Solving Consensus using $\Diamond W$
(a.k.a. The case of the rotating coordinator)

Actually…. $\Diamond S$!

The ingredients

- Asynchronous rounds
- In round $r$, only messages timestamped $r$ are sent and processed (expect for special DECIDE messages)
- Each process $p$ has an opinion $v_p \in \{0,1\}$
- Each opinion has a time of adoption $t_p$ (initially $t_p = 0$)
- Each round has a coordinator $c$ such that $c_{id} = (r \mod n) + 1$

One round, four phases

<table>
<thead>
<tr>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each process, including $c$, sends its opinion timestamped with $r$</td>
<td>$c$ waits for first $\left\lceil \frac{n}{2} \right\rceil$ opinions with timestamp $r$</td>
<td>Each $p$ waits for a suggestion, or for failure detector to signal $c$ is faulty</td>
</tr>
<tr>
<td>$c$ selects $v$, one of the most recently adopted opinions</td>
<td>$v$ becomes $c$'s suggestion for round $r$</td>
<td>If a suggestion is received, it is adopted: $v_p := v$; $t_p := r$. ACK to $c$</td>
</tr>
<tr>
<td>$c$ sends its suggestion to all</td>
<td></td>
<td>Otherwise, NACK to $c$</td>
</tr>
</tbody>
</table>

Phase 4

- $c$ waits for first $\left\lceil \frac{n}{2} \right\rceil$ responses
- If all ACKs, then $c$ decides on $v$
- $c$ sends DECIDE to all
- If $p$ receives DECIDE:
  - $p$ decides on $v$

Solving Consensus using $\Diamond S$

$v_p := \text{input bit}; r := 0; t_p := 0; \text{state}_p := \text{undecided}$

while $\text{state}_p \neq \text{decided}$ do

- $r := r + 1$
- $c := (r \mod n) + 1$

[Phase 1: all processes send estimates to current coordinator]
- $p$ sends $(c, r, v_p, t_p)$ to $c$

[Phase 2: current coordinator gathers a majority of estimated opinions]
- $c$ waits for first $\left\lceil \frac{n}{2} \right\rceil$ opinions $(q, r, v_q, t_q)$
- If suggestion is received, then $t_q := r$. ACK to $c$
- $c$ selects among them the value $v_q$ with the largest $t_q$
- $c$ sends $(c, r, v_q)$ to all

[Phase 3: all processes wait for new suggestions from the current coordinator]
- $p$ waits until suggestion $(c, r, v)$ arrives or $c \in S$
- If suggestion is received, then $t_q := r$. ACK to $c$
- Else $p$ sends $(r, \text{NACK})$ to $c$

[Phase 4: coordinator waits for a majority of replies. If majority adopted the coordinator's suggestion, then coordinator sends request to decide]
- $c$ waits for first $\left\lceil \frac{n}{2} \right\rceil$ responses $(r, \text{ACK})$ or $(r, \text{NACK})$
- If receives $\left\lceil \frac{n}{2} \right\rceil$ ACKs, then $c$ sends $(c, \text{DECIDE})$ to all
- When $p$ delivers $(r, \text{DECIDE})$, then $p$ decides $v$; $\text{state}_p := \text{decided}$
Validity

- The value decided upon must have been suggested by the coordinator in some round.
- A coordinator suggests a value only by selecting it among the participants’ opinions.

From the algorithm, it is clear that each opinion corresponds to a value proposed by some process.

