Ch 14 – Consensus

- The problem and its relevance
- The Binary Consensus
- Generalization to multivalued Consensus
- Basic lower bounds

Def - Consensus

Definition 14.1 (Consensus)**.** *Each node $v \in V_g$ is given an input $x_v \in X$. To solve* Consensus, *an algorithm must compute output values $o_v \in X$ at all correct nodes $v \in V_g$ meeting the following conditions:*

- ***Agreement:*** *There is $o \in X$ so that $o_v = o$ for all $v \in V_g$. We refer to o as the output of the Consensus algorithm.*
- ***Validity:*** *If there is $x \in X$ so that for all $v \in V_g$ it holds that $x_v = x$, then $o = x$.*
- ***Termination:*** *There is $r \in \mathbb{N}$ satisfying that each $v \in V_g$ terminates and outputs o_v by the end of round r.*

The algorithm has round complexity *$R \in \mathbb{N}$, if it terminates in R rounds in all executions.*

If $X = \{0, 1\}$, we refer to the task as Binary Consensus, *and denote the input of node v by b_v (to indicate that it is a bit).*

The Phase King Alg

- Binary Consensus
- Leader per phase
- f+1 phases
- Each phase is composed of 3 broadcasts

The idea: once we have a correct leader a value will be set and will never be changed again

---

Algorithm 18 Phase King Algorithm at node $i \in V_g$. Note that for convenience the code assumes that i also receives its own broadcasts and all messages are consistent with the format required by the algorithm (i.e., invalid or missing messages by faulty nodes are replaced by valid default values).

```
op ← b_i
for j = 1 ... f + 1 do
    strong ← 0
    broadcast op                                          ▹ first broadcast
    if received at least n − f times op then
        strong ← 1
    end if
    if strong = 1 then
        broadcast op                                      ▹ second broadcast
    end if
    if received fewer than n − f times op then
        strong ← 0
    end if
    if i = j then                                         ▹ king's broadcast
        if received at least f + 1 times 0 then
            broadcast 0
        else
            broadcast 1
        end if
    end if
    if strong = 0 and received b ∈ {0, 1} from node j then
        op ← b                                            ▹ if not sure, obey the king
    end if
end for
return op
```

Proof 2

Lemma 14.15. *Fix a phase $j \in \{1, \ldots, f+1\}$. There is a $b \in \{0, 1\}$ satisfying that each $i \in V_g$ holding strong $= 1$ after the first broadcast of phase j has $op_i = b$.*

```
2: for j = 1 ... f + 1 do
3:     strong ← 0
4:     broadcast op                              ▹ first broadcast
5:     if received at least n − f times op then
6:         strong ← 1
7:     end if
```

Count how many different pairs (i, b_i) are sent in total in line 4

- Each $i \in V_g$ contributes a single pair (i, b_i), since they send the same to all
- Each $j \in V \setminus V_g$ may contribute up to two such pairs
- the total $\leq n-f+2f = n+f$
- if there are two different values in line 5 there are at least $2(n-f)$ pairs
- $2(n-f) \leq$ total number of pairs $\leq n+f$

which implies $n \leq 3f$ – a contradiction

Proof 3

Lemma 14.16. *Let phase $j \in \{1, \ldots, f+1\}$ satisfies that node $j \in V_g$. There is some $b \in \{0, 1\}$ so that $op_i = b$ for all $i \in V_g$ at the end of phase j.*

```
8:   if strong = 1 then
9:       broadcast op                                   ▹ second broadcast
10:  end if
11:  if received fewer than n − f times op then
12:      strong ← 0
13:  end if
14:  if i = j then                                      ▹ king's broadcast
15:      if received at least f + 1 times 0 then
16:          broadcast 0
17:      else
18:          broadcast 1
19:      end if
```

- All correct that broadcast in line 9, broadcast the same value
- Observe – if there is $v \in V_g$ with strong = 1 past line 12 only this will be the value with f+1 multiplicity in line 15 at the leader.
- This value will be sent to all in the leader's broadcast

Proof 3

Lemma 14.16. *Let phase $j \in \{1, \ldots, f+1\}$ satisfies that node $j \in V_g$. There is some $b \in \{0, 1\}$ so that $op_i = b$ for all $i \in V_g$ at the end of phase j.*

```
14:     if i = j then                                   ▹ king's broadcast
15:         if received at least f + 1 times 0 then
16:             broadcast 0
17:         else
18:             broadcast 1
19:         end if
20:     end if
21:     if strong = 0 and received b ∈ {0, 1} from node j then
22:         op ← b                                 ▹ if not sure, obey the king
23:     end if
24: end for
25: return op
```

