4 Proofs of Construction

- approach to proving the correctness of our state channel network protocol

- the nature of state channels tends to make the logic complex - value is moved between participants by exchanging statements about the distribution of assets held on-chain - inevitably you end up reasoning about the statements you hold and their interpretation by the chain, and the possible actions of the other participants in the channel

- inherent danger, funds are locked on-chain - participants act maliciously and stop cooperating at any point in the protocol - safe at all times

4.1 Channel Funding and Value

We will start by considering the interpretation of the outcome of a state channel. Suppose A is a participant in a state channel, L, that reaches an (allocation) outcome, ω, that allocates x coins to A. What does that mean for A? In particular, how much more can A withdraw from the system due to that outcome? Understanding this is key to analysing state channel networks.

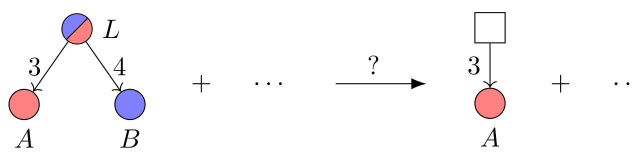


Figure 1

There is one case where the answer to these questions is very straightforward: where the channel L itself has enough coins in the adjudicator to cover the entire allocation. In this case, we say the channel is **directly funded**. If this happens, A will receive all x coins allocated to them in ω.

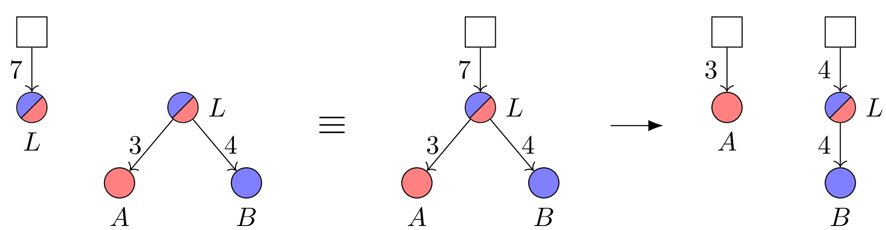


Figure 2

This is a good start, but the whole point of state channel *networks* is to move beyond the case where every channel needs to be directly funded. Suppose instead that L is not directly funded but there is another channel, L', that is. Further suppose that L' has reached an outcome where all its coins are allocated to L. Using this outcome, we know we can redistribute the coins in the adjudicator to L, recreating the situation above, where L was directly funded. In this situation we can also say that A will receive the x coins from the outcome of L. Note that we did not actually need to perform the redistribution on-chain to reach this conclusion - we just needed to be able to reason that the outcome enabled us to.

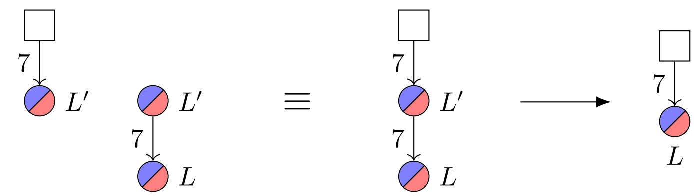


Figure 3

- so far we've been looking at outcomes - but in a state channel we don't have outcomes - we can see the state of the adjudicator: direct fundings, finalized outcomes - we have some private information: private keys, secrets for the channel algorithm (e.g. our ship positions in a game of battleships) - we have the statements we've received and the statements we've sent

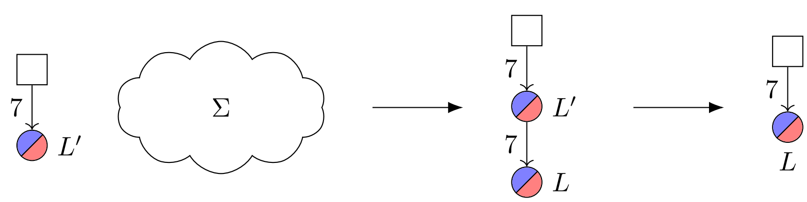


Figure 4

- definition of funding - definition of value

4.2 Constructing

- 1. show that a given network funds a channel - 2. show you can build it step-by-step

second point is important - single channel updates - keep that channels are independent - which allows us to reason about finalizability on a per channel basis

simple rule that you can transition between them if you have equal values

4.3 Unbeatable Strategies

- unbeatable strategy for obtaining a balance on-chain - two parts: finalization and redistribution

- finalization is per channel - channels independent - outcome isn't always determined

