We recommend following the guidelines below to ensure the tests achieve the highest degree of coverage and efficacy.

Manticore recommendations

❑ **Always use an UNSAT check to verify properties, not SAT.** Testing that the constraints of a path are satisfiable (SAT) does not mean that the constraints are always true, but only that these constraints are satisfiable for at least one valuation of the inputs. A property should be tested by checking the unsatisfiability (UNSAT) of the negation of the property.

❑ **Use `m.all_states` or `m.running_states` to check for properties on running states.** Some tests use `m.terminated_states`, which will not iterate over the running states. As a result, the tests will not cover all the paths.

❑ **Verify the existence of running states before checking any of their properties.** Some tests incorrectly assume that there are running states available and *pass* if they find none.

❑ **Use FilterFunction to restrict the functions to explore during the symbolic execution of transactions.** `FilterFunction` restricts exploration to a set of functions and provides.

❑ **Keep track of solver timeouts, since they could indicate missing paths to be explored.** If the solver times out during a property check, it is not possible to verify that the property is never violated.

❑ **Verify properties using different callers in the required transactions.** In Manticore, the caller of an ethereum transaction is always concrete, so it is essential to verify that a properties hold regardless of the used caller.

❑ **Check the length of `state.input_symbols.`** Checking `state.input_symbols` length would help to catch incorrect assumptions on its elements.