TWEAG

Dependency Update of MinSwap Test Suite

Smart Contract Verification Team

sha256sum	File Name
00cd88...d34361	audit/Audit.hs
1033e6...3728b1	audit/Audit/Attacks/DatumHijacking.hs
51f24c...a4b56c	audit/Audit/Attacks.hs
03995e...fa930e	audit/Audit/ConstantProductPool/Utils.hs
e3aec0...3be2f5	audit/Audit/ConstantProductPool/BasicOrderProcessing.hs
85d331...7ad55f	audit/Audit/ConstantProductPool/ProfitSharing.hs
3eeffa...30825a	audit/Audit/ConstantProductPool/WithdrawLiquidity.hs
152c91...846981	audit/Audit/ConstantProductPool/Batching.hs
ac31e0...47498e	audit/Audit/OffChain.hs
df3cf7...545de0	audit/Audit/MinSwapScripts.hs
e515fc...96d24c	audit/CookedAdditions.hs

Table 1: *List of files that were modified by Tweag.*

Purpose and Scope

The purpose of this assignment was to update the test suite that Tweag delivered to MinSwap with their audit, making it compile and run with the latest version of both `cooked-validators` and MinSwap's code. This does not constitute a subsequent audit, nor does it assess whether or not MinSwap's modifications to their product addresses the concerns listed on our audit report[1].

The scope of this assignment was twofold: (A) modifying the files under the `audit/` and (B) deleting the files under the `cooked-validators/` of MinSwap's repository. Table 1 lists the files that were modified by Tweag under (A), all the files under (B) were removed. The files under `src/` were *not* modified.

During our work we have not added nor removed any test. Figure 1 displays the results of running the test suite after this assignment. The only failing test is refered to by concern 2.2.3.1 on our original audit report. Finally, some modifications to MinSwap's protocol rendered three tests irrelevant. As a result, we have opted to marking them as *ignored* instead of removing them.

Deliverables

Tweag worked started from commit `95fc7c437` and finished at commit `d3e5f098e`. The differences between those two commits yields a patch file containing all the changes to MinSwap's repository which has a `sha256sum` of:

`5bb9c4bfc784fe6afb96bd923e9392017a623790ae1725cf07eecd46e86ab8ee`

This assignment was delivered through GitHub, in the form of a pull-request into MinSwap's repository `minswap/minswap-dex-tweag`, together with a copy of this report and the patch file with the above hash by e-mail.

[1] Delivered to MinSwap on the first of February of 2022, with a `sha256sum` matching `16e37fbc22...67c92d5e8`

```
Tweag Audit
  Apply Basic Orders
    Unit tests: one order at a time
      deposit order:                                            OK
      withdraw order:                                           OK
      one-side deposit order:                                   OK
      swap-exact-out order:                                     OK
      swap-exact-in order:                                      OK
    Unit test: price increases on successive orders:            OK
    Unit test: dishonest batcher can use custom pool:           FAIL (expected)
    Property tests
      generated order succeeds:                                 OK
      Price increases:                                          OK
    Multiple Pools
      batcher can batch on the pool chosen by the order author: OK
      batcher can't batch on a different pool (!):              FAIL
  Batching
    Unit tests
      two deposits:                                             OK
    Property tests
      can batch up to 24 orders at once:                        OK
      can batch up to 24 in different associations:             OK
    Obsolete Tests
      is associative modulo fees:                               IGNORED
      batchers cannot permute order lists (!):                  IGNORED
      is associative modulo fees:                               IGNORED
  Profit Sharing
    Unit tests
      can turn it on:                                           OK
    Property tests
      orders can be processed with profit-sharing on:           OK
  Withdraw Liquidity
    Unit tests
      can be withdrawn:                                         OK
  Attacks
    datum-hijacking on WithdrawLiquidity (!):                   OK
    order-stealing on WithdrawLiquidity (!):                    OK
    mint LP tokens on WithdrawLiquidity (!):                    OK
    mint LP tokens on ApplyOrder or on orders:                  OK
    mint pool tokens on ApplyOrder or on orders:                OK
```

FIGURE 1: *Tests results as of commit* `d3e5f098e`.