Código 1:

```python
solver.get_transaction_sequence(state, constraints +
state.world_state.constraints)
```

Explicação Detalhada: Este código é responsável por gerar uma sequência concreta de transações baseada no estado global e nas constraints fornecidas. Ele combina as constraints passadas como parâmetro com as constraints do `world_state` atual. O objetivo é criar uma sequência válida de transações que satisfaça todas as condições.

Código 2:

```python
def get_transaction_sequence(
    global_state: GlobalState, constraints: Constraints
) -> Dict[str, Any]:
    """Generate concrete transaction sequence.
    Note: This function only considers the constraints in
constraint argument,
    which in some cases is expected to differ from
global_state's constraints


    :param global_state: GlobalState to generate transaction
sequence for
    :param constraints: list of constraints used to generate
transaction sequence
    """
    transaction_sequence =
global_state.world_state.transaction_sequence
    concrete_transactions = []
    tx_constraints, minimize = _set_minimisation_constraints(
        transaction_sequence, constraints.copy(), [], 5000,
global_state.world_state
    )
```

```
    try:
        model = get_model(tx_constraints, minimize=minimize)
    except UnsatError:
        raise UnsatError


    if isinstance(transaction_sequence[0],
ContractCreationTransaction):
        initial_world_state =
transaction_sequence[0].prev_world_state
    else:
        initial_world_state =
transaction_sequence[0].world_state


    initial_accounts = initial_world_state.accounts


    for transaction in transaction_sequence:
        concrete_transaction =
_get_concrete_transaction(model, transaction)
        concrete_transactions.append(concrete_transaction)


    min_price_dict: Dict[str, int] = {}
    for address in initial_accounts.keys():
        min_price_dict[address] = model.eval(
            initial_world_state.starting_balances[
                symbol_factory.BitVecVal(address, 256)
            ].raw,
            model_completion=True,
        ).as_long()


    concrete_initial_state =
_get_concrete_state(initial_accounts, min_price_dict)
    if isinstance(transaction_sequence[0],
ContractCreationTransaction):
```

```
        code = transaction_sequence[0].code
        _replace_with_actual_sha(concrete_transactions, model,
code)
    else:
        _replace_with_actual_sha(concrete_transactions, model)
    _add_calldata_placeholder(concrete_transactions,
transaction_sequence)
    steps = {"initialState": concrete_initial_state, "steps":
concrete_transactions}


    return steps
```

Dados de debug:

Python

```
constraints  = [UGT(2_gas, 2300),
1271270613000041655817448348132275889066893754095 ==
Concat(0,
       If(1_calldatasize <= 12, 0, 1_calldata[12]),
       If(1_calldatasize <= 13, 0, 1_calldata[13]),
       If(1_calldatasize <= 14, 0, 1_calldata[14]),
       If(1_calldatasize <= 15, 0, 1_calldata[15]),
       If(1_calldatasize <= 16, 0, 1_calldata[16]),
       If(1_calldatasize <= 17, 0, 1_calldata[17]),
       If(1_calldatasize <= 18, 0, 1_calldata[18]),
       If(1_calldatasize <= 19, 0, 1_calldata[19]),
       If(1_calldatasize <= 20, 0, 1_calldata[20]),
       If(1_calldatasize <= 21, 0, 1_calldata[21]),
       If(1_calldatasize <= 22, 0, 1_calldata[22]),
       If(1_calldatasize <= 23, 0, 1_calldata[23]),
       If(1_calldatasize <= 24, 0, 1_calldata[24]),
       If(1_calldatasize <= 25, 0, 1_calldata[25]),
```

```
       If(1_calldatasize <= 26, 0, 1_calldata[26]),
       If(1_calldatasize <= 27, 0, 1_calldata[27]),
       If(1_calldatasize <= 28, 0, 1_calldata[28]),
       If(1_calldatasize <= 29, 0, 1_calldata[29]),
       If(1_calldatasize <= 30, 0, 1_calldata[30]),
       If(1_calldatasize <= 31, 0, 1_calldata[31])),
Or(Not(ULE(balance[100475310549029526324481294656594819817774295
8590],
           call_value1)),
   balance[100475310549029526324481294656594819817774295858
0] ==
   call_value1), True, call_value1 == 0, 1_calldatasize ==
4562, True, And(Or(1_calldatasize <= 11, 1_calldata[11] == 0),
    Or(1_calldatasize <= 10, 1_calldata[10] == 0),
    Or(1_calldatasize <= 9, 1_calldata[9] == 0),
    Or(1_calldatasize <= 8, 1_calldata[8] == 0),
    Or(1_calldatasize <= 7, 1_calldata[7] == 0),
    Or(1_calldatasize <= 6, 1_calldata[6] == 0),
    Or(1_calldatasize <= 5, 1_calldata[5] == 0),
    Or(1_calldatasize <= 4, 1_calldata[4] == 0),
    Or(1_calldatasize <= 3, 1_calldata[3] == 0),
    Or(1_calldatasize <= 2, 1_calldata[2] == 0),
    Or(1_calldatasize <= 1, 1_calldata[1] == 0),
    Or(1_calldatasize <= 0, 1_calldata[0] == 0)), Power(256,
0) == 1, Power(256, 0) == 1, Or(Not(ULE(Store(Store(balance,

5142144005605572834601741900166540107421644931 1,

balance[5142144005605572834601741900166540107421644931 1] +
                     call_value1),

100475310549029526324481294656594819817774295858 0,

balance[100475310549029526324481294656594819817774295858 0] +

1157920892373161954235709850086879078532699846656405640394575840079131296399 35*
                call_value1)[sender_2],
           call_value2)),
```

```
   Store(Store(balance,

5142144005605572834601741900166540107421644931 1,

balance[5142144005605572834601741900166540107421644931 1] +
               call_value1),
         1004753105490295263244812946565948198177742958590,

balance[1004753105490295263244812946565948198177742958590] +

1157920892373161954235709850086879078532699846656405640394575840079131296399 35*
         call_value1)[sender_2] ==
   call_value2), Or(sender_2 ==
   1004753105490295263244812946565948198177742958590,
   sender_2 ==
   1271270613000041655817448348132275889066893754095,
   sender_2 ==
   9743344248872686121357898884775220131039550286 50), ULE(4,
2_calldatasize), Not(ULE(2952712416,
        Concat(If(2_calldatasize <= 0, 0, 2_calldata[0]),
               If(2_calldatasize <= 1, 0, 2_calldata[1]),
               If(2_calldatasize <= 2, 0, 2_calldata[2]),
               If(2_calldatasize <= 3, 0, 2_calldata[3])))),
And(2_calldata[3] == 61,
    Not(2_calldatasize <= 3),
    2_calldata[2] == 181,
    Not(2_calldatasize <= 2),
    2_calldata[1] == 230,
    Not(2_calldatasize <= 1),
    2_calldata[0] == 33,
    Not(2_calldatasize <= 0)), call_value2 == 0, 32 <=
1157920892373161954235709850086879078532699846656405640394575840079131296399 32 +
2_calldatasize, And(Or(2_calldatasize <= 15, 2_calldata[15] ==
0),
    Or(2_calldatasize <= 14, 2_calldata[14] == 0),
    Or(2_calldatasize <= 13, 2_calldata[13] == 0),
    Or(2_calldatasize <= 12, 2_calldata[12] == 0),
```

```
    Or(2_calldatasize <= 11, 2_calldata[11] == 0),
    Or(2_calldatasize <= 10, 2_calldata[10] == 0),
    Or(2_calldatasize <= 9, 2_calldata[9] == 0),
    Or(2_calldatasize <= 8, 2_calldata[8] == 0),
    Or(2_calldatasize <= 7, 2_calldata[7] == 0),
    Or(2_calldatasize <= 6, 2_calldata[6] == 0),
    Or(2_calldatasize <= 5, 2_calldata[5] == 0),
    Or(2_calldatasize <= 4, 2_calldata[4] == 0)),
Not(And(Or(2_calldatasize <= 35, 2_calldata[35] == 0),
        Or(2_calldatasize <= 34, 2_calldata[34] == 0),
        Or(2_calldatasize <= 33, 2_calldata[33] == 0),
        Or(2_calldatasize <= 32, 2_calldata[32] == 0),
        Or(2_calldatasize <= 31, 2_calldata[31] == 0),
        Or(2_calldatasize <= 30, 2_calldata[30] == 0),
        Or(2_calldatasize <= 29, 2_calldata[29] == 0),
        Or(2_calldatasize <= 28, 2_calldata[28] == 0),
        Or(2_calldatasize <= 27, 2_calldata[27] == 0),
        Or(2_calldatasize <= 26, 2_calldata[26] == 0),
        Or(2_calldatasize <= 25, 2_calldata[25] == 0),
        Or(2_calldatasize <= 24, 2_calldata[24] == 0),
        Or(2_calldatasize <= 23, 2_calldata[23] == 0),
        Or(2_calldatasize <= 22, 2_calldata[22] == 0),
        Or(2_calldatasize <= 21, 2_calldata[21] == 0),
        Or(2_calldatasize <= 20, 2_calldata[20] == 0),
        Or(2_calldatasize <= 19, 2_calldata[19] == 0),
        Or(2_calldatasize <= 18, 2_calldata[18] == 0),
        Or(2_calldatasize <= 17, 2_calldata[17] == 0),
        Or(2_calldatasize <= 16, 2_calldata[16] == 0))),
Power(256, 0) == 1, Extract(159, 0, sender_2) ==
1004753105490295263244812946565948198177742958590, Power(256,
0) == 1, Not(2_extcodesize_Concat(0,
       If(1_calldatasize <= 12, 0, 1_calldata[12]),
       If(1_calldatasize <= 13, 0, 1_calldata[13]),
       If(1_calldatasize <= 14, 0, 1_calldata[14]),
       If(1_calldatasize <= 15, 0, 1_calldata[15]),
       If(1_calldatasize <= 16, 0, 1_calldata[16]),
       If(1_calldatasize <= 17, 0, 1_calldata[17]),
       If(1_calldatasize <= 18, 0, 1_calldata[18]),
       If(1_calldatasize <= 19, 0, 1_calldata[19]),
```

```
       If(1_calldatasize <= 20, 0, 1_calldata[20]),
       If(1_calldatasize <= 21, 0, 1_calldata[21]),
       If(1_calldatasize <= 22, 0, 1_calldata[22]),
       If(1_calldatasize <= 23, 0, 1_calldata[23]),
       If(1_calldatasize <= 24, 0, 1_calldata[24]),
       If(1_calldatasize <= 25, 0, 1_calldata[25]),
       If(1_calldatasize <= 26, 0, 1_calldata[26]),
       If(1_calldatasize <= 27, 0, 1_calldata[27]),
       If(1_calldatasize <= 28, 0, 1_calldata[28]),
       If(1_calldatasize <= 29, 0, 1_calldata[29]),
       If(1_calldatasize <= 30, 0, 1_calldata[30]),
       If(1_calldatasize <= 31, 0, 1_calldata[31])) ==
    0)]
```

