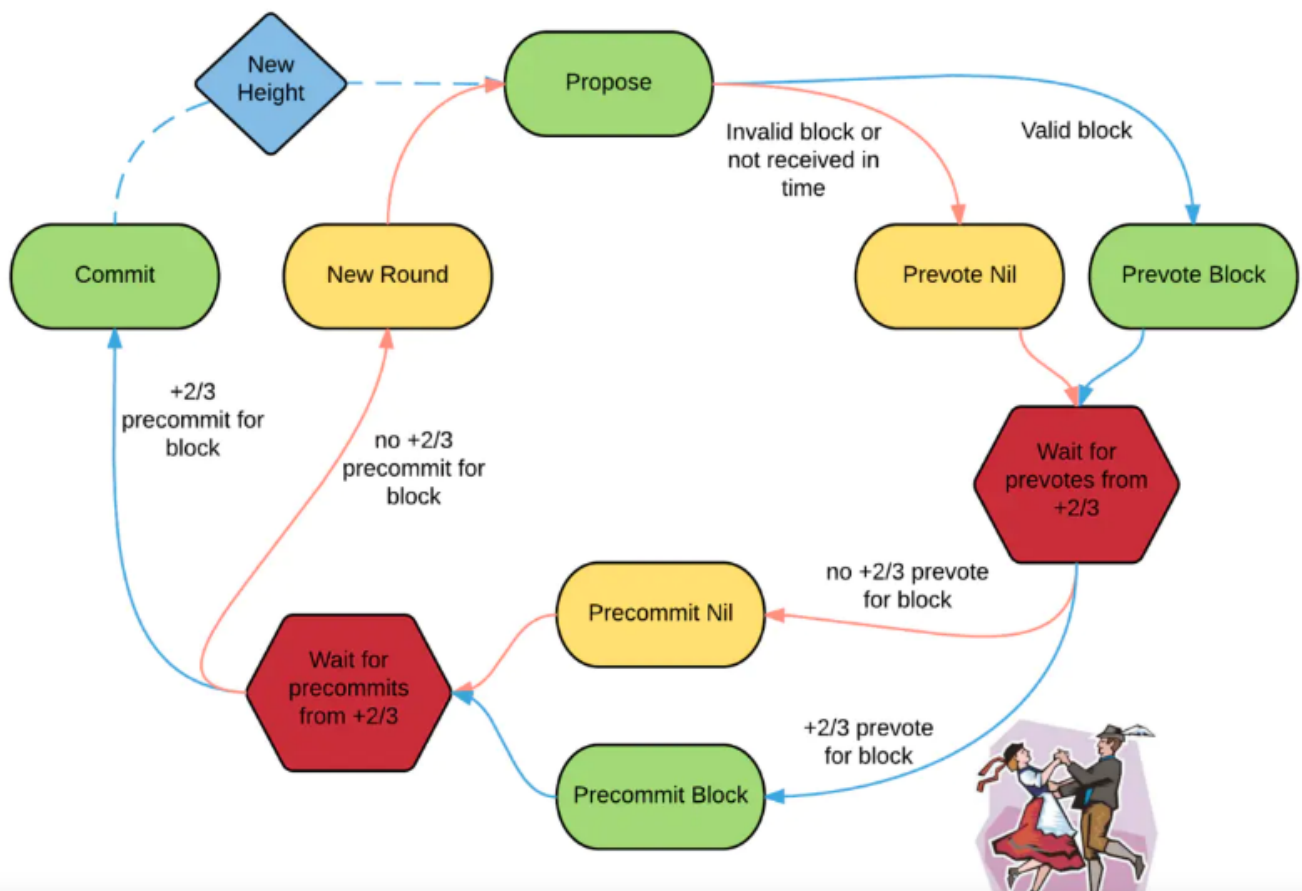


- 1. propose step
  - 1.1. upon entering propose
  - 1.2. Propose ends
- 2. Provote step
  - 2.1. upon entering prevote
  - 2.2. Provote ends
- 3. precommit step
  - 3.1. Upon entering Precommit, each validator broadcasts its precommit vote
  - 3.2. procommit ends
  - 3.3. commit step
- 4. NewHeight Step (height:H)



图中的**Round/Height**为直观意思，**step**指图中的每一步（**Proposal**、**PreVote**...）  
状态转换周期：

NewHeight -> (Propose -> Prevote -> Precommit)+ -> Commit -> NewHeight ->...

