중간 보고서

There currently is a Utreexo parallel validation node implemented. However, I must note that it is currently in alpha stage with many improvements to make, both in the protocol level and the code level. As we go forward, we're looking to make major refactors to the current code.

There is currently one major protocol change that we're planning to make in the Utreexo accumulator code that will be backwards incompatible with the current nodes. The new algorithm that is in the works is to help with the proof sizes of the Utreexo proofs and enable better locality, helping with caching of the leaves. This new algorithm will help with both the proof size and caching, helping Utreexo parallel nodes sync even faster. This work is somewhat planned to be finished before the end of the Capstone project deadline, though it may take longer. This change does not effect the progress of the parallel nodes.

The second big change that's ahead of us is to refactor the code to be independent of the Bitcoin node that we forked off of. To implement the parallel Utreexo nodes, we have forked a Golang implementation of Bitcoin called btcd (github.com/btcsuite/btcd) and have based the parallel Utreexo nodes off of that code. We have realized that the parallel node that we're building only needs a few parts of the btcd code, and is very much a departure from how the btcd nodes are constructed. Therefore, we have made the decision to refactor the code and internalize all the btcd code that we need to our current Utreexo repository (github.com/mit-dci/utreexo). This will help with readability and maintainability of the code, as we can get rid of duplicate code that was needed to keep both of the nodes working. After the refactoring has concluded, I'll once again pursue the parallel Utreexo node implementation. This work is planned to conclude before the Capstone project is to end.

Moving forward, we're looking to do more benchmarks and testing with the parallel Utreexo nodes and do a full release of our findings. I personally will be posting the results to the Bitmex Research, while Tadge Dryja will do the same for MIT Media Lab.