

Distributed Systems, Summer Term 2020

Exercise Sheet 1

1. Schedules

Consider three nodes, v_1 , v_2 , and v_3 , which are connected via FIFO channels, that is, messages between any two nodes are received in the same order they are sent. For example, if node v_1 sends first message m_1 then m_2 to node v_2 , then v_2 will first receive m_1 and then m_2 .

Devise **one** possible schedule S which is consistent with the following local restrictions to the three nodes.

- $S|1 = s_{1,3} s_{1,3} r_{1,2} r_{1,3} s_{1,2} r_{1,2} s_{1,3}$,
- $S|2 = s_{2,3} s_{2,1} r_{2,1} s_{2,1}$,
- $S|3 = r_{3,2} r_{3,1} s_{3,1} r_{3,1} r_{3,1}$.

$s_{i,j}$ denotes the send event from node i to node j and $r_{j,i}$ denotes the event that node j receives a message from node i .

Sample Solution

A solution can be obtained by drawing the diagram as in the lecture. One possible schedule is the following

$s_{2,3} s_{1,3} r_{3,2} r_{3,1} s_{2,1} s_{1,3} s_{3,1} r_{1,2} r_{1,3} r_{3,1} s_{1,2} r_{2,1} s_{2,1} r_{1,2} s_{1,3} r_{3,1}$.

2. The Level Algorithm

Consider the following algorithm between two connected nodes u and v :

The two nodes maintain levels ℓ_u and ℓ_v , which are both initialized to 0. One round of the algorithm works as follows:

1. Both nodes send their current level to each other
2. If u receives level ℓ_v from v , u updates its level to $\ell_u := \max\{\ell_u, \ell_v + 1\}$. If the message to node u is lost, node u does not change its level ℓ_u . Node v updates its level ℓ_v in the same (symmetric) way.

Argue that if the level algorithm runs for r rounds, the following properties hold:

- a) At the end, the two levels differ by at most one.
- b) If all messages succeed, both levels are equal to r .
- c) The level of a node is at least 1 if and only if the node received at least one message.

Sample Solution

- a) We show via induction on the number of rounds that after each round, the two levels differ by at most one. For a round r , let ℓ_u^r and ℓ_v^r be the levels of nodes u and v after round r . For $r = 0$ we have $\ell_u^0 = 0 = \ell_v^0$. Now assume that the statement holds after round r , i.e., we have $\ell_v^r - 1 \leq \ell_u^r \leq \ell_v^r + 1$. We have

$$\ell_u^{r+1} \leq \max\{\ell_u^r, \ell_v^r + 1\} \stackrel{\text{I.H.}}{=} \ell_v^r + 1 \leq \ell_v^{r+1} + 1$$

where the last inequation holds because levels can only increase.

Analogously, we prove $\ell_v^{r+1} \leq \ell_u^{r+1} + 1$.

- b) Induction on the number of rounds: At the beginning (after round 0), we have $\ell_u^0 = \ell_v^0 = 0$. Now assume $\ell_u^r = \ell_v^r = r$ and in round r both messages succeed. Then $\ell_u^{r+1} = \max\{\ell_u^r, \ell_v^r + 1\} = \max\{r, r + 1\} = r + 1$ and $\ell_v^{r+1} = \max\{\ell_v^r, \ell_u^r + 1\} = r + 1$.
- c) If a nodes never receives a message, it never updates its level (which is initially 0). So if its level is at least one, it must have received a message. On the other hand, if node u receives $\ell_v \geq 0$ in some round, its level becomes $\ell_v + 1 \geq 1$ which never decreases again.

3. (Variations) of Two Generals

In the lecture we considered the (deterministically unsolvable) **Two Generals** consensus problem:

- two deterministic nodes, synchronuous communication, unreliable messages,
- **input**: 0 or 1 for each node,
- **output**: each node needs to decide either 0 or 1,
- **agreement**: both nodes must output the same decision (0 or 1),
- **validity**: if both nodes have the same input $x \in \{0, 1\}$ and no messages are lost, both nodes output x ,
- **termination**: both nodes terminate in a bounded number of rounds.

In this exercise we consider three modifications of the model. For each of them, either give a (deterministic) algorithm or state a proof which shows that the variation cannot be solved deterministically.

- a) There is the guarantee that within the first 7 rounds at least *one* message in *each* direction succeeds.
- b) There is the guarantee that within the first 7 rounds at least *one* message succeeds (note that nodes are not allowed to stay silent).
- c) Let $k \in \mathbb{N}$ be a natural number. The input for each node is a number $x_i \in \{0, \dots, k\}$.

Goal: If no message gets lost *and* both have the same input $x \in \{0, \dots, k\}$, both have to output x . In all other cases the nodes should output numbers which do not differ by more than one. The algorithm still has to terminate in a finite number of rounds.

