We've recently released a write-up of how we used TLA+ to model, find and fix problems in a state channel protocol.

Tl;dr:

- we want to show that, assuming chain liveliness and lack of censorship, a state channel participant can withdraw their funds in a finite amount of time
- we reduced the problem to showing a "channel progression claim": that if the channel is at state n

, a participant can either force it to close, or force it to progress to state n+1

- we create a TLA+ model that pitches an honest participant Alice, against an adversary Eve, who controls all the other participants and can front-run Alice's transactions
- we realise that there's an attack against the initial protocol, fix it, find another problem, fix that, and end up with a protocol that has the required properties, and also seems simpler and more aesthetically pleasing

You can find the code here, which includes instructions on how to get started and run the model yourself.

We found the experience of using TLA+ pretty interesting. There was a bit of a learning curve at the beginning, but it was really exciting where it started finding succinct examples of attacks that we were only partially aware of.

Wanted to share with ETHReseach to hear people's thoughts, and in case it's useful in anyone else's projects.