Synod Algorithm - Leslie Lamport

Acceptor state:

\[ \text{maxPrepare}, \text{initially } 0 \]
\[ \text{accNum}, \text{initially } 0 \]
\[ \text{accVal}, \text{initially } \bot \]

When Proposer X wants to write value \( v \) to replicated log:

Select proposal number \( n \) larger than any proposal number seen so far
Send \text{prepare}(n)\) to all acceptors

When Acceptor receives \text{prepare}(n)\) from Proposer X:

If \( n > \text{maxPrepare} \)
\[ \text{maxPrepare} = n \]
Send \text{promise}(n, \text{accNum}, \text{accVal})\) to proposer X

If Proposer X receives \text{promise} from majority of acceptors:

if every \text{promise} has \text{accVal} = \bot
\[ \text{val} = v \]
else
\[ \text{val} = \text{received accVal with highest accNum} \]
Send \text{accept}(n, \text{val})\) to all acceptors

If Acceptor receives \text{accept}(n, \text{val})\) from Proposer X

If \( n \geq \text{maxPrepare} \)
\[ \text{accNum} = n \]
\[ \text{accVal} = \text{val} \]
Send \text{ack}(\text{accNum}, \text{accVal})\) to Proposer X

If Proposer X receives \text{ack} from majority of acceptors:
Send \text{commit}(\text{accVal})\) to all acceptors

If Acceptor receives \text{commit}(\text{accVal})\):
write \text{accVal} to log
**Invariant/Properties for Paxos**

(P1) An acceptor must accept the first value it receives.

(P2) If proposal \((n,v)\) is chosen, every higher numbered proposal that is chosen also has value \(v\).

(P2a) If proposal \((n,v)\) is chosen, every higher numbered proposal accepted by any acceptor has value \(v\).

(P2b) If proposal \((n,v)\) is chosen, every higher numbered proposal issued by any proposer has value \(v\).

(P2c) For any \(v\) and \(n\), if proposal \((n,v)\) is issued, then there is a majority set \(S\) of acceptors such that either:

\(\begin{align*}
\text{a) } & \text{ no acceptor in } S \text{ has accepted any value or} \\
\text{b) } & \text{ } v \text{ is the value of the highest numbered proposal with number less than } n \text{ that was accepted by any acceptor in } S.
\end{align*}\)