Hint: This last problem is solvable. You can use the level algorithm from task 2.

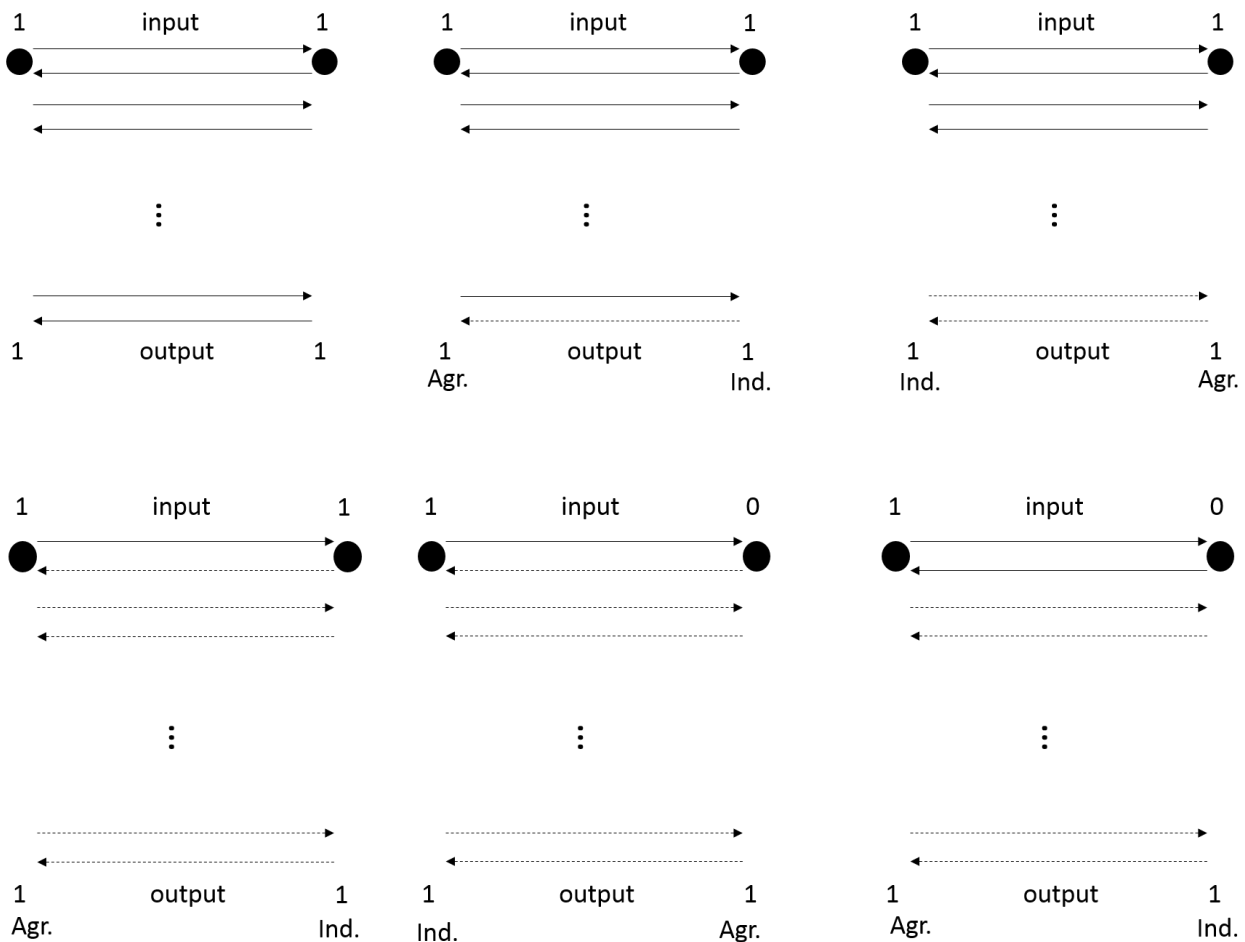
Sample Solution

- a) As there is the guarantee that at least one message in *each* direction succeeds every node sends its value repeatedly for seven rounds. At least one time it will reach the other node. So both nodes know both values and can decide on a output by a previously fixed algorithm, e.g., output $val_1 \cdot val_2$.
- b) The problem is not solvable for two deterministic generals. Assume that it is solvable in T rounds. By a sequence of executions we show that both nodes need to have the same output in the following two executions.
- E : both inputs are 1, all messages are delivered
 - E' : both inputs are 0, all messages are delivered

Because of validity both nodes need to output 1 in the case of E and 0 in the case of E' , a contradiction.

The proof is similar to the proof in the lecture - one only has to be careful in the last step to always keep at least one successfully delivered message.

Important is that we drop at most one message at a time to keep the two following executions similar (see definition of similar in the lecture). The reasoning is always that two similar executions are indistinguishable for one node and thus he has to output the same value in both execution. Then, to fulfill agreement all nodes need to output the same in both executions.