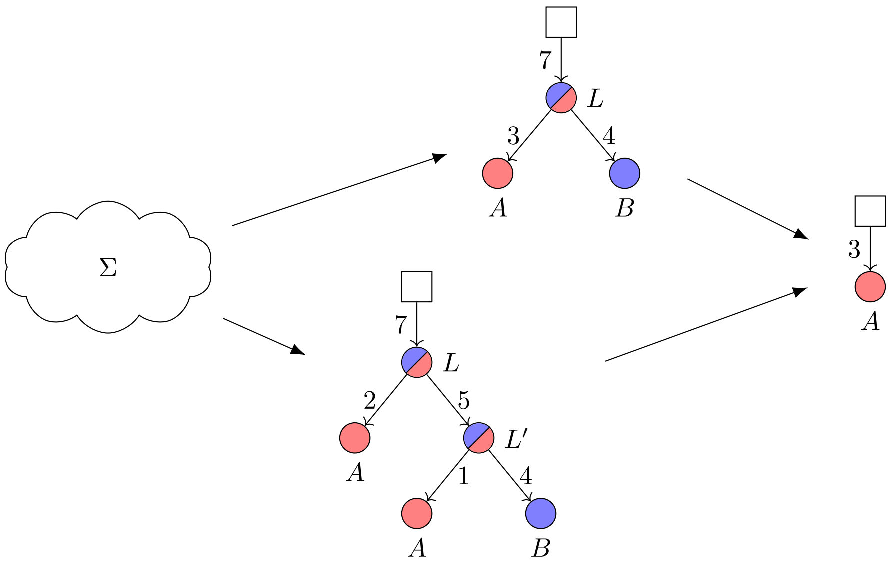


Figure 5: *Cool, huh?*

- more in finalizable outcomes

- redistribution is hard - need to consider all different possibilities - develop tools for the protocol to argue about this - will be done in the turbo / nitro section

- rules

4.4 Finalizable Outcomes

- definition in terms of unbeatable strategy

- example: next mover

- different possibilities - finalized

- universal finalizability - two examples - diagram: FM states

- enabled outcomes

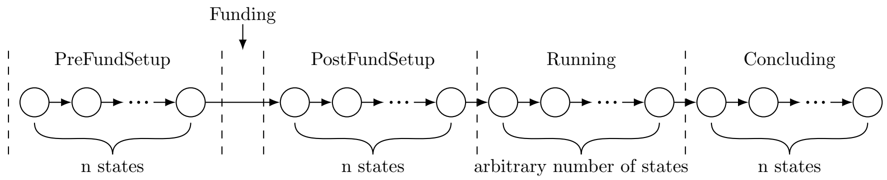


Figure 6: *Cool, huh?*

4.5 Consensus Game

- consensus game if FM application - deals in outcome. accepted outcome, propose a new one - has the property that the only two possible outcomes are A and B

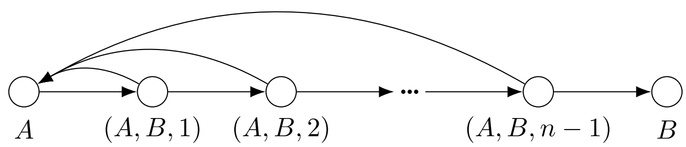


Figure 7: *Cool, huh?*

4.6 Outcomes First

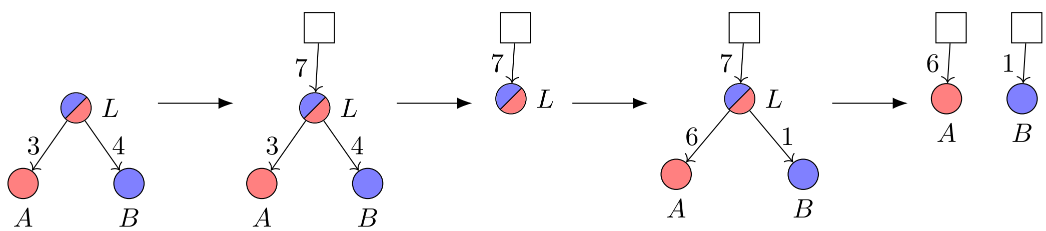
- in practice it is hard to write states - instead we will write outcomes and reason about when you can transition between them

- use special type of channel - consensus game channel

- if I have two network outcomes that differ in the outcome of a single CG channel - then I can find a sequence of single-update network states that interpolate between them

- write down a sequence of outcomes - update one channel at a time - and have the same value to all participants

- the start and conclude states are also finalizable



(a) (b) (c) (d) (e)

Figure 8: *Cool, huh?*