outras informações de debug que foram sendo geradas ao executar o codigo 2:

```
transaction_sequence =
[<mythril.laser.ethereum.transaction.transaction_models.Contra
ctCreationTransaction object at 0x7f758b6b2270>,
<mythril.laser.ethereum.transaction.transaction_models.Message
CallTransaction object at 0x7f75889b25d0>]

tx_constraints = [UGT(2_gas, 2300),
127127061300004165581744834813227588906689375409 5 ==
Concat(0,
       If(1_calldatasize <= 12, 0, 1_calldata[12]),
       If(1_calldatasize <= 13, 0, 1_calldata[13]),
       If(1_calldatasize <= 14, 0, 1_calldata[14]),
       If(1_calldatasize <= 15, 0, 1_calldata[15]),
       If(1_calldatasize <= 16, 0, 1_calldata[16]),
       If(1_calldatasize <= 17, 0, 1_calldata[17]),
       If(1_calldatasize <= 18, 0, 1_calldata[18]),
       If(1_calldatasize <= 19, 0, 1_calldata[19]),
       If(1_calldatasize <= 20, 0, 1_calldata[20]),
       If(1_calldatasize <= 21, 0, 1_calldata[21]),
       If(1_calldatasize <= 22, 0, 1_calldata[22]),
       If(1_calldatasize <= 23, 0, 1_calldata[23]),
       If(1_calldatasize <= 24, 0, 1_calldata[24]),
       If(1_calldatasize <= 25, 0, 1_calldata[25]),
```

```
       If(1_calldatasize <= 26, 0, 1_calldata[26]),
       If(1_calldatasize <= 27, 0, 1_calldata[27]),
       If(1_calldatasize <= 28, 0, 1_calldata[28]),
       If(1_calldatasize <= 29, 0, 1_calldata[29]),
       If(1_calldatasize <= 30, 0, 1_calldata[30]),
       If(1_calldatasize <= 31, 0, 1_calldata[31])),
Or(Not(ULE(balance[100475310549029526324481294656594819817774295
8590],
           call_value1)),
   balance[100475310549029526324481294656594819817774295858
0] ==
   call_value1), True, call_value1 == 0, 1_calldatasize ==
4562, True, And(Or(1_calldatasize <= 11, 1_calldata[11] == 0),
    Or(1_calldatasize <= 10, 1_calldata[10] == 0),
    Or(1_calldatasize <= 9, 1_calldata[9] == 0),
    Or(1_calldatasize <= 8, 1_calldata[8] == 0),
    Or(1_calldatasize <= 7, 1_calldata[7] == 0),
    Or(1_calldatasize <= 6, 1_calldata[6] == 0),
    Or(1_calldatasize <= 5, 1_calldata[5] == 0),
    Or(1_calldatasize <= 4, 1_calldata[4] == 0),
    Or(1_calldatasize <= 3, 1_calldata[3] == 0),
    Or(1_calldatasize <= 2, 1_calldata[2] == 0),
    Or(1_calldatasize <= 1, 1_calldata[1] == 0),
    Or(1_calldatasize <= 0, 1_calldata[0] == 0)), Power(256,
0) == 1, Power(256, 0) == 1, Or(Not(ULE(Store(Store(balance,

5142144005605572834601741900166540107421644931 1,

balance[5142144005605572834601741900166540107421644931 1] +
                     call_value1),

100475310549029526324481294656594819817774295858 0,

balance[100475310549029526324481294656594819817774295858 0] +

1157920892373161954235709850086879078532699846656405640394575840079131296399 35*
                call_value1)[sender_2],
           call_value2)),
```

```
   Store(Store(balance,

5142144005605572834601741900166540107421644931 1,

balance[5142144005605572834601741900166540107421644931 1] +
               call_value1),
         1004753105490295263244812946565948198177742958590,

balance[1004753105490295263244812946565948198177742958590] +

1157920892373161954235709850086879078532699846656405640394575840079131296399 35*
         call_value1)[sender_2] ==
   call_value2), Or(sender_2 ==
   1004753105490295263244812946565948198177742958590,
   sender_2 ==
   1271270613000041655817448348132275889066893754095,
   sender_2 ==
   9743344248872686121357898884775220131039550286 50), ULE(4,
2_calldatasize), Not(ULE(2952712416,
        Concat(If(2_calldatasize <= 0, 0, 2_calldata[0]),
               If(2_calldatasize <= 1, 0, 2_calldata[1]),
               If(2_calldatasize <= 2, 0, 2_calldata[2]),
               If(2_calldatasize <= 3, 0, 2_calldata[3])))),
And(2_calldata[3] == 61,
    Not(2_calldatasize <= 3),
    2_calldata[2] == 181,
    Not(2_calldatasize <= 2),
    2_calldata[1] == 230,
    Not(2_calldatasize <= 1),
    2_calldata[0] == 33,
    Not(2_calldatasize <= 0)), call_value2 == 0, 32 <=
1157920892373161954235709850086879078532699846656405640394575840079131296399 32 +
2_calldatasize, And(Or(2_calldatasize <= 15, 2_calldata[15] ==
0),
    Or(2_calldatasize <= 14, 2_calldata[14] == 0),
    Or(2_calldatasize <= 13, 2_calldata[13] == 0),
    Or(2_calldatasize <= 12, 2_calldata[12] == 0),
```

```
    Or(2_calldatasize <= 11, 2_calldata[11] == 0),
    Or(2_calldatasize <= 10, 2_calldata[10] == 0),
    Or(2_calldatasize <= 9, 2_calldata[9] == 0),
    Or(2_calldatasize <= 8, 2_calldata[8] == 0),
    Or(2_calldatasize <= 7, 2_calldata[7] == 0),
    Or(2_calldatasize <= 6, 2_calldata[6] == 0),
    Or(2_calldatasize <= 5, 2_calldata[5] == 0),
    Or(2_calldatasize <= 4, 2_calldata[4] == 0)),
Not(And(Or(2_calldatasize <= 35, 2_calldata[35] == 0),
        Or(2_calldatasize <= 34, 2_calldata[34] == 0),
        Or(2_calldatasize <= 33, 2_calldata[33] == 0),
        Or(2_calldatasize <= 32, 2_calldata[32] == 0),
        Or(2_calldatasize <= 31, 2_calldata[31] == 0),
        Or(2_calldatasize <= 30, 2_calldata[30] == 0),
        Or(2_calldatasize <= 29, 2_calldata[29] == 0),
        Or(2_calldatasize <= 28, 2_calldata[28] == 0),
        Or(2_calldatasize <= 27, 2_calldata[27] == 0),
        Or(2_calldatasize <= 26, 2_calldata[26] == 0),
        Or(2_calldatasize <= 25, 2_calldata[25] == 0),
        Or(2_calldatasize <= 24, 2_calldata[24] == 0),
        Or(2_calldatasize <= 23, 2_calldata[23] == 0),
        Or(2_calldatasize <= 22, 2_calldata[22] == 0),
        Or(2_calldatasize <= 21, 2_calldata[21] == 0),
        Or(2_calldatasize <= 20, 2_calldata[20] == 0),
        Or(2_calldatasize <= 19, 2_calldata[19] == 0),
        Or(2_calldatasize <= 18, 2_calldata[18] == 0),
        Or(2_calldatasize <= 17, 2_calldata[17] == 0),
        Or(2_calldatasize <= 16, 2_calldata[16] == 0))),
Power(256, 0) == 1, Extract(159, 0, sender_2) ==
1004753105490295263244812946565948198177742958590, Power(256,
0) == 1, Not(2_extcodesize_Concat(0,
       If(1_calldatasize <= 12, 0, 1_calldata[12]),
       If(1_calldatasize <= 13, 0, 1_calldata[13]),
       If(1_calldatasize <= 14, 0, 1_calldata[14]),
       If(1_calldatasize <= 15, 0, 1_calldata[15]),
       If(1_calldatasize <= 16, 0, 1_calldata[16]),
       If(1_calldatasize <= 17, 0, 1_calldata[17]),
       If(1_calldatasize <= 18, 0, 1_calldata[18]),
       If(1_calldatasize <= 19, 0, 1_calldata[19]),
```

```
       If(1_calldatasize <= 20, 0, 1_calldata[20]),
       If(1_calldatasize <= 21, 0, 1_calldata[21]),
       If(1_calldatasize <= 22, 0, 1_calldata[22]),
       If(1_calldatasize <= 23, 0, 1_calldata[23]),
       If(1_calldatasize <= 24, 0, 1_calldata[24]),
       If(1_calldatasize <= 25, 0, 1_calldata[25]),
       If(1_calldatasize <= 26, 0, 1_calldata[26]),
       If(1_calldatasize <= 27, 0, 1_calldata[27]),
       If(1_calldatasize <= 28, 0, 1_calldata[28]),
       If(1_calldatasize <= 29, 0, 1_calldata[29]),
       If(1_calldatasize <= 30, 0, 1_calldata[30]),
       If(1_calldatasize <= 31, 0, 1_calldata[31])) ==
    0), Or(Not(ULE(5000, 1_calldatasize)), 1_calldatasize ==
5000), Or(Not(ULE(10000000000000000000,

balance[100475310549029526324481294656594819817774295859O])),
   balance[100475310549029526324481294656594819817774295859O]
==
   10000000000000000000), Or(Not(ULE(5000, 2_calldatasize)),
2_calldatasize == 5000), Or(Not(ULE(10000000000000000000,
balance[sender_2])),
   balance[sender_2] == 10000000000000000000),
Or(Not(ULE(10000000000000000000,

balance[100475310549029526324481294656594819817774295859O])),
   balance[100475310549029526324481294656594819817774295859O]
==
   10000000000000000000), Or(Not(ULE(10000000000000000000,

balance[127127061300004165581744834813227588906689375409I])),
   balance[127127061300004165581744834813227588906689375409I]
==
   10000000000000000000), Or(Not(ULE(10000000000000000000,

balance[514214400560557283460174190016654010742164493ll])),
   balance[514214400560557283460174190016654010742164493ll] ==
   10000000000000000000)]
```

```
minimize = (1_calldatasize, call_value1, 2_calldatasize,
call_value2)

model = <mythril.laser.smt.model.Model object at
0x7f758b5011c0>

model.raw = [[balance = K(BitVec(256), 0),
 2_gas = 10494,
 2_extcodesize_Concat(0,
       If(1_calldatasize <= 12, 0, 1_calldata[12]),
       If(1_calldatasize <= 13, 0, 1_calldata[13]),
       If(1_calldatasize <= 14, 0, 1_calldata[14]),
       If(1_calldatasize <= 15, 0, 1_calldata[15]),
       If(1_calldatasize <= 16, 0, 1_calldata[16]),
       If(1_calldatasize <= 17, 0, 1_calldata[17]),
       If(1_calldatasize <= 18, 0, 1_calldata[18]),
       If(1_calldatasize <= 19, 0, 1_calldata[19]),
       If(1_calldatasize <= 20, 0, 1_calldata[20]),
       If(1_calldatasize <= 21, 0, 1_calldata[21]),
       If(1_calldatasize <= 22, 0, 1_calldata[22]),
       If(1_calldatasize <= 23, 0, 1_calldata[23]),
       If(1_calldatasize <= 24, 0, 1_calldata[24]),
       If(1_calldatasize <= 25, 0, 1_calldata[25]),
       If(1_calldatasize <= 26, 0, 1_calldata[26]),
       If(1_calldatasize <= 27, 0, 1_calldata[27]),
       If(1_calldatasize <= 28, 0, 1_calldata[28]),
       If(1_calldatasize <= 29, 0, 1_calldata[29]),
       If(1_calldatasize <= 30, 0, 1_calldata[30]),
       If(1_calldatasize <= 31, 0, 1_calldata[31])) = 1,
 2_calldata = Store(Store(Store(Store(Store(K(BitVec(256),
                                            0),
                                            2,
                                            181),
                                       3,
                                       61),
                                  0,
                                  33),
                             1,
                             230),
```

```
                    35,
                    1),
 2_calldatasize = 36,
 1_calldata =
Store(Store(Store(Store(Store(Store(Store(Store(Store(Store(St
ore(Store(Store(Store(Store(Store(Store(Store(Store(Store(K(Bi
tVec(256),
                                          0),
                                          14,
                                          190),
                                          25,
                                          173),
                                          22,
                                          190),
                                          21,
                                          173),
                                          29,
                                          173),
                                          23,
                                          239),
                                          15,
                                          239),
                                          16,
                                          222),
                                          30,
                                          190),
                                          12,
                                          222),
                                          31,
                                          239),
                                          20,
                                          222),
                                          24,
                                          222),
                                          19,
                                          239),
                                          13,
                                          173),
                                          18,
```

```
                                        190),
                                    28,
                                    222),
                                17,
                                173),
                            27,
                            239),
                        26,
                        190),
 sender_2 = 100475310549029526324481294656594819817774295859 0,
 call_value1 = 0,
 1_calldatasize = 4562,
 call_value2 = 0,
 Power = [else -> 1],
 keccak256_512 = [else ->

115792089237316195423570985008687907450124440181300284663682953187099114743296],
 keccak256_512-1 = [else ->

115792089237316195423570985008687907853269984665640564039457584007913129639940]]]

transaction_sequence[0] =
```

