Exercise 4, FoDS

Yannick Martin
Anastasia Markusheva

Task 1.)

1.) Yes, they need consensus algorithm. We thought it's maybe not needed if they continuously talk with each other. Then two generals can convince the third one. But since their radio communication is unreliable this is not given. Therefor they can't decide twice and they need all properties.

2) No, because of C3. In consensus it not allowed to decide twice. Therefore it's not possible for a distinct to decide with regular consensus, when the population want to change there meaning because of their neighbours

3) Yes, because all properties C_1 till C_4 are fullfilled. So at the end of the computation all processes proposes a model. And they take the best one at the end.

4) No not necessary. C_3 is not necessary They can convince a roomate that another person fits to them. Because hose at the end it's important to have a new roommate and not that every person is 100% happy.

Task 2:

C_1 Termination: This property is fullfilled since only q and r are correct process and both decided a value.

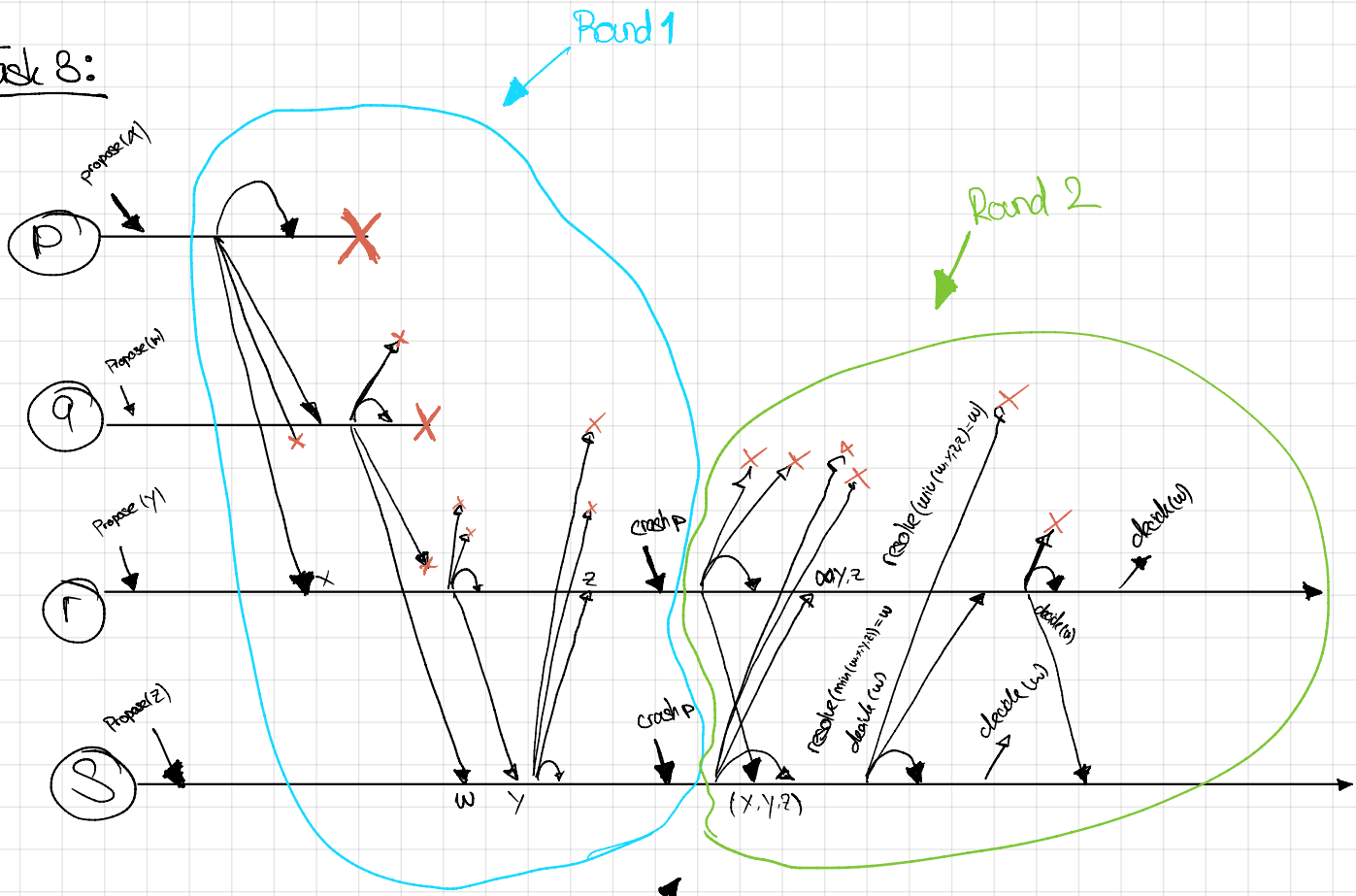
C_2 Validity: q and r decided both 1, which was proposed by process p. → Property fullfilled

