# Fortnight Review

### Formal Specification （形式化验证）

The formal specifications are mathematically based techniques to prove the correctness of software algorithm. They describe a system in a more abstract and mathematical language. By using rigorous reasoning, formal specification can avoid subtle errors (like data race, deadlocks) in concurrent algorithms. The language we use is TLA+, proposed by Leslie Lamport.

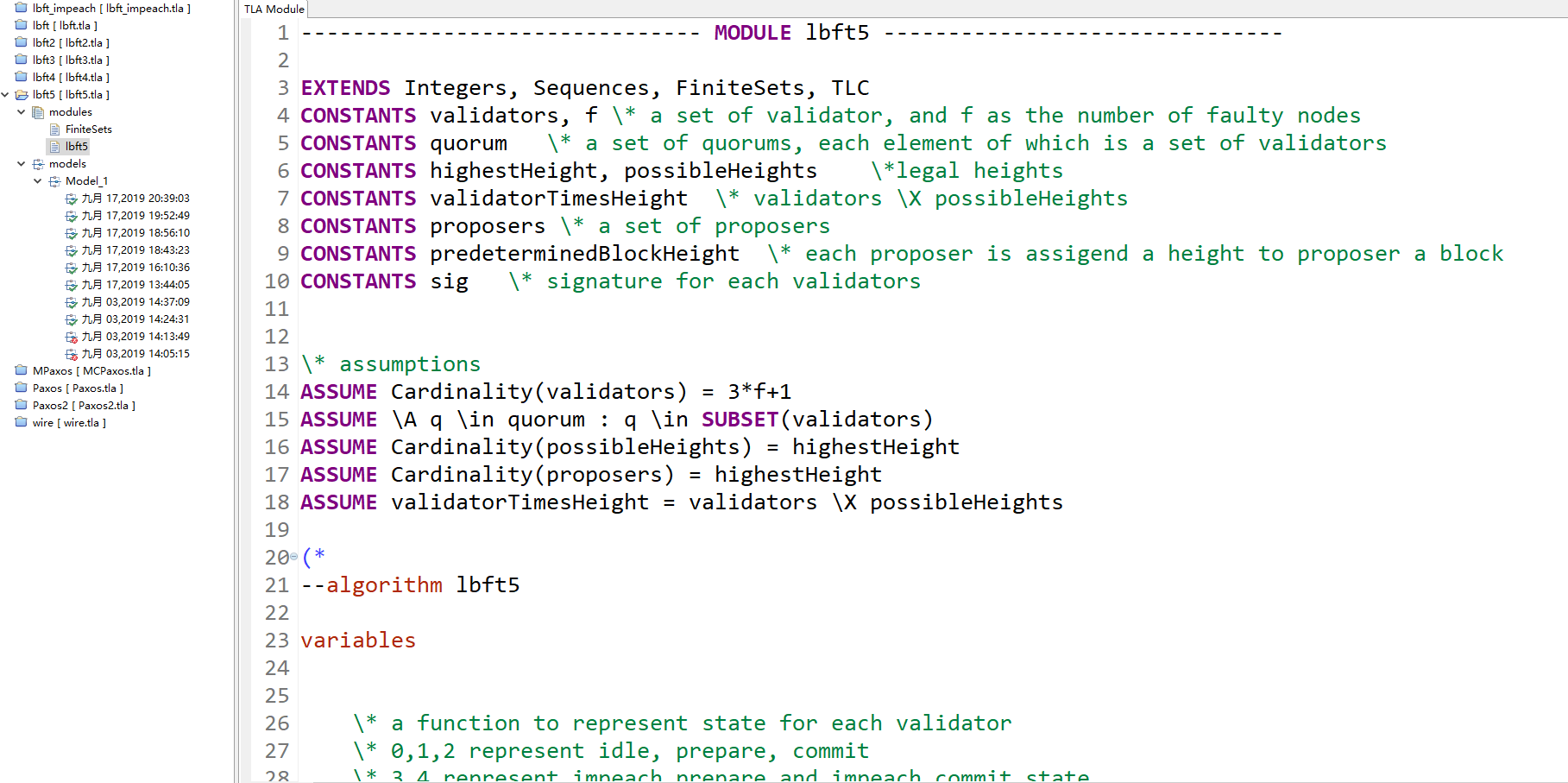
Our TLA+ code is mainly used for check LBFT2.0 consensus algorithm. There are three main objectives by running formal specification:

1. If there is deadlock in consensus algorithm;
2. If the validators can reach a consensus given a proposed block;
3. If there is only one possible consensus can be achieved;

The first and second objectives is to assure the **liveness** of the algorithm. In other word, the algorithm can terminate in a finite time. The third objective is to assure the **safety** of the algorithm, i.e., only one block is legal for a given block height.

Although the TLA+ code in our GitHub repo remains unchanged for several months, we have modified a lot in our local repository. Compared to the GitHub version, the newest version has

1. Constants and assumes to constrain the variables;
2. Concurrent processes to simulate concurrent threads;
3. Abstracted functions to avoid repeated codes;
4. Refined invariants to check safety of the algorithm.



Local TLA+ code for LBFT. Currently we are modifying the 5th version.