- Again, if there is $v \in V_g$ with strong = 1 past line 12 this value will be adopted by all past line 22
- Otherwise, every correct adopts the value sent by the correct leader past line 22

Proof 4

Theorem 14.18. *Algorithm 18 solves Binary Consensus in the synchronous model. It runs for* $R(f) = 3(f+1) \in O(f)$ *rounds and correct nodes communicate exclusively by* 1*-bit broadcasts.*

- Once all correct hold the same value – that remains the only value
- all correct return that value - Agreement

Definition 14.1 (Consensus). *Each node* $v \in V_g$ *is given an input* $x_v \in X$. *To solve* Consensus, *an algorithm must compute output values* $o_v \in X$ *at all correct nodes* $v \in V_g$ *meeting the following conditions:*

- ***Agreement:*** *There is* $o \in X$ *so that* $o_v = o$ *for all* $v \in V_g$. *We refer to* o *as the output of the Consensus algorithm.*
- ***Validity:*** *If there is* $x \in X$ *so that for all* $v \in V_g$ *it holds that* $x_v = x$, *then* $o = x$.
- ***Termination:*** *There is* $r \in \mathbb{N}$ *satisfying that each* $v \in V_g$ *terminates and outputs* o_v *by the end of round* r.

Ch 14 – Consensus

- The problem and its relevance
- The binary Consensus
- Generalization to multivalued Consensus
- Basic lower bounds

Reducing Consensus to Binary Consensus

- The idea: reduce to 2 possible values then use the Binary Algorithm
- Correct nodes will proceed with a non-defalt value only if it may be the cse that all correct has it as an input value
- The faulty nodes will do their best to confuse the correcct nodes

Multi Value Consensus

Algorithm 19 Consensus algorithm for input set X based on a Binary Consensus algorithm. The code is for node $i \in V_g$. For convenience, we assume that nodes also receive their own messages and that all received messages not adhering to the used format are replaced by valid default values.

$c \leftarrow 0$ ▷ default output value, w.l.o.g. $0 \in X$
broadcast x_i ▷ first broadcast
if received at least $n - f$ times x_i **then**
    $c \leftarrow x_i$ ▷ all correct nodes might have this input
end if
$b \leftarrow 0$ ▷ input value for binary instance
broadcast c ▷ second broadcast
if received at least $n - f$ times $c' \in X \setminus \{0\}$ **then**
    $c \leftarrow c'$ ▷ there can be at most on such c'
    $b \leftarrow 1$ ▷ c' is known to all nodes
else if received at least $f + 1$ times $c' \in X \setminus \{0\}$ **then**
    $c \leftarrow c'$ ▷ there can be at most on such c'
end if
participate in binary consensus instance with input b
if output is 1 **then**
    return c ▷ can only happen if everyone knows c
else
    return 0
end if

Multi Value Consensus

Algorithm 19 Consensus algorithm for input set X based on a Binary Consensus algorithm. The code is for node $i \in V_g$. For convenience, we assume that nodes also receive their own messages and that all received messages not adhering to the used format are replaced by valid default values.

$c \leftarrow 0$ ▹ default output value, w.l.o.g. $0 \in X$
broadcast x_i ▹ first broadcast
if received at least $n - f$ times x_i **then**
    $c \leftarrow x_i$ ▹ all correct nodes might have this input
end if

- First stage: Reducing the possible values
- c is non 0 only if there is vast support to a different value
  - similar to previous arguments we saw no two correct will

- Thus, by the end of this stage all correct hold either 0
  or some other specific value c. Two possible values

Multi Value Consensus

$c \leftarrow 0$ ▷ default output value, w.l.o.g. $0 \in X$
broadcast x_i ▷ first broadcast
if received at least $n - f$ times x_i **then**
    $c \leftarrow x_i$ ▷ all correct nodes might have this input
end if
$b \leftarrow 0$ ▷ input value for binary instance
broadcast c ▷ second broadcast
if received at least $n - f$ times $c' \in X \setminus \{0\}$ **then**
    $c \leftarrow c'$ ▷ there can be at most on such c'
    $b \leftarrow 1$ ▷ c' is known to all nodes
else if received at least $f + 1$ times $c' \in X \setminus \{0\}$ **then**
    $c \leftarrow c'$ ▷ there can be at most on such c'
end if