- prs: Peer Round State
- rs: Round State 首先看两个知识点：

1. 异步：系统中各个节点可能存在较大的时钟差异，同时消息传输时间是任意长的，各节点对消息进行处理的时间也可能是任意长的，这就造成无法判断某个消息迟迟没有被响应是哪里出了问题，且其中一个环节出问题导致整个共识过程无法达成

2. 由于异步的任意时长等待，导致某个验证节点会参与在不同的Round。（当消息无法响应，Step无法进行下去，Round没有终止，是如何出现不同的Round？，my解释：当前Round超时或处于等待时，自动开启下一Round）

Q:blockpart是什么？

A:将一个块分成几个部分？ block parts header that describes the valid block and is used to obtain all block parts 能够快速的发送区块，降低网络负载

## 1. propose step

---

客户端发起一笔交易gentx，通过rpc发到tendermint节点后进入mempool cache，tm调用checktx的abci接口验证交易，如果验证成功就进入mempool，proposer节点选择打包交易到区块中，并将区块附到proposalMsg中，对此msg进行签名，然后通过gossip把此msg广播到区块链网络中。

一个proposal包含一个(H,R)的block，和一个可选的最近的PoLC-Round < R，当然前提是proposer知道此可选参数（PoLC: proof of lock change，指的2/3+ prevote某个block），这样适当保住了共识网络的liveness。

### 1.1. upon entering propose

指定的proposer 提议一个block(H,R).

### 1.2. Propose ends

1. timeoutProposeR 超时，则进入-->Prevote(H,R)9（不过打包的是空区块）
2. 当收到了proposal block，并且改validator的所有prevote step 都在PoLC-Round，则进入-->Prevote(H,R)。（要理解这点，重点要理解某个validator有可能involve在若干个round，相当于说你其他轮中都过了prevote的step，才能进入新一轮的prevote step？但如果有锁的情况下即使进入了prevote轮，你也不能prevote了

## 2. Provote step

---

每个节点收到propose消息后，会先对区块进行验证，如果验证通过才开始prevote签名，然后广播到网络中。

### 2.1. upon entering prevote

1. First, if the validator is locked on a block since LastLockRound but now has a PoLC for something else at round PoLC-Round where LastLockRound < PoLC-Round < R, then it unlocks.（解锁条件，重点还是要理解异步）。
2. If the validator is still locked on a block, it prevotes that.（这条是因为锁的规则，他只能prevote自己lock住的block，但是会引起重复prevote？这个还得结合源码研究）。
3. Else, if the proposed block from Propose(H,R) is good, it prevotes that.（验证了proposal后正常prevote）
4. Else, if the proposal is invalid or wasn't received on time, it prevotes .（proposal无效，或者定时器超时，就prevote）

### 2.2. Provote ends

- After +2/3 prevotes for a particular block or . --> goto Precommit(H,R)
- After timeoutPrevote after receiving any +2/3 prevotes. --> goto Precommit(H,R)

## 3. precommit step

---

### 3.1. Upon entering Precommit, each validator broadcasts its precommit vote

1. If the validator has a PoLC at (H,R) for a particular block B, it (re)locks (or changes lock to) and precommits B and sets LastLockRound = R.(说明了什么时候上锁)
2. Else, if the validator has a PoLC at (H,R) for , it unlocks and precommits . (说明了什么时候precommit nil, 简言之, 就是获得了POLC的)
3. Else, it keeps the lock unchanged and precommits .

### 3.2. procommit ends

- After +2/3 precommits for . --> goto Propose(H,R+1) (注意, 并没有commit nil。prevote nil、precommit nil进入下一轮, 因此tendermint中没有类似PBFT中的viewchange)
- After timeoutPrecommit after receiving any +2/3 precommits. --> goto Propose(H,R+1) (我理解此处+2/3 precommits是笔误? 应该是+2/3 prevotes? )

### 3.3. commit step

1. Set CommitTime = now()
2. Wait until block is received. --> goto NewHeight(H+1) //注意此处要等到本轮的块到了才会进入NewHeight step.

## 4. NewHeight Step (height:H)

---

Move Precommits to LastCommit and increment height.

Set StartTime = CommitTime+timeoutCommit

Wait until StartTime to receive straggler commits. --> goto Propose(H,0)

1. 为什么会产生多个轮次? 多个轮次是并发过程吗? 安排多个轮次的作用是什么? 不同轮次打包的区块是同一交易还是不同交易?
2. 为什么要引入锁, 为什么不能直接根据区块ID投票? 锁的作用是什么? 什么时候上锁? 什么时候解锁?
3. 谁来统计投票结果和节点的数量? 如何获取投票结果?