition Diagram of Channels



# Specification of Data Used

# Data Used

- Boolean values for bits
- Natural numbers for ordinals of packets
- Packets
- Pairs of Boolean values & packets
- Cells (for channels) of pairs of BVs & pacs
- Cells (for channels) of Boolean values
- List of packets

# Data Modules

- Modules

  EQBOOL, PNAT, PACKET, PAIR, CELL, LIST, PACKET-LIST, BOOL-PACKET-PAIR, BOOL-CELL, BOOL-PACKET-PAIR-CELL, EQTRIV
- Views

  EQTRIV2PACKET, EQTRIV2EQBOOL, EQTRIV2BOOL-PACKET-PAIR

Let us take a look at the file "scp.mod".



# Exercises on Specification & Verification

# Exercises

1. Write the module SCP in which the model of SCP is specified.
2. Make some experiments based on the specification.
3. Verify that SCP satisfies the reliable communication property by writing proof scores.