# I2CN exercise sheet 2

Dimitri Tabatadze ~ 2025-03-19

## 1 Formalizing multisets

Formalize $S \in \mathbb{M}(\mathcal{M})$ as a mapping

$$S : \mathcal{M} \to \mathbb{N} \tag{1}$$

where $S(m) = n$ means, that message $m$ is in a multiset $S$ exactly $n$ times.[1] How do you formalize for multisets $S$ and $T$ then

1. $S$ contains the single message $m$
2. $S \cup T$
3. $S \cap T$
4. $S \setminus T$

### Solution

1. $$S(x) = \begin{cases} 1 & \text{if } x = m \\ 0 & \text{if } x \neq m \end{cases} \tag{2}$$
2. $$(S \cup T)(x) = S(x) + T(x) \tag{3}$$
3. $$(S \cap T)(x) = \min\{S(x), T(x)\} \tag{4}$$
4. $$(S \setminus T)(x) = \max\{0, S(x) - T(x)\} \tag{5}$$

## 2 Simulating asynchronous communication by synchronous communication

We denote components $X$ of the system with asynchronous communication as on the slides and their counterparts (if existing) in the simulating system with synchronous communication by $\tilde{X}$.

- As the set of states we choose:

  $$\tilde{Z}_p = Z_p \times \mathbb{M}(\mathcal{M}) \tag{6}$$

  i.e. we include the simulated message buffer in the state of the simulating process.
- for a send transition:

  $$c \to_p^s (x, m, d) \text{ with } q = link_p(x) \tag{7}$$

  and corresponding receive transition

[1] Recall that we denote with $\mathbb{N}$ the natural numbers including 0.

$$(e,m) \rightarrow_q^r f \tag{8}$$

of the simulated system we choose for all $S, T$

$$(S,c) \tilde{\rightarrow}_p^s (x,m,S) \text{ and } ((e,T),m) \tilde{\rightarrow}_q^r (f, T \cup \{m\}) \tag{9}$$

Specify for the simulating system

1. the send transition $\tilde{\rightarrow}_p^s$.
2. the internal transition $\tilde{\rightarrow}_p^i$.

Hint: you need rules for the simulated internal step and for the simulated receive steps.

3. State a simulation theorem between computations of the two systems. Hint: don't forget to couple the start configurations of the system in the simulation relation.

## Solution

1. For the send transition, we will have:

$$(c,S) \tilde{\rightarrow}_p^s ((x,S),m,d) \iff c \rightarrow_p^s (x,m,d) \tag{10}$$

2. For the receive transition, we will have:

$$(a,S) \tilde{\rightarrow}_p^i (b,S) \iff a \rightarrow_p^i b \tag{11}$$

and

$$((e,S),m) \tilde{\rightarrow}_p^r (f, S \cup \{m\}) \iff (e,m) \rightarrow_p^r f \tag{12}$$

# 3 Schedules

Consider the system from exercise 1 of sheet 1. Prove or disprove: in every run of the system and for every natural number $n$ we eventually have $n \in S$.

## Solution

This statement is false. Part 2 of exercise 2 of sheet 1 had us come up with a counterexample. But here it is again:

No matter what we do, the internal transition always picks 0 and puts it into $R$. This way, we'll always have $S \subset \{0,1\}$.

# 4 Norm Functions

Lemma 2 from the slides: There is a typo on the slides. The following condition is part of the definition of norm functions.

- if $E$ terminates in a state $\gamma$, the $P(\gamma)$ holds in that state.

Now prove: Let $f$ be a norm function for predicate $P$. Then for each execution of $E$ of system $S$ predicate $P(\gamma)$ holds in some configuration of $E$.

## Solution

*Proof*: We need to consider two cases:

- **Execution is finite.** In this case, there will be a configuration $\gamma$ in which the execution terminates and by the updated definition, the predicate will hold (i.e. $P(\gamma)$).
- **Execution is infinite.** In this case, we will have infinite configurations

  $$\gamma_1 \to \gamma_2 \to \cdots \tag{13}$$

  Applying the norm function to each gives us the following:

  $$\forall i > 1 \to f(\gamma_{i-1}) > f(\gamma_i) \vee P(\gamma). \tag{14}$$

  Since $(W, <)$ is a well ordering, there will always exist some $j$ where $f(\gamma_{j-1}) \not> f(\gamma_j)$ and therefore $P(\gamma_j)$.

□

# 5 Making the leader known

Modify the deterministic leader election algorithm with UIDs on a ring, such that

- Computations on all nodes terminate (not necessarily at the same time)
- Each node knows the UID of the elected leader at the time when its computation has terminated.

For ring size $N$ estimate

1. the total number of messages sent
2. the number of steps until the last node terminates.

## Solution

There will be two stages for the algorithm: finding the leader, announcing the leader. In the finding the leader stage, we will do everything exactly as we did it before. In the announcing the leader stage, the leader will send an announcement in both directions. After receive the announcement, each node will forward it and terminate immediately.

1. The total number of messages sent can vary very much.
   - **Minimum:** if the UIDs are assigned in an *ascending* order

     $$i < j \to u_i < u_j \tag{15}$$

     then the number of messages will be $N + N$.

- **Maximum:** if the UIDs are assigned in a *descending* order

  $$i < j \rightarrow u_j < u_i \tag{16}$$

  then the number of messages sent will be $\frac{N(N+1)}{2} + N$.

So the general answer is $O(N^2)$.

2. It will take $N$ rounds for the leader to receive their own UID and by that time the UID will have gone through every other node. However, for the other nodes to know to terminate, the leader should inform them somehow. This will take at least $\lceil \frac{N}{2} \rceil$ steps (that's the case where the leader sends termination signals in both directions). So in total $N + \lceil \frac{N}{2} \rceil$ or $O(N)$ steps should suffice.