### The Locking Lemma - I

**Input:** \( r, x, \varepsilon \) \( y \), \( \phi \), \( y \) undecided while \( y \) is not decided.

\[
\begin{align*}
\text{Input: } \langle r, x, \varepsilon \rangle, \langle y \rangle, \phi, y \text{ undecided while } y \text{ is not decided.} \\
\text{Output: } \langle r, x, y, \phi, y \rangle \text{ decided.}
\end{align*}
\]

**Proof:**

- **Phase 1:** All processes send estimates to the current coordinator.
  
  \[
  (\text{Phase 1: all processes send estimates to the current coordinator})
  \]

- **Phase 2:** The current coordinator gathers a majority of estimates.
  
  \[
  (\text{Phase 2: current coordinator gathers a majority of estimates})
  \]

- **Phase 3:** All processes wait for new suggestions from the current coordinator.
  
  \[
  (\text{Phase 3: all processes wait for new suggestions from the current coordinator})
  \]

- **Phase 4:** Coordinator waits for a majority of replies. If majority replies are received, the coordinator adopts the suggestion of a coordinator.
  
  \[
  (\text{Phase 4: coordinator waits for a majority of replies. If majority replies are received, the coordinator adopts the suggestion of a coordinator})
  \]

### The Locking Lemma - II

**Input:** \( r, x, \varepsilon \) \( y \), \( \phi \), \( y \) undecided while \( y \) is not decided.

\[
\begin{align*}
\text{Input: } \langle r, x, \varepsilon \rangle, \langle y \rangle, \phi, y \text{ undecided while } y \text{ is not decided.} \\
\text{Output: } \langle r, x, y, \phi, y \rangle \text{ decided.}
\end{align*}
\]

**Proof:**

- **Focus on the earliest round** \( r \) in which a majority of \( \langle c, \text{ACK} \rangle \) have been sent to the coordinator of \( r \).
- **Consider the largest time** \( t \) such that \( r \) is decided by \( c \) in Phase 2 of round \( r \) or \( \phi \).
- **Consider the largest time** \( t \) such that \( r \) is decided by \( c \) in Phase 2 of round \( r \) or \( \phi \).
- **Let \( v \) be the value suggested by c in Phase 2 of round \( r \).**
- **Locking Lemma:** For all rounds \( r \) such that \( r \) suggests \( \phi \) and \( v \) is decided, \( v = v \).

Agreement

- **Strong Agreement:** All processes that decide, decide the same value.

**Proof:**

- **Trivially true** if no process decides.
- **If some process decides, it has delivered \( r, \langle \text{DECIDE}, v \rangle, \phi \) from a coordinator.
- **That coordinator has received a majority** of \( \langle c, \text{ACK} \rangle \).
- **Focus on the earliest round** \( r \) in which a majority of \( \langle c, \text{ACK} \rangle \) have been sent to the coordinator of \( r \).
- **Let \( v \) be the value suggested by c in Phase 2 of round \( r \).**

**Locking Lemma:** For all rounds \( r \), if a coordinator \( c \) sends \( v \), then \( v = v \).

**Proof (continued):**

- **Consider t, the largest time** of adoption collected by \( c \) in round \( r \).
- **Suppose process p sent \( p, \langle q, \varepsilon \rangle, \phi \).**
- The coordinator of round \( r \) sent its suggestion in Phase 2 of round \( t \), where \( r < t \).
- **By the Induction Hypothesis, that coordinator sent \( v \).**
- **Then, \( c \) sets \( v \) to \( v \).**

The decision value is locked as soon as a majority of processes sends an ACK back.
**Agreement**

All processors that decide, decide on the locked value.

Proof:
- Suppose process \( p \) delivers \((r^*, \text{DECIDE}, v)\).
- The coordinator for round \( r^* \) has sent \((r^*, \text{DECIDE}, v)\) in Phase 4 of round \( r^* \).
- To do so, the coordinator must have received a majority of \( (r^*, \text{ACK}) \) in Phase 4 of round \( r^* \).
- Recall that \( r \) is the earliest round in which a majority of \((r, \text{ACK})\) have been sent to a round’s coordinator.
- Clearly, \( r = r^* \).
- By the Locking Lemma, coordinator of round \( r^* \) suggested (and later decided) on the locked value.

**Termination**

- No correct process remains blocked forever at one of the wait statements.
- Eventually, a correct coordinator succeeds in sending \( (\cdot, \text{DECIDE}, \cdot) \).
- If some correct process decides, eventually all correct processes decide on the same value by Agreement.