Homework 1 Solution Sketch

These are not complete (formal) proofs, but should provide sufficient basis to develop a formal proof.

1. **Necessity**:

   Necessity of n >= 3f+1 was proved in class for binary inputs. That necessity extends to non-binary inputs as well.

   Now prove that n >= fm+1 is necessary. The proof is by contradiction. Suppose that n = fm (similar proof will apply for n < 3m as well). Consider the scenario in which all n processes are non-faulty, and exactly f processes have input i for each i in [0,m-1].

   Consider a process P whose input is 0. From its perspective. It is possible that all the processes with input equal to 1 are faulty – this is possible since exactly f processes have input 1. If this were the case, then process P must not choose 1 as its output, because none of the remaining processes have input 1. This argument can be applied at process P for each input that is not equal to 0. This implies that process P (with input 0) can only choose its output as 0.

   We can repeat this argument for each process, and show that each process may only choose its own input as the output. This implies that agreement cannot be achieved, proving that n > fm is necessary.

   Sufficiency:

   Sufficiency can be proved by providing an algorithm that works correctly. One such possible algorithm uses “Byzantine broadcast” as a primitive. Byzantine broadcast is performed by n processes wherein one process is designated as the “sender” (or “Commander” in the Byzantine Generals paper). The sender has an input.

   Byzantine broadcast ensures the following two properties: (a) If the sender is non-faulty, the output of each process is the sender’s input, and (b) all non-faulty processes produce the same output.

   Using Byzantine broadcast, we can now implement Byzantine consensus as follows:

   Step 1: Each process acts as sender in an instance of Byzantine broadcast to sends its input to all the processes. Correctness of Byzantine broadcast requires that n > 3f.

   Step 2: At the end of Step 1, each non-faulty process will have received n values, with the values corresponding to non-faulty processes being the inputs at those processes.

Step 3: Choose as output the value that occurs most frequently among the above n values, breaking ties using a deterministic rule.

Correctness of the above algorithm follows from the fact that at least one value will occur more than f times among the n values at Step 3 (because n > fm). Secondly, any value that occurs f+1 times must correspond to the input of at least one process (since there are at most f faulty processes).

2. n>=mf+1 is necessary and sufficient. Necessity argument is same as problem 1. Sufficiency is similar to problem 1 as well, except we do not need to use Byzantine broadcast, since the network supports broadcast functionality.

3. At least one round is necessary – otherwise, it should be easy to see that agreement cannot be achieved.

   One round suffices. In round 1, each process transmits its input on the broadcast channel. At the end of the round, each process chooses its output as the smallest value thus received.