ContractCreationTransaction 1 from 100475310549029526324481294656594819817774295859 0 to 0x901d12ebe1b195e5aa8748e62bd7734ae19b51f

```
initial_world_state =
<mythril.laser.ethereum.state.world_state.WorldState object at 0x7f758b6b2f90>

concrete_transaction = {'input':
'0x6080604052348015610010576000080fd5b50604051610fd2380380610fd28339818101604052810190610032919061006100ce565b336000806101000a81548173ffffffffffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffffffffffffffffffffffff160217905550806001600061010000a81548173ffffffffffffffffffffffffffffffffffffffff02191690837 3ffffffffffffffffffffffffffffffffffffffff160217905550506101405
```

```
65b60008151905061 00c881610129565b92915050565b60006020828403121
56100e057600080fd5b60006100ee848285016100b9565b915050929150505
65b600061010282610109565b9050919050565b600073ffffffffffffffffff
ffffffffffffffffffffffff8216905091905056 5b610132816100f7565b811
461013d57600080fd5b50565b610e838061014f6000396000f3fe608060405
26004361061008a5760003560e01c8063affed0e011610059578063affed0e
014610150578063d493b9ac1461017b578063f2fde38b146101a4578063f85
1a440146101cd578063fc0c546a146101f857610091565b806321e6b53d146
100965780638326acce146100bf57806391dae519146100e8578063a090588
014610125576100915 65b3661009157005b600080fd5b3480156100a257600
080fd5b506100bd6004803603810190610 0b89190610a65565b610223565b0
05b3480156100cb57600080fd5b506100e6600480360381019061 00e191906
10add565b610345565b005b3480156100f457600080fd5b5061010f6004803
60381019061010a9190610b55565b610652565b60405161011c9190610cfa5
65b60405180910390f35b34801561013157600080fd5b5061013a610672565
b60405161014791906 10c63565b60405180910390f35b34801561015c57600
080fd5b506101656106985 65b60405161017291906 10d70565b60405180910
390f35b34801561018757600080fd5b506101a2600480360381019061019d9
190610a8e565b61069e565b005b3480156101b057600080fd5b506101cb600
48036038101906101c69190610a65565b610878565b005b3480156101d9576
00080fd5b506101e26109c7565b6040516101ef9190610c63565b604051809
10390f35b34801561020457600080fd5b5061020d6109eb565b60405161021
a9190610d15565b60405180910390f35b600073ffffffffffffffffffffffff
ffffffffffffffff168173ffffffffffffffffffffffffffffffffffffffff
f16141561025d57600080fd5b60008054906101000a900473ffffffffffffff
ffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffff
ffffffff163373ffffffffffffffffffffffffffffffffffffffff1614610
2b557600080fd5b60016000905490 6101000a900473ffffffffffffffffff
ffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffff
fff1663f2fde38b826040518263ffffffff1660e01b8152600401610310919
0610c63565b60006040518083038160008780 3b15801561032a57600080fd5
b505af115801561033e573d6000803e3d6000fd5b5050505050565b6000309
05060008054906101000a900473ffffffffffffffffffffffffffffffffffff
fffff1673ffffffffffffffffffffffffffffffffffffffff163373ffffffff
ffffffffffffffffffffffffffffffff16146103d8576040517f08c379a00
0000000000000000000000000000000000000000000000000008152600
4016103cf90610d30565b60405180910390fd5b600015156004600084815260
0200190815260200160002060009054906101000a900460ff1615151461043
f576040517f08c379a0000000000000000000000000000000000000000000000
```

```
0000000000000815260040161043690610d50565b60405180910390fd5b826
0016000905490610100 0a900473ffffffffffffffffffffffffffffffffffff
fffff1673fffffffffffffffffffffffffffffffffffff166370a082318
36040518263ffffffff1660e01b8152600401610 49b9190610c63565b60206
0405180830381600087803b1580156104b557600080fd5b505af1158015610
4c9573d6000803e3d6000fd5b505050506040513d601f19601f82011682018
0604052508101906104ed9190610b7e565b10156104f857600080fd5b60016
0046000848152602001908152602001600020600061010 00a81548160ff021
916908315150217905550600160009054906101000a900473ffffffffffff
ffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffff
ffffffff1663a9059cbb858560405183 63ffffffff1660e01b81526004016
10581929190610cd1565b602060405180830381600087803b15801561059b5
7600080fd5b505af11580156105af573d6000803e3d6000fd5b50505050604
0513d601f19601f820116820180604052508101906105d39190610b2c565b5
06001808111156106 0d577f4e487b7100000000000000000000000000000
0000000000000000000000000600052602160045260246000fd5b7f2775754
2a5e1b9e8cef80f584e094d4eb63b9802f355c61b3640b71b618d5c8e82868
642876040516106449594939291906 10c7e565b60405180910390a25050505
0565b600460205280600052604060002060009150549061010 00a900460ff1
681565b600360009054906101000a900473ffffffffffffffffffffffffff
ffffffffffff1681565b60025481565b600080549061010 00a900473fffff
ffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffff
ffffffffffffffff163373ffffffffffffffffffffffffffffffffffffff
f161461072c576040517f08c379a0000000000000000000000000000000
0000000000000000000815260040161072390610d30565b60405180910
390fd5b806001600090549061010 00a900473ffffffffffffffffffffffff
ffffffffffffff1673ffffffffffffffffffffffffffffffffffffff166
370a082313060405182 63ffffffff1660e01b8152600401610 7889190610c6
3565b602060405180830381600087803b1580156107a257600080fd5b505af
11580156107b6573d6000803e3d6000fd5b505050506040513d601f19601f8
2011682018060405250810190610 8729190610b2c565b50505050565b600073fffff
ffffffffffffffffffffffffffffffff168173ffffffffffffffffffff
ffffffffffffffff1614156108b257600080fd5b600080549061010 00a9
00473ffffffffffffffffffffffffffffffffffffff1673ffffffffffff
```

```
fffffffffffffffffffffffff163373ffffffffffffffffffffffffff
fffffffffff161461090a57600080fd5b8073ffffffffffffffffffffff
fffffffffffffff16600080549061010000a900473fffffffffffffffffff
fffffffffffffffffff1673ffffffffffffffffffffffffffffffffffff
f167f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6
b6457e0604051604051809103900a3806000806101000a81548173ffffffff
ffffffffffffffffffffffffffff021916908373ffffffffffffffffff
fffffffffffffffffff1602179055505056 5b6000805490610100 0a90047
3ffffffffffffffffffffffffffffffffffffff1681565b6001600090549
06101000a900473ffffffffffffffffffffffffffffffffffff1681565
b600081359050610a2081610e08565b92915050565b600081519050610a358
1610e1f565b92915050565b600081359050610a4a81610e36565b929150505
65b600081519050610a5f81610e36565b92915050565b60006020828403121
5610a7757600080fd5b6000610a8584828501610a11565b915050929150505
65b60008060006060848631215610aa357600080fd5b6000610ab18682870
1610a11565b9350506020610ac286828701610a11565b9250506040610ad38
6828701610a3b565b91505092509250925092565b60008060006060848603121 56
10af257600080fd5b6000610b0086828701610a11565b9350506020610b118
6828701610a3b565b9250506040610b2286828701610a3b565b91505092509
25092565b60006020828403121 5610b3e57600080fd5b6000610b4c8482850
1610a26565b91505092915050565b60006020828403121 5610b6757600080f
d5b6000610b7584828501610a3b565b91505092915050565b6000602082840
31215610b9057600080fd5b6000610b9e84828501610a50565b91505092915
050565b610bb081610d9c565b82525050565b610bbf81610dae565b8252505
0565b610bce81610de4565b82525050565b6000610be1600a83610d8b565b9
1507f6f6e6c792061646d696e0000000000000000000000000000000000000
0000000600083015260208201905091905056 5b6000610c21601a83610d8b5
65b91507f7472616e7366657220616c7265616479207072 6f6365737365640
0000000000060008301526020820190509190505 65b610c5d81610dda565b8
2525050565b60006020820190506 10c786000830184610ba7565b929150505
65b600060a082019050610c936000830188610ba7565b610ca060208301876
10ba7565b610cad6040830186610c54565b610cba6060830185610c54565b6
10cc76080830184610c54565b9695505050505050565b60006040820190506
10ce66000830185610ba7565b610cf36020830184610c54565b93925050505
65b60006020820190506 10d0f6000830184610bb6565b92915050565b60006
020820190506 10d2a6000830184610bc5565b92915050565b6000602082019
0508181036000830152610d4981610bd4565b9050919050565b60006020820
190508181036000830152610d6981610c14565b9050919050565b600060208
2019050610d856000830184610c54565b92915050565b60008282526020820
```

```
19050929150505 65b6000610da782610dba565b9050919050565b600081151
59050919050565b600073ffffffffffffffffffffffffffffffffffffffff8
2169050919050565b60008190509190505 65b6000610def82610df6565b905
0919050565b6000610e0182610dba565b9050919050565b610e1181610d9c5
65b8114610e1c57600080fd5b50565b610e2881610dae565b8114610e33576
00080fd5b50565b610e3f81610dda565b8114610e4a57600080fd5b5056fea
2646970667358221220 8a8a2d65bde047a11d42eeb90ebce9ea9eed3856369
7d75f3fef84f6bdfc6b4d64736f6c6343000800003300000000000000000000
00000deadbeefdeadbeefdeadbeefdeadbeefdeadbeef0000000000000000000
0000000000000000000000000000000000000000000000000000000000000…ma
is zeros…', 'value': '0x0', 'origin':
'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe', 'address': ''}


concrete_transactions = [{'input':
'0x6080604052348015610010576000 80fd5b50604051610fd2380380610fd
2833981810160405281019061003291906100ce565b336000806101000a815
48173ffffffffffffffffffffffffffffffffffffffff021916908373ffffff
ffffffffffffffffffffffffffffffffff160217905550806001600061010
00a81548173ffffffffffffffffffffffffffffffffffffffff02191690837
3ffffffffffffffffffffffffffffffffffffffff160217905550506101405
65b600081519050610 0c881610129565b92915050565b60006020828403121
56100e057600080fd5b60006100ee848285016100b9565b915050929150505
65b60006101028261 0109565b9050919050565b600073ffffffffffffffff
ffffffffffffffffffffffff8216905091905056 5b6101328161 00f7565b811
461013d57600080fd5b50565b610e838061014f6000396000f3fe608060405
26004361061008a5760003560e01c8063affed0e011610059578063affed0e
014610150578063d493b9ac1461017b578063f2fde38b146101a4578063f85
1a440146101cd578063fc0c546a146101f857610091565b806321e6b53d146
100965780638326acce146100bf57806391dae519146100e8578063a090588
014610125576100915 65b3661009157005b600080fd5b3480156100a257600
080fd5b506100bd6004803603810190 6100b89190610a65565b610223565b0
05b3480156100cb57600080fd5b506100e6600480360381019061 00e191906
10add565b610345565b005b3480156100f457600080fd5b5061010f6004803
6038101906101 0a9190610b55565b610652565b60405161011c9190610cfa5
65b60405180910390f35b34801561013157600080fd5b5061013a610672565
b60405161014791 90610c63565b60405180910390f35b34801561015c57600
080fd5b5061016561069856 5b60405161017291906 10d70565b60405180910
390f35b34801561018757600080fd5b506101a260048036038101906101 9d9
```

