

## Accept动作分析

根据TPaxos  $\Rightarrow$  EagerVoting的refinement mapping:

$\text{maxBal} \leftarrow \text{states}[p].\text{maxBal}$

并且TPaxos执行的Accept(p, b, v)动作对应的应该是EagerVoting中的VoteFor(a, b, v)动作:

---

$$\begin{aligned} \text{VoteFor}(a, b, v) &\triangleq \\ &\wedge \text{maxBal}[a] \leq b \text{ keep promise} \\ &\wedge \forall vt \in \text{votes}[a] : vt[1] \neq b \\ &\wedge \forall c \in \text{Acceptor} \setminus \{a\} : \\ &\quad \forall vt \in \text{votes}[c] : (vt[1] = b) \Rightarrow (vt[2] = v) \\ &\wedge \exists Q \in \text{Quorum} : \text{ShowsSafeAt}(Q, b, v) \text{ safe to vote} \\ &\wedge \text{votes}' = [\text{votes EXCEPT } ![a] = \text{votes}[a] \cup \{\langle b, v \rangle\}] \text{ vote} \end{aligned}$$

Accept(p, b, v)应该满足VoteFor的动作, 先分析第一条, 即转换成 $\text{states}[p].\text{maxBal} \leq b$ , 当p能进行b轮的accept阶段意味着该轮的prepare阶段已经进行过了, 即 $\text{states}[p].\text{maxBal} \geq b$ , 所以 $\text{states}[p].\text{maxBal} = b$ 。

对于第二条而言, 我们需要在Accept中限制b轮的Accept动作不能发生第二次, 是不是可以添加条件 $\text{states}[p].\text{maxVBal} \neq b$  即  $\text{states}[p].\text{maxVBal} < b$  (根据上面maxBal分析得出)。

添加了这两个条件后, 严格限制了Accept动作的执行时间, 当且仅当b对应的参与者通过了prepare请求并且它**没有对更高的编号make promise**。Accept阶段相当于Paxos中的P2a+一个P2b, 相对于Paxos的p2a阶段由于p2b加了限制。

对于原TPaxos算法而言, 文字描述的是经过了 $\text{issue}(m_i)$ 后等到了多数派的认可后可以进行 $\text{issue}(P_i)$ , 如果这两个动作之间发生了OnMessage并make promise, 能不能进行 $\text{issue}(P_i)$ ?