- Finalizing the reduction: you continue with b=1 only if all correct know what is the alternative value (c')
- if any correct have b=1, every correct sees at least f+1 support to the value and only to that value

Multi Value Consensus

$c \leftarrow 0$ ▷ default output value, w.l.o.g. $0 \in X$
broadcast x_i ▷ first broadcast
if received at least $n - f$ times x_i **then**
    $c \leftarrow x_i$ ▷ all correct nodes might have this input
end if
$b \leftarrow 0$ ▷ input value for binary instance
broadcast c ▷ second broadcast
if received at least $n - f$ times $c' \in X \setminus \{0\}$ **then**
    $c \leftarrow c'$ ▷ there can be at most on such c'
    $b \leftarrow 1$ ▷ c' is known to all nodes

- The result of the Binary Consensus determines whether we output 0 or c.

participate in binary consensus instance with input b
if output is 1 **then**
    return c ▷ can only happen if everyone knows c
else
    return 0
end if

Multi Value Consensus - proof

Lemma 14.19. *If the Binary Consensus algorithm called in Algorithm 19 satisfies validity, so does Algorithm 19.*

1: $c \leftarrow 0$ ▷ default output value, w.l.o.g. $0 \in X$
2: broadcast x_i ▷ first broadcast
3: **if** received at least $n - f$ times x_i **then**
4: $\quad c \leftarrow x_i$ ▷ all correct nodes might have this input
5: **end if**
6: $b \leftarrow 0$ ▷ input value for binary instance
7: broadcast c ▷ second broadcast
8: **if** received at least $n - f$ times $c' \in X \setminus \{0\}$ **then**
9: $\quad c \leftarrow c'$ ▷ there can be at most on such c'
10: $\quad b \leftarrow 1$ ▷ c' is known to all nodes
11: **else if** received at least $f + 1$ times $c' \in X \setminus \{0\}$ **then**
12: $\quad c \leftarrow c'$ ▷ there can be at most on such c'
13: **end if**

If all correct start with the same input value:

- all set it in line 4 and 9
- all enter the Binary Consensus with b=1
- all output that value

Multi Value Consensus - proof

Lemma 14.20. *If $n > 3f$, there is at most one value $c \in X \setminus \{0\}$ sent by correct nodes in the second broadcast.*

1: $c \leftarrow 0$ ▹ default output value, w.l.o.g. $0 \in X$
2: broadcast x_i ▹ first broadcast
3: **if** received at least $n - f$ times x_i **then**
4: $\quad c \leftarrow x_i$ ▹ all correct nodes might have this input
5: **end if**
6: $b \leftarrow 0$ ▹ input value for binary instance
7: broadcast c ▹ second broadcast

If all correct start with the same input value:

- all set it in line 4 and 9
- all enter the Binary Consensus with b=1
- all output that value

Multi Value Consensus - proof

Lemma 14.21. *If $n > 3f$ and the Binary Consensus algorithm called in Algorithm 19 satisfies agreement and validity, Algorithm 19 satisfies agreement.*

7: broadcast c ▷ second broadcast
8: **if** received at least $n - f$ times $c' \in X \setminus \{0\}$ **then**
9: $\quad c \leftarrow c'$ ▷ there can be at most on such c'
10: $\quad b \leftarrow 1$ ▷ c' is known to all nodes
11: **else if** received at least $f + 1$ times $c' \in X \setminus \{0\}$ **then**
12: $\quad c \leftarrow c'$ ▷ there can be at most on such c'
13: **end if**
14: participate in binary consensus instance with input b
15: **if** output is 1 **then**
16: $\quad$ **return** c ▷ can only happen if everyone knows c
17: **else**
18: $\quad$ **return** 0
19: **end if**

- By the previous lemma in line 7 every correct broadcasts either 0 or c.
- if all correct hold b=0 - done
- if there is a correct with b=1, all correct hold an identical c value
- thus, either all output c or 0.