190610a8e565b61069e565b005b3480156101b057600080fd5b506101cb600
48036038101906101c69190610a65565b610878565b005b3480156101d9576
00080fd5b506101e26109c7565b6040516101ef9190610c63565b604051809
10390f35b34801561020457600080fd5b5061020d6109eb565b60405161021
a9190610d15565b60405180910390f35b600073fffffffffffffffffffffff
fffffffffffffffff168173fffffffffffffffffffffffffffffffffffffff
f16141561025d57600080fd5b6000805490610100 0a900473fffffffffffff
fffffffffffffffffffffffffff1673fffffffffffffffffffffffffffffff
fffffffff163373ffffffffffffffffffffffffffffffffffffffff1614610
2b557600080fd5b6001600090549061010 00a900473fffffffffffffffffff
fffffffffffffffffffff1673fffffffffffffffffffffffffffffffffffff
fff1663f2fde38b826040518263ffffffff1660e01b8152600401610310919
0610c63565b600060405180830381600087803b15801561032a57600080fd5
b505af115801561033e573d6000803e3d6000fd5b5050505050565b6000309
05060008054906101000a900473fffffffffffffffffffffffffffffffffff
fffff1673ffffffffffffffffffffffffffffffffffffffff163373fffffff
fffffffffffffffffffffffffffffffff16146103d8576040517f08c379a00
00000000000000000000000000000000000000000000000000000008152600
4016103cf90610d30565b60405180910390fd5b600015156004600084815260
200190815260200160002060009054906101000a900460ff1615151461043
f576040517f08c379a00000000000000000000000000000000000000000000
00000000000008152600401610436 90610d50565b60405180910390fd5b826
0016000905490610100 0a900473fffffffffffffffffffffffffffffffffff
fffff1673ffffffffffffffffffffffffffffffffffffffff166370a082318
36040518263ffffffff1660e01b815260040161049b9190610c63565b60206
040518083038160008780 3b158015610 4b557600080fd5b505af1158015610
4c9573d6000803e3d6000fd5b5050505060405 13d601f19601f82011682018
0604052508101906104ed9190610b7e565b10156104f857600080fd5b60016
00460008481526020019081526020016000206000610100 0a81548160ff021
9169083151502179055506001600090549061010 00a900473fffffffffffff
fffffffffffffffffffffffffff1673fffffffffffffffffffffffffffffff
fffffffff1663a9059cbb858560405183 63ffffffff1660e01b81526004016
10581929190610cd1565b602060405180830381600087803b15801561059b5
7600080fd5b505af11580156105af573d6000803e3d6000fd5b50505050604
0513d601f19601f820116820180604052508101906105d39190610b2c565b5
0600180811115610 60d577f4e487b710000000000000000000000000000000
0000000000000000000000000600052602160045260246000fd5b7f2775754
2a5e1b9e8cef80f584e094d4eb63b9802f355c61b3640b71b618d5c8e82868
64287604051610644959493929190610c7e565b60405180910390a25050505

