Algorithm 1 Multi-Paxos: Prepare Phase

Implements:

AbortableSequenceConsensus, **instance** *asc*.

Uses:

FIFOPerfectPointToPointLinks, **instance** *fpl*.

```
upon event ⟨ $asc, Init$ ⟩ do
    $t := 0$;                                   ▷ logical clock
    $prepts := 0$;                              ▷ acceptor: prepared timestamp
    $(ats, av, al) := (0, \langle\rangle, 0)$;  ▷ acceptor: timestamp, accepted seq, length of decided seq
    $(pts, pv, pl) := (0, \langle\rangle, 0)$;  ▷ proposer: timestamp, proposed seq, length of learned seq
    $proposedValues := \langle\rangle$;         ▷ proposer: values proposed while preparing
    $readlist := [\bot]^N$;
    $accepted := [0]^N$;     ▷ proposer's knowledge about length of acceptor's longest accepted seq
    $decided := [0]^N$;      ▷ proposer's knowledge about length of acceptor's longest decided seq

upon event ⟨ $asc, Propose \mid v$ ⟩ do
    $t := t + 1$;
    if $pts = 0$ then
        $pts := t \times N + rank(self)$;
        $pv := prefix(av, al)$;
        $pl := 0$;
        $proposedValues := \langle v\rangle$;
        $readlist := [\bot]^N$;
        $accepted := [0]^N$;
        $decided := [0]^N$;
        for all $p \in \Pi$ do
            trigger ⟨ $fpl, Send \mid p, [\text{PREPARE}, pts, al, t]$ ⟩;
    else if $\#(readlist) \leq \lfloor N/2 \rfloor$ then
        $proposedValues := proposedValues + \langle v\rangle$;   ▷ append to sequence
    else if $v \notin pv$ then
        $pv := pv + \langle v\rangle$;
        for all $p \in \Pi$ such that $readlist[p] \neq \bot$ do
            trigger ⟨ $fpl, Send \mid p, [\text{ACCEPT}, pts, \langle v\rangle, \#(pv) - 1, t]$ ⟩;

upon event ⟨ $fpl, Deliver \mid q, [\text{PREPARE}, ts, l, t']$ ⟩ do
    $t := max(t, t') + 1$;
    if $ts < prepts$ then
        trigger ⟨ $fpl, Send \mid q, [\text{NACK}, ts, t]$ ⟩;
    else
        $prepts := ts$;
        trigger ⟨ $fpl, Send \mid q, [\text{PREPAREACK}, ts, ats, suffix(av, l), al, t]$ ⟩;

upon event ⟨ $fpl, Deliver \mid q, [\text{NACK}, pts', t']$ ⟩ do
    $t := max(t, t') + 1$;
    if $pts' = pts$ then
        $pts := 0$;
        trigger ⟨ $asc, Abort$ ⟩
```

Algorithm 2 Multi-Paxos: Accept Phase

```
upon event $\langle\ \mathit{fpl}, \mathit{Deliver} \mid q, [\textsc{PrepareAck}, \mathit{pts}', \mathit{ts}, \mathit{vsuf}, l, t']\ \rangle$ do
    $t := \mathit{max}(t, t') + 1$;
    if $\mathit{pts}' = \mathit{pts}$ then
        $\mathit{readlist}[q] := (\mathit{ts}, \mathit{vsuf})$;
        $\mathit{decided}[q] := l$;
        if $\#(\mathit{readlist}) = \lfloor N/2 \rfloor + 1$ then
            $(\mathit{ts}', \mathit{vsuf}') := (0, \langle\rangle)$;
            for all $(\mathit{ts}'', \mathit{vsuf}'') \in \mathit{readlist}$ do
                if $\mathit{ts}' < \mathit{ts}'' \lor \big(\mathit{ts}' = \mathit{ts}'' \land \#(\mathit{vsuf}') < \#(\mathit{vsuf}'')\big)$ then
                    $(\mathit{ts}', \mathit{vsuf}') := (\mathit{ts}'', \mathit{vsuf}'')$;
            $\mathit{pv} := \mathit{pv} + \mathit{vsuf}'$;
            for all $v \in \mathit{proposedValues}$ such that $v \notin \mathit{pv}$ do
                $\mathit{pv} := \mathit{pv} + \langle v \rangle$;
            for all $p \in \Pi$ such that $\mathit{readlist}[p] \neq \bot$ do
                $l' := \mathit{decided}[p]$;
                trigger $\langle\ \mathit{fpl}, \mathit{Send} \mid p, [\textsc{Accept}, \mathit{pts}, \mathit{suffix}(\mathit{pv}, l'), l', t]\ \rangle$;
        else if $\#(\mathit{readlist}) > \lfloor N/2 \rfloor + 1$ then
            trigger $\langle\ \mathit{fpl}, \mathit{Send} \mid q, [\textsc{Accept}, \mathit{pts}, \mathit{suffix}(\mathit{pv}, l), l, t]\ \rangle$;
            if $\mathit{pl} \neq 0$ then
                trigger $\langle\ \mathit{fpl}, \mathit{Send} \mid q, [\textsc{Decide}, \mathit{pts}, \mathit{pl}, t]\ \rangle$;

upon event $\langle\ \mathit{fpl}, \mathit{Deliver} \mid q, [\textsc{Accept}, \mathit{ts}, \mathit{vsuf}, \mathit{offs}, t']\ \rangle$ do
    $t := \mathit{max}(t, t') + 1$;
    if $\mathit{ts} \neq \mathit{prepts}$ then
        trigger $\langle\ \mathit{fpl}, \mathit{Send} \mid q, [\textsc{Nack}, \mathit{ts}, t]\ \rangle$;
    else
        $\mathit{ats} := \mathit{ts}$;
        if $\mathit{offs} < \#(\mathit{av})$ then
            $\mathit{av} := \mathit{prefix}(\mathit{av}, \mathit{offs})$;                    ▷ truncate sequence
        $\mathit{av} := \mathit{av} + \mathit{vsuf}$;
        trigger $\langle\ \mathit{fpl}, \mathit{Send} \mid q, [\textsc{AcceptAck}, \mathit{ts}, \#(\mathit{av}), t]\ \rangle$;

upon event $\langle\ \mathit{fpl}, \mathit{Deliver} \mid q, [\textsc{AcceptAck}, \mathit{pts}', l, t']\ \rangle$ do
    $t := \mathit{max}(t, t') + 1$;
    if $\mathit{pts}' = \mathit{pts}$ then
        $\mathit{accepted}[q] := l$;
        if $\mathit{pl} < l \land \#(\{p \in \Pi \mid \mathit{accepted}[p] \geq l\}) > \lfloor N/2 \rfloor$ then
            $\mathit{pl} := l$;
            for all $p \in \Pi$ such that $\mathit{readlist}[p] \neq \bot$ do
                trigger $\langle\ \mathit{fpl}, \mathit{Send} \mid p, [\textsc{Decide}, \mathit{pts}, \mathit{pl}, t]\ \rangle$;

upon event $\langle\ \mathit{fpl}, \mathit{Deliver} \mid q, [\textsc{Decide}, \mathit{ts}, l, t']\ \rangle$ do
    $t := \mathit{max}(t, t') + 1$;
    if $\mathit{ts} = \mathit{prepts}$ then
        while $\mathit{al} < l$ do
            trigger $\langle\ \mathit{asc}, \mathit{Decide} \mid \mathit{av}[\mathit{al}]\ \rangle$;          ▷ zero-based indexing
            $\mathit{al} := \mathit{al} + 1$;
```