Multi Value Consensus - proof

Theorem 14.22. *Suppose we are given a fully connected network G of n nodes and a Binary Consensus algorithm $\mathcal{A}$ for it that tolerates $f < \frac{n}{3}$ Byzantine faults. Then Algorithm 19 is a Consensus algorithm on G for inputs from X that tolerates f faults. In addition to calling $\mathcal{A}$ once as a subroutine, it runs for $2\left\lceil \frac{\log |X|}{B} \right\rceil$ rounds, during which nodes broadcast messages of size B; here, $B \in \mathbb{N}_{>0}$ can be chosen freely.*

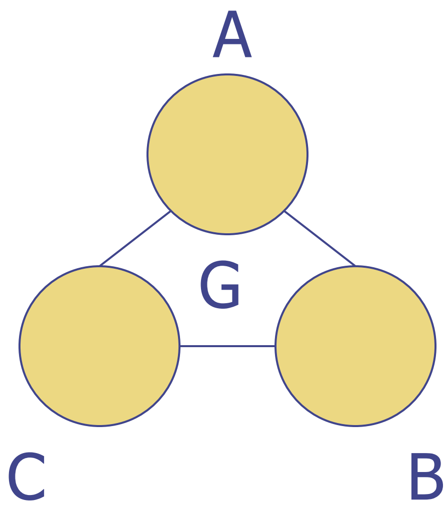
Ch 14 – Consensus

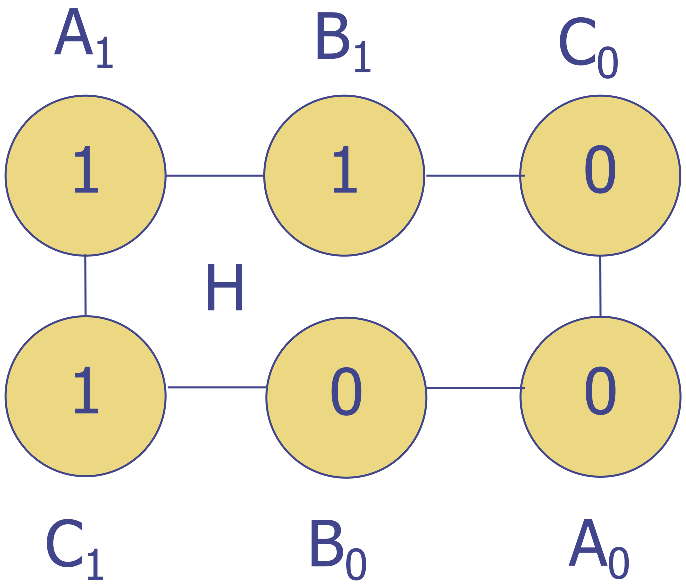
- The problem and its relevance
- The binary Consensus
- Generalization to multivalued Consensus
- Basic lower bounds

Theorem 14.29. *If $3 \leq n \leq 3f$, Consensus with Byzantine faults cannot be solved.*

n < 3f+1 lower bound

Theorem 14.29. *If $3 \leq n \leq 3f$, Consensus with Byzantine faults cannot be solved.*