0565b6004602052806000526040600020600091505490610100 0a900460ff1
681565b60036000905490610100 0a900473fffffffffffffffffffffffff
fffffffffff1681565b60025481565b600080549061010 0a900473fffff
ffffffffffffffffffffffffffffffff1673ffffffffffffffffffffff
ffffffffffffffff163373ffffffffffffffffffffffffffffffffffff
f161461072c576040517f08c379a0000000000000000000000000000000
0000000000000000000081526004016107239061 0d30565b60405180910
390fd5b80600160009054906101000a900473ffffffffffffffffffffff
ffffffffffffff1673ffffffffffffffffffffffffffffffffffff166
370a08231306040518263ffffffff1660e01b81526004016107889190610c6
3565b602060405180830381600087803b1580156107a257600080fd5b505af
11580156107b6573d6000803e3d6000fd5b505050506040513d601f19601f8
201168201806040525081019061 07da9190610b7e565b10156107e55760008
0fd5b8273ffffffffffffffffffffffffffffffffffffffff1663a9059cbb8
3836040518363ffffffff1660e01b8152600401610820929190610cd1565b6
020604051808303816000878 03b15801561083a57600080fd5b505af115801
561084e573d6000803e3d6000fd5b505050506040513d601f19601f8201168
201806040525081019061 08729190610b2c565b50505050565b600073fffff
ffffffffffffffffffffffffffffffff168173ffffffffffffffffffff
ffffffffffffffff1614156108b257600080fd5b60008054906101000a9
00473ffffffffffffffffffffffffffffffffffffffff1673ffffffffffff
ffffffffffffffffffff163373ffffffffffffffffffffffffffffffff
ffffffffff161461090a57600080fd5b8073ffffffffffffffffffffffff
ffffffffffffff166000805490610100 0a900473ffffffffffffffffffff
ffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff
f167f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6
b6457e0604051604051809103 90a3806000806101000a81548173ffffffff
ffffffffffffffffffffffffffff02191690837 3ffffffffffffffffff
ffffffffffffffffff1602179055505056 5b60008054906101000a90047
3ffffffffffffffffffffffffffffffffffffffff1681565b6001600090549
06101000a900473ffffffffffffffffffffffffffffffffffffffff1681565
b600081359050610a2081610e08565b92915050565b600081519050610a358
1610e1f565b92915050565b600081359050610a4a81610e36565b929150505
65b600081519050610a5f81610e36565b92915050565b600060208284031215
5610a7757600080fd5b6000610a8584828501610a11565b915050929150505
65b60008060006060848631215610aa357600080fd5b6000610ab18682870
1610a11565b935050602061 0ac286828701610a11565b9250506040610ad38
6828701610a3b565b91505092509250925 92565b60008060006060848603121 56
10af257600080fd5b6000610b0086828701610a11565b935050602061 0b118

