S

$s_0 \in S$

$s_0, s_1, s_2, s_3 \ldots\ldots s_n \ldots$

Discrete system evolving over time

$S = \mathbb{N}$

State:

$c : \mathbb{N}$

$s_0,$ $c = 1, c = 2, c = 3 \ldots$

$c = 0, c = 0, c = 0 \ldots$

Could be any sequence of natural numbers

Init:

$c = 0$

Implementable specification?

Step:

$$((c' = c + 1) \wedge (c < 9))$$
$$\vee$$
$$((c' = c - 1) \wedge (c > 0))$$

These are not functions, but predicates!

Current state

next state

$c = 0, c = 1, c = 0, c = 1, c = 0 \ldots$

$c = 0,\ c = 1,\ c = 2,\ c = 1,\ c = 0$

$c = 0,\ c = 1,\ c = 2,\ c = 3,\ c = 2,\ c = 3 \ldots$

CP: $\forall_i : 0 <= c^i <= 9$

Axiomatic formulation

Every behaviour here satisfies this correctness predicate but how do we prove it?

We can do this via induction, plugging in some values and proving that it always stays <= 9.

$C = \{r, g, b\}$

State:

$O : 2^C$

$O \subseteq C$

$O^0 = \{\}$

$O^1 = \{r\}$

$O^2 = \{b\}$

This isn't valid under the correctness predicate of consensus, as we CHANGE the value we chose

CT: $\forall_{i,j} :: \forall_{c_1, c_2} \in C, \in O^i \wedge c_2 \in O^j \Rightarrow c_1 = c_2$

$\{r\}$ $\{r\}$

$\emptyset$ $\{b\} \cup \emptyset$

Once chosen, we must never UNCHOOSE

$\{r\}$ $\{r, b\}$ a color

$\{\}, \{r\}, \{\} \ldots \{r\}, \{r\} \ldots\ldots$

This is valid as we don't necessarily require that we ALWAYS have a value chosen. This is akin to having all replicas down, we have an EMPTY LOG, which is why going from non empty → empty is valid.

Consensus

$I \quad O = \emptyset$

$S: \quad O' = O \vee (O = \emptyset \wedge O' \subseteq C \wedge |O'| = 1)$

Voting

$P = \{P_1, P_2, P_3\}$

$O \equiv \{c \in C \mid |\{\ldots\}| >= 2\}$

State:

$\{p \in P \mid c \in V_p\}$

$V_{P_1}, V_{P_2}, V_{P_3}$ $\quad V_p \subseteq C$

↳ the set of processes that voted for this color is >= 2

Init: $V_p = \{\}$

Step: $V_p' = V_p \vee (V_p \in \emptyset \wedge V_p' \subseteq C \wedge |V_p'| = 1)$

- Consensus is a property of the entire system and it may be achieved without any component in the system realizing it

- However, if $\{\{r\}, \{g\}, \{b\}\}$ is chosen then the outcome is $\{\}$, but the system is stuck! How does this get resolved?
- Perhaps one way to resolve is the autocratic system?

Step: $V_0 = V_0'$

$$\vee \left(V_0 = \emptyset \wedge |V_0'| = 1 \right)$$

$$V_1 = V_1'$$

$$\vee \left(V_1 = \emptyset \wedge V_1' = V_0 \right)$$

Make sure all except p_0 vote for p_1's choice

- However there is still the case where we want to choose a NEW LEADER if the current one is indisposed... how?

- For every instance in time, for every process, we have a set of votes..

$V^{i}_{p,v}$

This is not a chronological timeline, could happen concurrently, before/after

	V_0	V_1	V_2	V_3	$\top$	...
p_0						
p_1						
p_2						

- What if you always try to NOT conflict with votes to our left? So we only care about a finite subset of the table.
- Moreover, you get another process to promise not to vote in any rounds to the left! If that replica has choosen a different one in the previous round, you ALSO propose the same.
- As a primary, talk to a majority we ask majority to abstain from voting in previous ballot. Also, we choose the value they chose in the highest ballot they voted for. By induction we rely on old values to be safe!
- As a primary, we choose a value that is SAFE in that round, so as to not get you into trouble.. So you make sure old ballots don't complete, by getting promises from a majority!