C_3 Integrity: q and r decided only one, p didn't decide. There here no process decided twice. → Property fullfilled

C_4 Agreement: q and r decided both 1. Therefore no two correct process decides differently → Property fullfilled

If the process crash before he broadcastet his decision. With an asynchronous system the other process don't no if a process crashed or the message just takes his time.

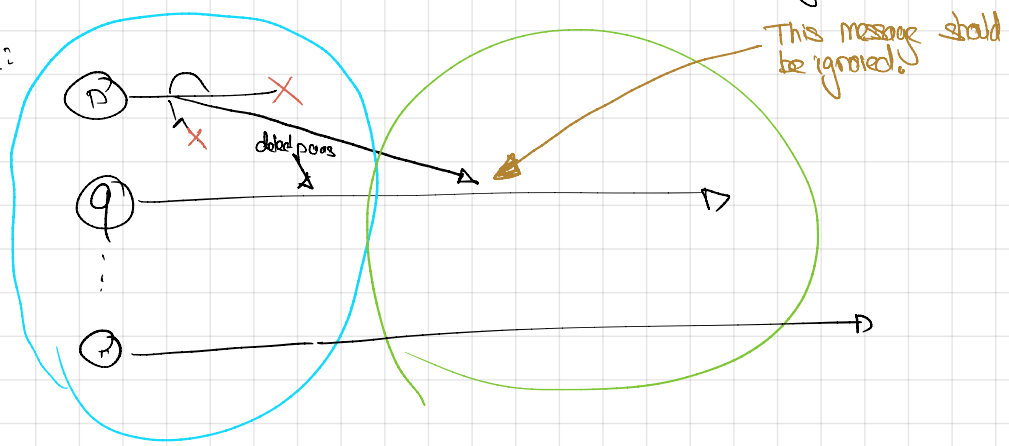
Task 8:



Since S received all (non-crashed) processes in S perspective, it can decide safely

4) The property that could be affected is C4. No two correct process decides differently. Therefore we have to make sure that a late proposal should be ignored.

Example:

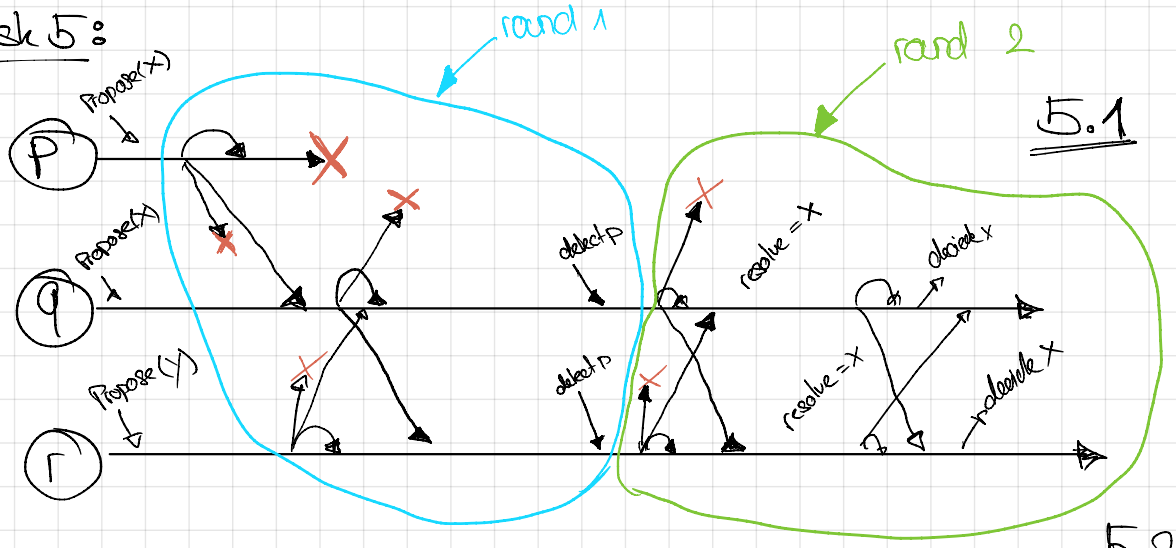


upon event ⟨beb, Deliver | p [Proposal, r, ps]⟩ such that r = round do
receivedfrom := receivedfrom ∪ {p};
proposalset := proposalset ∪ ps;

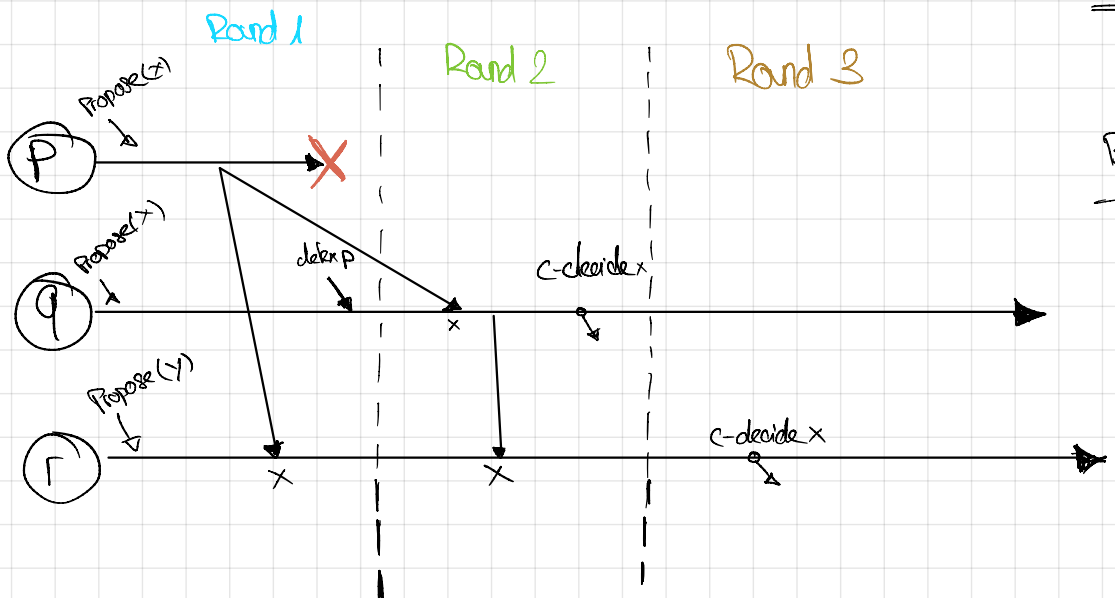
These lines of code prevent that we put late proposals in our set "receivedfrom" because we execute this lines only when r is equal to "round". And for late proposal r < round.

Task 5:

5.1



5.2



Book-Notation

Hierarchical Consensus doesn't need a minimum resolve function because of the rank function. Our proposal will be mapped from 1 to N and this value will be stored in "rank."
Now the processes decides for the proposal with the highest rank (if this process doesn't crash)

In general the flooding algorithm uses less communication steps than hierarchical one.
Since the flooding one only need N steps if N-1 process in sequence fail while for the hierarchical algorithm we need always N steps.

In Hierarchical Consensus we have $O(N^2)$ which is less than $O(N^3)$ for the flooding consensus.