6828701610a3b565b9250506040610b2286828701610a3b565b91505092509
25092565b600060208284031215610b3e57600080fd5b6000610b4c8482850
1610a26565b91505092915050565b600060208284031215610b6757600080f
d5b6000610b7584828501610a3b565b91505092915050565b6000602082840
31215610b9057600080fd5b6000610b9e84828501610a50565b91505092915
050565b610bb081610d9c565b82525050565b610bbf81610dae565b8252505
0565b610bce81610de4565b82525050565b6000610be1600a83610d8b565b9
1507f6f6e6c792061646d696e0000000000000000000000000000000000000
0000000600083015260208201905091905056 5b6000610c21601a83610d8b5
65b91507f7472616e7366657220616c72656164792070726f6365737365640
0000000000060008301526020820190509190505 65b610c5d81610dda565b8
2525050565b600060208201905061 0c78600830184610ba7565b929150505
65b600060a082019050610c936000830188610ba7565b610ca060208301876
10ba7565b610cad6040830186610c54565b610cba6060830185610c54565b6
10cc76080830184610c54565b969550505050505050565b60006040820190506
10ce66000830185610ba7565b610cf36020830184610c54565b93925050505
65b600060208201905061 0d0f6000830184610bb6565b92915050565b60006
02082019050610d2a6000830184610bc5565b92915050565b6000602082019
0508181036000830152610d4981610bd4565b9050919050565b60006020820
1905081810360008301526 10d6981610c14565b9050919050565b600060208
2019050610d856000830184610c54565b92915050565b60008282602082 0
1905092915050565b6000610da782610dba565b9050919050565b600081151
59050919050565b600073ffffffffffffffffffffffffffffffffffffffff8
2169050919050565b60008190509190505 65b6000610def82610df6565b905
0919050565b6000610e0182610dba565b9050919050565b610e1181610d9c5
65b8114610e1c57600080fd5b50565b610e2881610dae565b8114610e33576
00080fd5b50565b610e3f81610dda565b8114610e4a57600080fd5b5056fea
2646970667358221220 8a8a2d65bde047a11d42eeb90ebce9ea9eed3856369
7d75f3fef84f6bdfc6b4d64736f6c6343000800003300000000000000000000
00000deadbeefdeadbeefdeadbeefdeadbeefdeadbeef0000000000000000000
0000000000000000000000000000000000000000000000000…**mais
zeros…**', 'value': '0x0', 'origin':
'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe', 'address': ''},
{'input':
'0x21e6b53d0000000000000000000000000000000000000000000000000000
0000000000001', 'value': '0x0', 'origin':
'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe', 'address':
'0x901d12ebe1b195e5aa8748e62bd7734ae19b51f'}]

```
min_price_dict =
{100475310549029526324481294656594819817774295859 0: 0}

min_price_dict =
{100475310549029526324481294656594819817774295859 0: 0,
127127061300004165581744834813227589066893754095 0: 0}

concrete_initial_state = {'accounts':
{'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe': {...},
'0xdeadbeefdeadbeefdeadbeefdeadbeefdeadbeef': {...}}}

code = <mythril.disassembler.disassembly.Disassembly object at
0x7f758b50e240>

concrete_transaction = {'input':
'0x21e6b53d0000000000000000000000000000000000000000000000000000
0000000000001', 'value': '0x0', 'origin':
'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe', 'address':
'0x901d12ebe1b195e5aa8748e62bd7734ae19b51f', 'calldata':
'0x21e6b53d0000000000000000000000000000000000000000000000000000
0000000000001'}

concrete_transactions = [{'input':
'0x6080604052348015610010576000080fd5b50604051610fd2380380610fd
2833981810160405281019061003291906100ce565b336000806101000a815
48173ffffffffffffffffffffffffffffffffffffffff021916908373ffffff
ffffffffffffffffffffffffffffffffffff160217905550806001600061010
00a81548173ffffffffffffffffffffffffffffffffffffffff02191690837
3ffffffffffffffffffffffffffffffffffffffff160217905550506101405
65b600081519050610c881610129565b92915050565b60006020828403121
56100e057600080fd5b60006100ee848285016100b9565b915050929150505
65b600061010282610109565b9050919050565b600073ffffffffffffffff
ffffffffffffffffffffffff8216905091905056 5b610132816100f7565b811
461013d57600080fd5b50565b610e838061014f6000396000f3fe608060405
26004361061008a5760003560e01c8063affed0e011610059578063affed0e
014610150578063d493b9ac1461017b578063f2fde38b146101a4578063f85
1a440146101cd578063fc0c546a146101f857610091565b806321e6b53d146
100965780638326acce146100bf57806391dae519146100e8578063a090588
014610125576100915 65b3661009157005b600080fd5b3480156100a257600
```

```
080fd5b506100bd60048036038101906100b89190610a65565b610223565b0
05b3480156100cb57600080fd5b506100e660048036038101906100e191906
10add565b610345565b005b3480156100f457600080fd5b5061010f6004803
60381019061010a9190610b55565b610652565b60405161011c9190610cfa5
65b60405180910390f35b34801561013157600080fd5b5061013a610672565
b604051610147919061 0c63565b60405180910390f35b34801561015c57600
080fd5b506101656106985 65b604051610172919061 0d70565b60405180910
390f35b34801561018757600080fd5b506101a2600480360381019061019d9
190610a8e565b61069e565b005b3480156101b057600080fd5b506101cb600
48036038101906101c69190610a65565b610878565b005b3480156101d9576
00080fd5b506101e26109c7565b6040516101ef9190610c63565b604051809
10390f35b34801561020457600080fd5b5061020d6109eb565b60405161021
a9190610d15565b60405180910390f35b600073ffffffffffffffffffffffff
ffffffffffffffff168173ffffffffffffffffffffffffffffffffffffffff
f16141561025d57600080fd5b60008054906101000a900473ffffffffffff
ffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffff
ffffffff163373ffffffffffffffffffffffffffffffffffffffff1614610
2b557600080fd5b60016000905490610100 0a900473ffffffffffffffffff
ffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffff
fff1663f2fde38b826040518263ffffffff1660e01b81526004016103109 19
0610c63565b60006040518083038160008 7803b15801561032a57600080fd5
b505af115801561033e573d6000803e3d6000fd5b5050505050565b6000309
0506000805490610100 0a900473ffffffffffffffffffffffffffffffffff
fffff1673ffffffffffffffffffffffffffffffffffffffff163373ffffffff
ffffffffffffffffffffffffffffffff16146103d8576040517f08c379a00
0000000000000000000000000000000000000000000000000008152600
4016103cf90610d30565b6040518091039 0fd5b60001515600460008481526
020019081526020016000206000905490610100 0a900460ff1615151461043
f576040517f08c379a0000000000000000000000000000000000000000000
0000000000008152600401610436906 10d50565b6040518091039 0fd5b826
00160009054906101000a900473ffffffffffffffffffffffffffffffffff
fffff1673ffffffffffffffffffffffffffffffffffffffff166370a082318
36040518263ffffffff1660e01b81526004016104 9b9190610c63565b60206
040518083038160008 7803b1580156104b557600080fd5b505af1158015610
4c9573d6000803e3d6000fd5b505050506040513d601f19601f82011682018
06040525081019061 06104ed9190610b7e565b10156104f857600080fd5b60016
00460008481526020019081526020016000206000 6101000a81548160ff021
916908315150217905550600160009054906101000a900473ffffffffffff
ffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffff
```

```
fffffffff1663a9059cbb85856040518363ffffffff1660e01b81526004016
10581929190610cd1565b60206040518083038160008780 3b15801561059b5
7600080fd5b505af11580156105af573d6000803e3d6000fd5b50505050604
0513d601f19601f820116820180604052508101906105d39190610b2c565b5
0600180811115610 60d577f4e487b7100000000000000000000000000000000
0000000000000000000000000600052602160045260246000fd5b7f2775754
2a5e1b9e8cef80f584e094d4eb63b9802f355c61b3640b71b618d5c8e82868
64287604051610644959493929190610c7e565b60405180910390a25050505
0565b600460205280600052604060002060009150549061010 00a900460ff1
681565b60036000905490610100 0a900473ffffffffffffffffffffffffff
ffffffffffff1681565b60025481565b600080549061010 00a900473fffff
ffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffff
ffffffffffffffff163373ffffffffffffffffffffffffffffffffffffffff
f161461072c576040517f08c379a0000000000000000000000000000000000
0000000000000000000081526004016107239061 0d30565b60405180910
390fd5b806001600090549061010 0a900473ffffffffffffffffffffffff
ffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff166
370a08231306040518263ffffffff1660e01b81526004016107889190610c6
3565b602060405180830381600087803b1580156107a257600080fd5b505af
11580156107b6573d6000803e3d6000fd5b505050506040513d601f19601f8
2011682018060405250810190610 7da9190610b7e565b10156107e55760008
0fd5b8273ffffffffffffffffffffffffffffffffffffffff1663a9059cbb8
3836040518363ffffffff1660e01b81526004016108209291906 10cd1565b6
0206040518083038160008780 3b15801561083a57600080fd5b505af115801
561084e573d6000803e3d6000fd5b505050506040513d601f19601f8201168
2018060405250810190610 8729190610b2c565b50505050565b600073fffff
ffffffffffffffffffffffffffffffff168173ffffffffffffffffffff
ffffffffffffffff1614156108b257600080fd5b60008054906101000a9
00473ffffffffffffffffffffffffffffffffffffffff1673ffffffffffff
ffffffffffffffffffffff163373ffffffffffffffffffffffffffffffff
fffffffffff161461090a57600080fd5b8073ffffffffffffffffffffffff
ffffffffffffff166000805490610100 0a900473ffffffffffffffffffff
ffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff
f167f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6
b6457e06040516040518091039 0a38060008061010 00a81548173ffffffff
ffffffffffffffffffffffffffff02191690837 3ffffffffffffffffff
ffffffffffffffffff1602179055505056 5b60008054906101000a90047
3ffffffffffffffffffffffffffffffffffffffff1681565b6001600090549
06101000a900473ffffffffffffffffffffffffffffffffffffffff1681565
```

```
b600081359050610a2081610e08565b92915050565b600081519050610a358
1610e1f565b92915050565b600081359050610a4a81610e36565b929150505
65b600081519050610a5f81610e36565b92915050565b60006020828403121
5610a7757600080fd5b6000610a8584828501610a11565b915050929150505
65b60008060006060848031215610aa357600080fd5b6000610ab18682870
1610a11565b9350506020610ac286828701610a11565b9250506040610ad38
6828701610a3b565b915050925092509256 5b60008060006060848603121 56
10af257600080fd5b6000610b0086828701610a11565b9350506020610b118
6828701610a3b565b9250506040610b2286828701610a3b565b91505092509
25092565b600060208284031215610b3e57600080fd5b6000610b4c8482850
1610a26565b9150509291505056 5b600060208284031215610b6757600080f
d5b6000610b7584828501610a3b565b9150509291505056 5b6000602082840
31215610b9057600080fd5b6000610b9e84828501610a50565b91505092915
050565b610bb081610d9c565b82525050565b610bbf81610dae565b8252505
0565b610bce81610de4565b82525050565b6000610be1600a83610d8b565b9
1507f6f6e6c792061646d696e0000000000000000000000000000000000000
0000000600083015260208201905091905056 5b6000610c21601a83610d8b5
65b91507f7472616e7366657220616c72656164792070726f6365737365640
0000000000060008301526020820190509190505 65b610c5d81610dda565b8
2525050565b600060208201905061 0c786000830184610ba7565b929150505
65b600060a08201905061 0c936000830188610ba7565b610ca060208301876
10ba7565b610cad6040830186610c54565b610cba6060830185610c54565b6
10cc76080830184610c54565b969550505050505050565b60006040820190506
10ce66000830185610ba7565b610cf36020830184610c54565b93925050505
65b600060208201905061 0d0f6000830184610bb6565b92915050565b60006
02082019050610d2a6000830184610bc5565b92915050565b6000602082019
0508181036000830152610d4981610bd4565b9050919050565b60006020820
1905081810360008301526 10d6981610c14565b9050919050565b600060208
2019050610d856000830184610c54565b92915050565b60008282526020820
1905092915050565b6000610da782610dba565b9050919050565b600081151
59050919050565b600073ffffffffffffffffffffffffffffffffffffffff8
2169050919050565b60008190509190505 65b6000610def82610df6565b905
0919050565b6000610e0182610dba565b9050919050565b610e1181610d9c5
65b8114610e1c57600080fd5b50565b610e2881610dae565b8114610e33576
00080fd5b50565b610e3f81610dda565b8114610e4a57600080fd5b5056fea
2646970667358221208a8a2d65bde047a11d42eeb90ebce9ea9eed3856369
7d75f3fef84f6bdfc6b4d64736f6c6343000800003300000000000000000000
00000deadbeefdeadbeefdeadbeefdeadbeefdeadbeef0000000000000000000
0000000000000000000000000000000000000000000000…mais zeros…',
```

```
'value': '0x0', 'origin':
'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe', 'address': '',
'calldata':
'0000000000000000000000deadbeefdeadbeefdeadbeefdeadbeefdeadb
eef0000000000000000000000000000000000000000000000000000…ma
is zeros…'}, {'input':
'0x21e6b53d0000000000000000000000000000000000000000000000000000
0000000000001', 'value': '0x0', 'origin':
'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe', 'address':
'0x901d12ebe1b195e5aa8748e62bd7734ae19b51f', 'calldata':
'0x21e6b53d0000000000000000000000000000000000000000000000000000
0000000000001'}]


steps = {'initialState': {'accounts': {...}}, 'steps': [{...},
{...}]}

steps['initialState']  = {'accounts':
{'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe': {...},
'0xdeadbeefdeadbeefdeadbeefdeadbeefdeadbeef': {...}}}

steps['steps'] = [{'input':
'0x6080604052348015610010576000080fd5b50604051610fd2380380610fd
2833981810160405281019061003291906100ce565b336000806101000a815
48173ffffffffffffffffffffffffffffffffffffffff021916908373ffffff
ffffffffffffffffffffffffffffffffffff160217905550806001600061010
00a81548173ffffffffffffffffffffffffffffffffffffffff02191690837
3ffffffffffffffffffffffffffffffffffffffff160217905550506101405
65b600081519050610c881610129565b92915050565b60006020828403121
56100e057600080fd5b60006100ee848285016100b9565b915050929150505
65b600061010282610109565b9050919050565b600073ffffffffffffffff
ffffffffffffffffffffffff82169050919050565b610132816100f7565b811
461013d57600080fd5b50565b610e838061014f6000396000f3fe608060405
26004361061008a5760003560e01c8063affed0e011610059578063affed0e
014610150578063d493b9ac1461017b578063f2fde38b146101a4578063f85
1a440146101cd578063fc0c546a146101f857610091565b806321e6b53d146
100965780638326acce146100bf57806391dae519146100e8578063a090588
014610125576100915b3661009157005b600080fd5b3480156100a257600
080fd5b506100bd600480360381019061009b9190610a65565b610223565b0
```

```
05b3480156100cb57600080fd5b506100e660048036038101906100e191906
10add565b610345565b005b3480156100f457600080fd5b5061010f6004803
60381019061010a9190610b55565b610652565b60405161011c9190610cfa5
65b60405180910390f35b34801561013157600080fd5b5061013a610672565
b60405161014791906 10c63565b60405180910390f35b34801561015c57600
080fd5b5061016561069856 5b60405161017291906 10d70565b60405180910
390f35b34801561018757600080fd5b506101a260048036038101906101d9
190610a8e565b61069e565b005b3480156101b057600080fd5b506101cb600
48036038101906101c69190610a65565b610878565b005b3480156101d9576
00080fd5b506101e26109c7565b6040516101ef9190610c63565b604051809
10390f35b34801561020457600080fd5b5061020d6109eb565b60405161021
a9190610d15565b60405180910390f35b600073ffffffffffffffffffffff
ffffffffffffffff168173ffffffffffffffffffffffffffffffffffffffff
f16141561025d57600080fd5b60008054906101000a900473ffffffffffff
ffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffff
ffffffff163373ffffffffffffffffffffffffffffffffffffffff1614610
2b557600080fd5b60016000905490610100 0a900473ffffffffffffffffff
ffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffff
fff1663f2fde38b826040518263ffffffff1660e01b8152600401610310919
0610c63565b60006040518083038160008 7803b15801561032a57600080fd5
b505af115801561033e573d6000803e3d6000fd5b5050505050565b6000309
0506000805490610100 0a900473ffffffffffffffffffffffffffffffffff
fffff1673ffffffffffffffffffffffffffffffffffffffffff163373ffffff
ffffffffffffffffffffffffffffff16146103d8576040517f08c379a00
0000000000000000000000000000000000000000000000000008152600
4016103cf90610d30565b60405180910390fd5b600015156004600848152 6
020019081526020016000206000905490610100 0a900460ff1615151461043
f576040517f08c379a0000000000000000000000000000000000000000000
0000000000008152600401610436 90610d50565b60405180910390fd5b826
00160009054906101000a900473ffffffffffffffffffffffffffffffffff
fffff1673ffffffffffffffffffffffffffffffffffffffffff166370a082318
36040518263ffffffff1660e01b8152600401610 49b9190610c63565b60206
040518083038160008 7803b1580156104b557600080fd5b505af1158015610
4c9573d6000803e3d6000fd5b505050506040513d601f19601f82011682018
060405250810190610 4ed9190610b7e565b10156104f857600080fd5b60016
004600084815260020019081526020016000206000 6101000a81548160ff021
916908315150217905550600160009054906101000a900473ffffffffffff
ffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffff
ffffffff1663a9059cbb8585604051836 3ffffffff1660e01b81526004016
```

```
10581929190610cd1565b602060405180830381600087803b15801561059b5
7600080fd5b505af11580156105af573d6000803e3d6000fd5b50505050604
0513d601f19601f820116820180604052508101906105d39190610b2c565b5
0600180811115610d577f4e487b7100000000000000000000000000000000
0000000000000000000000000600052602160045260246000fd5b7f2775754
2a5e1b9e8cef80f584e094d4eb63b9802f355c61b3640b71b618d5c8e82868
642876040516106449594939291906l0c7e565b60405180910390a25050505
0565b600460205280600052604060002060009150549061010000a900460ff1
681565b600360009054906101000a900473ffffffffffffffffffffffffff
ffffffffffff1681565b60025481565b600080549061010000a900473fffff
fffffffffffffffffffffffffffffff1673ffffffffffffffffffffffff
ffffffffffffffff163373ffffffffffffffffffffffffffffffffffffff
f161461072c576040517f08c379a0000000000000000000000000000000
0000000000000000000081526004016107239061d30565b6040518091
390fd5b806001600090549061010000a900473ffffffffffffffffffffffff
ffffffffffffff1673ffffffffffffffffffffffffffffffffffffff166
370a08231306040518263ffffffff1660e01b81526004016107889190610c6
3565b602060405180830381600087803b1580156107a257600080fd5b505af
11580156107b6573d6000803e3d6000fd5b505050506040513d601f19601f8
2011682018060405250810190610da9190610b7e565b10156107e55760008
0fd5b8273ffffffffffffffffffffffffffffffffffffffff1663a9059cbb8
3836040518363ffffffff1660e01b8152600401610820929190610cd1565b6
0206040518083038160008780 3b15801561083a57600080fd5b505af115801
561084e573d6000803e3d6000fd5b505050506040513d601f19601f8201168
2018060405250810190610872919061 0b2c565b50505050565b600073fffff
fffffffffffffffffffffffffffffff168173ffffffffffffffffffff
ffffffffffffffff1614156108b257600080fd5b6000805490610100 0a9
00473ffffffffffffffffffffffffffffffffffffff1673ffffffffffff
ffffffffffffffffffff163373ffffffffffffffffffffffffffffffff
fffffffffff161461090a57600080fd5b8073ffffffffffffffffffffffff
ffffffffffffff16600080549061010000a900473ffffffffffffffffffff
ffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff
f167f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6
b6457e0604051604051809103906000806101000a81548173ffffffff
ffffffffffffffffffffffffffff02191690837 3ffffffffffffffffff
ffffffffffffffffffff16021790555050565b600080549061010000a90047
3ffffffffffffffffffffffffffffffffffffff1681565b6001600090549
06101000a900473ffffffffffffffffffffffffffffffffffffffff1681565
b600081359050610a2081610e08565b92915050565b600081519050610a358
```