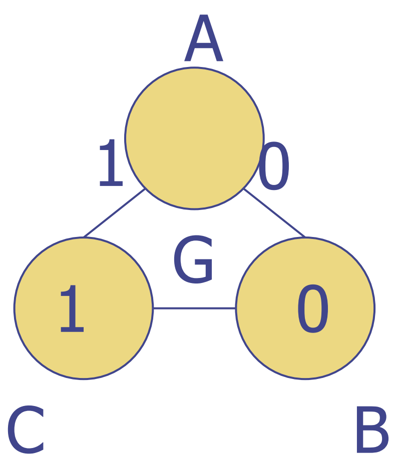


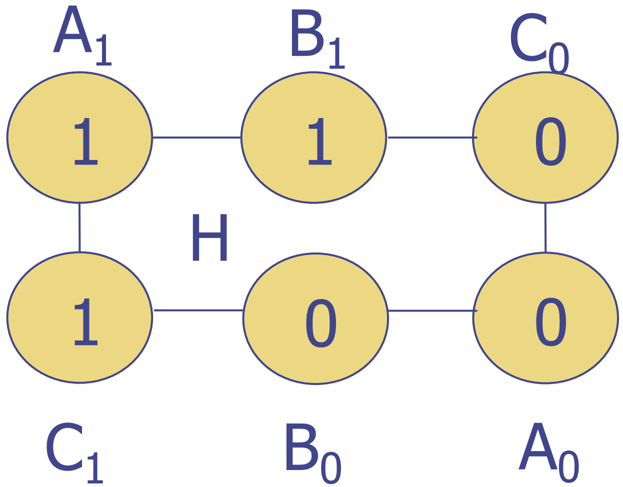


- Divide G to 3 sets of size up to f each.
- Assume there is an algorithm $\mathcal{A}$ that ensures consensus on G for any inputs.
- Construct H as described.
- We can execute $\mathcal{A}$ on H – since each node in H sees the same structure as in G. It's state machine determined by $\mathcal{A}$ functions the same in both graphs.
- E will be the execution on H with the specific inputs (it may detect an error)

n < 3f+1 lower bound

Lemma 14.24. *For $k \in \{0, 1\}$, consider the copies B' and C' of B and C, respectively, that in Figure 14.2 appear after A_k in clockwise direction (e.g., for $k = 0$, $B' = B_0$, $C' = C_0$). Then there is an execution $\mathcal{A}_{A_k}$ of $\mathcal{A}$ on G with fault set A such that the unique node with label i in $B' \cup C'$ cannot distinguish $\mathcal{E}$ from $\mathcal{E}_{A_k}$ at i.*

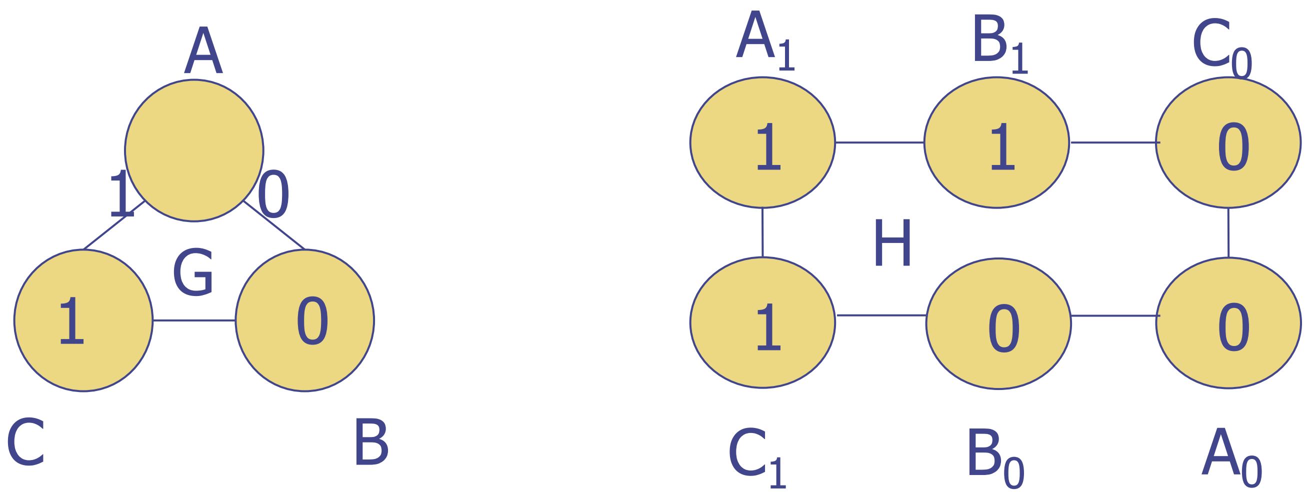




- By induction on the round number
- r=0 : input.
- r=1: first message exchange
- Assuming r clearly r+1 hold.
- **the arguments hold for any pair of adjacent nodes**.

n < 3f+1 lower bound

Corollary 14.27. *In $\mathcal{E}$, each node terminates and outputs the same value as it would output with $\mathcal{A}$ on G.*

