CS5320 Theory Assignment - 3

Vishal Vijay Devadiga (CS21BTECH11061)

Question 1

Early-stopping algorithm for Consensus under Crash Failures

Modify the Consensus Algorithm for Crash Failures (synchronous system) to terminate within $f + 1$ rounds when the actual number of stop-failures (f_a) is lower the f.

The Algorithm gives a consensus algorithm for n processes, where up to f processes, where $f < n$, may fail in the fail-stop model. Here, the consensus variable x is integer-valued. Each process has an initial value x_i.

If up to f failures are to be tolerated, then the algorithm has f + 1 rounds.

Modify this algorithm to include an early stopping mechanism where the algorithm terminated when the actual number of stop failures f_a is less then f.

Solution

In the original algorithm, the process broadcasts the value of the consensus variable x only if the value of the consensus changed in the previous round. However, in the modified algorithm, the process will broadcast the value of the consensus variable x in every round.

After each round, the processes will check if the number of processes that have stopped is greater than f.

If the number of processes that have stopped is greater than f, then the process continues to the next round.

If the number of processes that have stopped is less than f, then the process terminates.

```
f <- maximum number of crash failures tolerated

x <- initial value of the consensus variable

For each process Pi (1 <= i <= n) do
  for round from 1 to f + 1 do:
```

```
    broadcast x
    Yj <- value received from process Pj
    x <- min(x, Yj)
    Fa <- number of processes that have stopped
    if Fa < f then
      break
  output x
```

Question 2

Generalizing the Consensus Problem with binary inputs to work with multi-valued inputs

Assume that you have a solution to the Consensus Problem problem that works with binary inputs.

Can you use this to solve the Consensus Problem to work with multi-valued inputs.

Solution

Suppose the limit of the multi valued input is m.

Then, the number of bits required to represent the input is $\log_2 m + 1$.

The binary input can be represented as a binary string of length $\log_2 m + 1$.

The binary input can be divided into $\log_2 m + 1$ parts, each of length 1.

Now, for each part, we can use the binary input consensus algorithm to get the consensus value.

The consensus value for the multi-valued input can be obtained by concatenating the consensus values of each part.

```
m <- maximum value of the multi-valued input
b <- ceil(log2(m))

for i from 1 to b do:
  x <- ith bit of the binary representation of the multi-valued input
  y <- binary input consensus algorithm(x)
  ans <- ans + y * 2^(i - 1)
output ans
```