1610e1f565b92915050565b600081359050610a4a81610e36565b929150505
65b600081519050610a5f81610e36565b92915050565b60006020828403121
5610a7757600080fd5b6000610a8584828501610a11565b915050929150505
65b60008060006060848603121561 0aa357600080fd5b6000610ab18682870
1610a11565b9350506020610ac286828701610a11565b9250506040610ad38
6828701610a3b565b91505092509250925092565b60008060006060848603121 56
10af257600080fd5b6000610b0086828701610a11565b9350506020610b118
6828701610a3b565b9250506040610b2286828701610a3b565b91505092509
25092565b60006020828403121561 0b3e57600080fd5b6000610b4c8482850
1610a26565b91505092915050565b60006020828403121561 0b6757600080f
d5b6000610b7584828501610a3b565b91505092915050565b6000602082840
31215610b9057600080fd5b6000610b9e84828501610a50565b91505092915
050565b610bb081610d9c565b82525050565b610bbf81610dae565b8252505
0565b610bce81610de4565b82525050565b6000610be1600a83610d8b565b9
1507f6f6e6c792061646d696e0000000000000000000000000000000000000
0000000600083015260208201905091905056 5b6000610c21601a83610d8b5
65b91507f7472616e7366657220616c72656164792070726f6365737365640
0000000000060008301526020820190509190505 65b610c5d81610dda565b8
2525050565b60006020820190506 10c786000830184610ba7565b929150505
65b600060a082019050610c936000830188610ba7565b610ca060208301876
10ba7565b610cad6040830186610c54565b610cba6060830185610c54565b6
10cc76080830184610c54565b969550505050505050565b60006040820190506
10ce66000830185610ba7565b610cf36020830184610c54565b93925050505
65b60006020820190506 10d0f6000830184610bb6565b92915050565b60006
02082019050610d2a6000830184610bc5565b92915050565b6000602082019
0508181036000830152610d4981610bd4565b9050919050565b60006020820
1905081810360008301526 10d6981610c14565b9050919050565b600060208
2019050610d856000830184610c54565b92915050565b60008282526020820
1905092915050565b6000610da782610dba565b9050919050565b600081151
59050919050565b600073ffffffffffffffffffffffffffffffffffffffff8
2169050919050565b6000819050919050565b6000610def82610df6565b905
0919050565b6000610e0182610dba565b9050919050565b610e1181610d9c5
65b8114610e1c57600080fd5b50565b610e2881610dae565b8114610e33576
00080fd5b50565b610e3f81610dda565b8114610e4a57600080fd5b5056fea
2646970667358221220 8a8a2d65bde047a11d42eeb90ebce9ea9eed3856369
7d75f3fef84f6bdfc6b4d64736f6c6343000800003300000000000000000000
00000deadbeefdeadbeefdeadbeefdeadbeefdeadbeef0000000000000000000
0000000000000000000000000000000000000000000000000000000000000
0000000000…mais zeros…', 'value': '0x0', 'origin':

```
'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe', 'address': '',
'calldata':
'0000000000000000000000deadbeefdeadbeefdeadbeefdeadbeefdeadb
eef0000000000000000000000000000000000000000000000000000…m
ais zeros…'}, {'input':
'0x21e6b53d0000000000000000000000000000000000000000000000000
0000000000001', 'value': '0x0', 'origin':
'0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe', 'address':
'0x901d12ebe1b195e5aa8748e62bd7734ae19b51f', 'calldata':
'0x21e6b53d0000000000000000000000000000000000000000000000000
0000000000001'}]
```

