Homework 1

- The R-multicast algorithm satisfies validity, since a correct process will eventually *B-deliver* the message to itself.

- The algorithm satisfies integrity, because of

  (1) the integrity property of the underlying communication channels

  (2) the fact that duplicates are not delivered.

What about agreement? It follows because... HOMEWORK! :-)

Homework 2

TOPIC 2 (28%): Multicast Communication

Let us consider ordered multicast in distributed systems. Well known ordering delivery requirements are *total ordering*, *causal ordering* and *FIFO ordering*.

PROBLEM 2.1 (6%)

For each of the three ordering delivery requirements, define the delivery requirement and draw an example of computation in which the requirement is met.

PROBLEM 2.2 (12%)

For each of the following 6 statements indicate whether the statement is true or false. In the latter case (that is, the statement is false), justify your answer by means of a counterexample.

Statement 1 (2%): Total ordering implies FIFO ordering.

Statement 2 (2%): FIFO ordering implies total ordering.

Statement 3 (2%): Total ordering implies causal ordering.

Statement 4 (2%): Causal ordering implies total ordering.

Statement 5 (2%): FIFO ordering implies causal ordering.

Statement 6 (2%): Causal ordering implies FIFO ordering.

PROBLEM 2.3 (10%)

Show informally that if two processes use a FIFO-ordered variant of B-multicast, then the totally ordered multicast is also causally ordered.

Homework 3

FIFO ordered multicast can be achieved by means of sequence numbers. A key assumption of the basic FIFO multicast algorithm (informally sketched in Figure 2) is that processes are organized in non-overlapping groups.

- A process p has variables:
  - S^p_g : how many messages p has sent to group g
  - R^q_g : sequence number of the latest message p has delivered from process q that was sent to group g
- For p to *FO-multicast* a message to group g:
  it piggy backs the value S^p_g onto the message;
  it *B-multicasts* the message to g;
  $S^p_g = S^p_g + 1$.
- Upon a receipt of a message from q bearing the seq. number S, p checks:
  IF ($S = R^q_g + 1$) **THEN** it *FO-delivers* the message, setting $R^q_g := S$.
  ELSIF ($S > R^q_g + 1$) **THEN**
  it places the message in its *hold-back queue* until
  the intervening messages have been delivered and
  $S = R^q_g + 1$.

Figure 2: Basic FIFO Multicast Algorithm

Homework 3 (cont.)

Question 1: Show that the FIFO multicast algorithm does not work for overlapping groups.

Question 2: Adapt the FIFO multicast algorithm to work for this case, that is considering also overlapping groups of processes.

Homework 4

- Suggest how to adapt the causally ordered multicast protocol to handle overlapping groups.

Homework 5

- Let us consider the reliable multicast algorithm. Explain why reversing the order of lines 11-12 makes the algorithm no longer satisfy agreement.

```
1   On initialization
2       Received := {};
3
4   For process p to R-multicast message m to group g
5       B-multicast(g, m);          // p ∈ g is included as a destination
6
7   On B-deliver(m) at process q with g = group(m)
8       if (m ∉ Received)
9       then
10                  Received := Received ∪ {m};
11                  if (q ≠ p) then B-multicast(g, m); end if
12                  R-deliver m;
13      end if
```