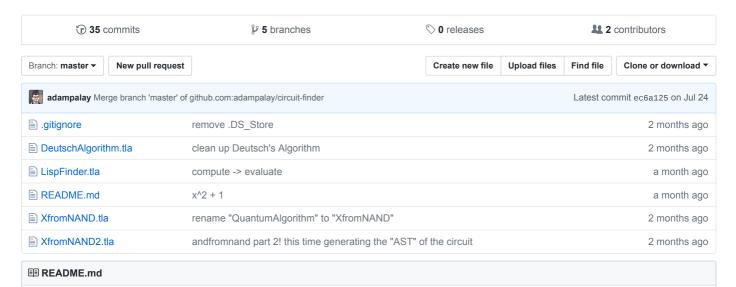
adampalay / circuit-finder

use tla+ to find programs, from circuits, quantum, to Lisp



circuit-finder

use tla+ to synthesize lisp expressions and circuits (classical or quantum)

TLA+ helps you check for bugs in the designs of your systems. The idea is you use TLA+ to specify your program as a state machine, then check to verify that certain undesireable states are unreachable. For example, you might want to check that your algorithm for facilitating intra-bank transfers doesn't change the total amount of money the bank holds.

Circuit-finder is based on an idea that you can think of activity of programming itself as a kind of state machine. As you construct a program, line by line, you alter the "state" of the program you're writing. If we work with a restricted programming language and have clear specifications, then presumably we can use TLA+ to construct programs from their specifications.

This repo contains some examples of these generated programs.

XfromNAND simulates classical chip construction. Starting with two input wires and only using NAND gates, XfromNAND can then be induced to construct any number of other logic gates. In this case, it implements an adder with a carry. But you can imagine changing Goal to the truth table of an AND gate, a NOT gate, or any other number of logical gates.

DeutschAlgorithm "discovers" a simple quantum computing algorithm, Deutsch's Algorithm, through specifying available quantum gates and the condition we want the algorithm to discover.

This approach to program generation generally isn't scalable, since TLA+ exhaustively searches the space of possible programs. But it works very well for these three examples!