Fluxo de Execução e Análise:

1. **Inicialização:**
    - O código recebe como entrada o estado global (`global_state`) e as constraints.
    - `transaction_sequence` contém transações como `ContractCreationTransaction` e `MessageCallTransaction`.
    - `concrete_transactions` é uma lista inicializada vazia para armazenar as transações concretas.
2. **Processamento das Constraints:**
    - As constraints incluem verificações importantes, como:
        - **Gás:** `UGT(2_gas, 2300)`
        - **Tamanho de `calldata`:** `1_calldatasize == 4562`
        - **Balanço e valores de chamada:** `call_value1 == 0`
        - **Endereços específicos:** Condições sobre remetentes e destinatários.
        - **Dados da chamada:** Restrições múltiplas sobre o conteúdo do `calldata`.
3. **Minimização:**
    - O tuple `minimize = (1_calldatasize, call_value1, 2_calldatasize, call_value2)` guia o solver para gerar valores mais realistas e eficientes.
4. **Solver e Modelo:**
    - O solver tenta obter um modelo satisfatório usando:

```python
get_model(tx_constraints, minimize=minimize)
```

- Se as constraints não forem satisfeitas, uma `UnsatError` é lançada.
- O modelo gerado contém valores concretos para todas as variáveis simbólicas das constraints.

Relação do Solver com as Constraints:

1. **Função do Solver:**
   - O solver analisa as constraints para encontrar valores concretos que satisfaçam todas elas. As constraints representam condições que devem ser verdadeiras para explorar vulnerabilidades no contrato.
2. **Valores Gerados pelo Solver:**
   - Gás: `2_gas = 10494`
   - `calldata` e `calldatasize`
   - Valores zero para `call_value1` e `call_value2`
   - Endereços específicos (`sender_2`)
   - Estados de balanço inicial
3. **Saída Final:**
   - Estado inicial concreto
   - Sequência de transações concretizadas
   - Valores concretos para todas as variáveis simbólicas

Construção do Estado Inicial e Processamento de Transações:

1. Para a `ContractCreationTransaction`:
   - Input: Bytecode do contrato (`0x608060405234...`)
   - Origin: `0xaffeaffeaffeaffeaffeaffeaffeaffeaffeaffe`
   - Value: `0x0`
   - Address: ""
2. Para a transação subsequente:
   - Input: Chamada de função (`0x21e6b53d...`)
   - Origin: Mesmo endereço
   - Address: `0x901d12ebe1b195e5aa8748e62bd7734ae19b51f`

3. Criação do estado inicial:

```python
concrete_initial_state = _get_concrete_state(initial_accounts, min_price_dict)
```

    - Gera o estado inicial concreto com contas e balanços específicos.

Conexão com o Mythril: O Mythril utiliza o solver para traduzir estados simbólicos em valores concretos, permitindo identificar vulnerabilidades potenciais no contrato. Ele valida as transações, mantém a consistência do blockchain e verifica condições como balanço, inputs, origem e destino.