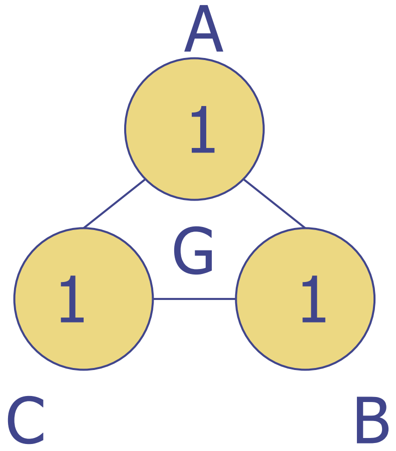


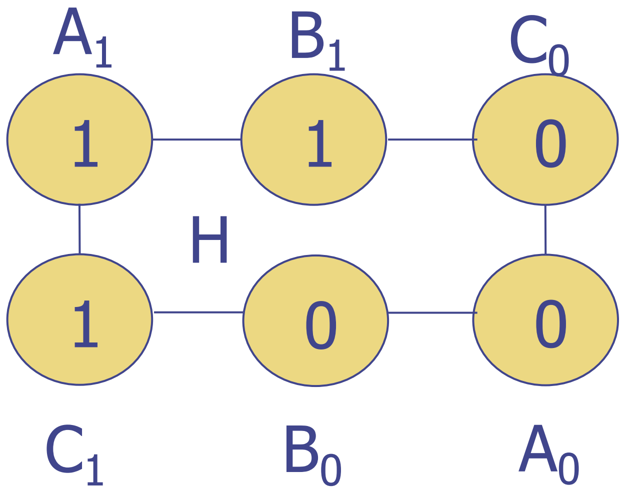
- Since nodes in any pair can't tell the difference their state machines produce the same output in H as it would in G

- Thus, specifically both C_1 and B_0 output the same value, say b

n < 3f+1 lower bound

Corollary 14.28. *In $\mathcal{E}$, nodes in A_0 output 0, while nodes in A_1 output 1.*

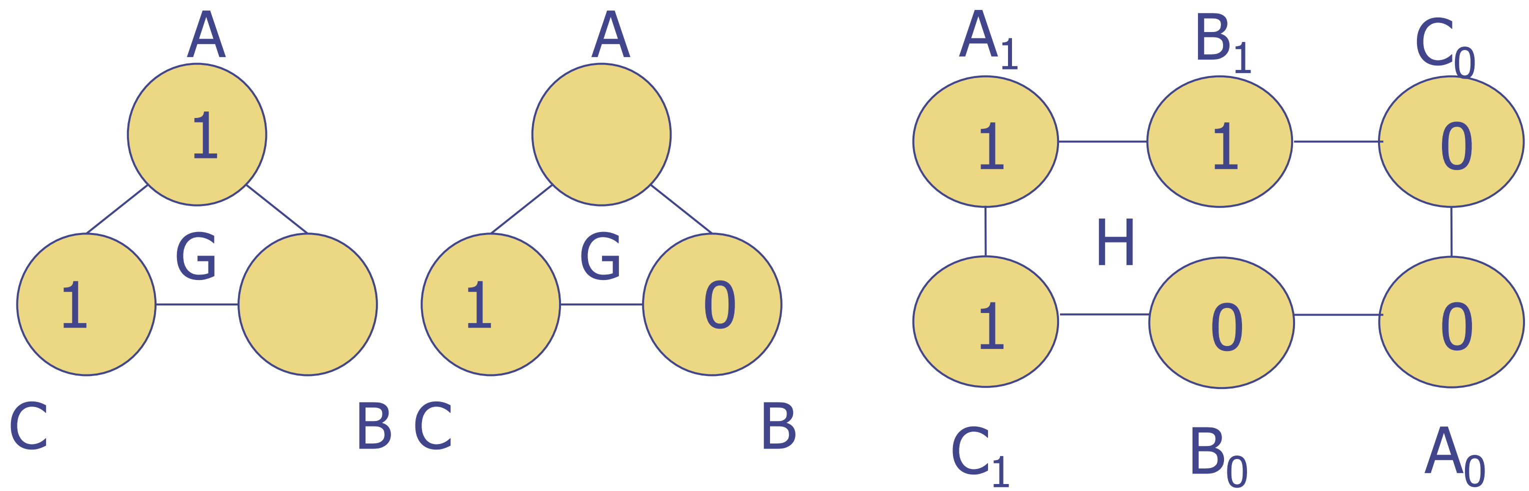




- A_i should output "i" since it's sate machine is consistent with the case in which the other correct also starts with input "i"

n < 3f+1 lower bound

Theorem 14.29. *If $3 \le n \le 3f$, Consensus with Byzantine faults cannot be solved.*



- As we claimed before, both C_1 and B_0 output the same value, say b
- Assume b=0.
- But, the left graph shows that both A_1 and C_1 should output "1"
- A contradiction