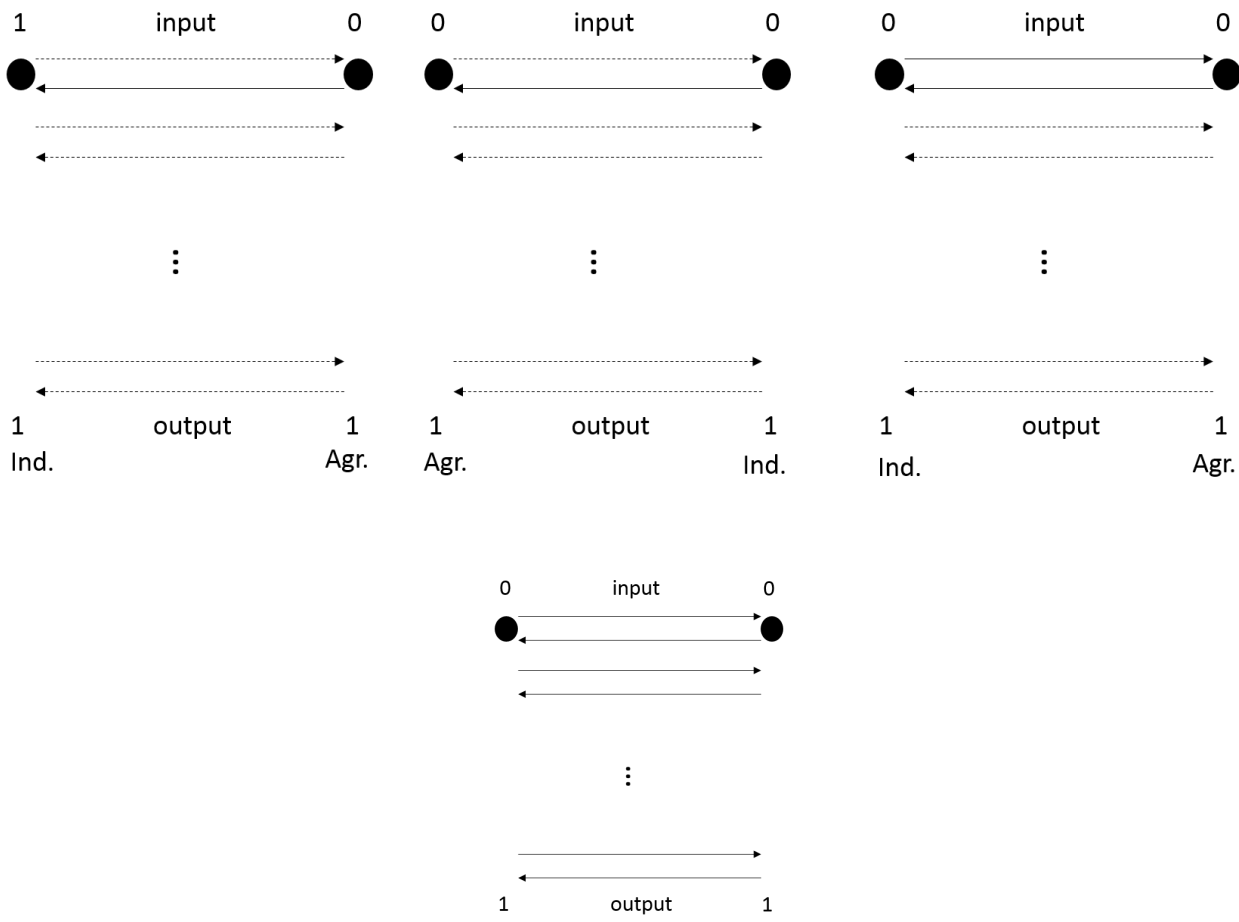


Figure 1: We begin with the execution E and show that it is indistinguishable from E' via a long chain of indistinguishable executions.

- c) The problem is solvable with the help of the level algorithm from the lecture. Let x_u and x_v be respectively the input of node u and the input of node v . The algorithm is the following: each node runs the Level algorithm for as many rounds as the value they have in input, and then they output their level.
- Case 1: $x_u = x_v = x$. In this case, both nodes will run the Level algorithm for x many rounds. We distinguish two cases.
 - (a) If no messages are lost, by exercise 2 b we know that both levels are equal to x , hence the output of both nodes will be x , as desired.
 - (b) If there are lost messages, by exercise 2 a we know that the nodes' levels differ by at most one, hence their output will differ by at most one, as desired.
 - Case 2: $x_u \neq x_v$. In this case, one of the nodes will run for more rounds than the other. Notice that this is analogous to the case where both nodes have the same input but some messages are lost, and for this case we proved above that the outputs differ by at most one, as desired.