Código 1: `solver.get_transaction_sequence()`

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

Dados de debug:

```
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]))]

state.world_state.constraints
=

[Or(Not(ULE(balance[1004753105490295263244812946565948198177742958590],
           call_value1)),
   balance[1004753105490295263244812946565948198177742958590] ==
   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,
                       514214400560557283460174190016654010742164493 11,

balance[514214400560557283460174190016654010742164493 11] +
                       call_value1),
                 1004753105490295263244812946565948198177742958590,
                 balance[1004753105490295263244812946565948198177742958590]
+

115792089237316195423570985008687907853269984665640564039457584007913129639
35*
                 call_value1)[sender_2],
           call_value2)),
   Store(Store(balance,
               514214400560557283460174190016654010742164493 11,
               balance[514214400560557283460174190016654010742164493 11] +
               call_value1),
         1004753105490295263244812946565948198177742958590,
         balance[1004753105490295263244812946565948198177742958590] +

115792089237316195423570985008687907853269984665640564039457584007913129639
35*
         call_value1)[sender_2] ==
   call_value2), Or(sender_2 ==
   1004753105490295263244812946565948198177742958590,
   sender_2 ==
   1271270613000041655817448348132275889066893754095,
   sender_2 ==
   974334424887268612135789888477522013103955028650), 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 <=
115792089237316195423570985008687907853269984665640564039457584007913129639932 +
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)]

state =
<mythril.laser.ethereum.state.global_state.GlobalState object at
0x7f758b500770>
```

Fluxo de Execução:

1. A função recebe dois argumentos principais:
    - `state`: Um objeto `GlobalState` que representa o estado atual da execução.
    - **Uma soma de duas listas de constraints**:
        - `constraints`: Lista com restrições específicas da execução atual.
        - `state.world_state.constraints`: Lista com restrições do estado global.
2. O operador **+** aciona o método `__add__` da classe `Constraints`, levando ao **Código 2**.

Relação com Constraints:

- Este código é responsável por **juntar todas as restrições** que serão usadas pelo solver.
- As **constraints** incluem verificações de:
    - **Tamanho e conteúdo de `calldata`**.
    - **Valores de `gas`**.
    - **Balanços de contas**.
    - **Restrições de estado do contrato**.
    - **Validações de endereços**.

Código 2: Método `__add__`

```Python
def __add__(self, constraints: List[Union[bool, Bool]]) ->
"Constraints":
    constraints_list = self._get_smt_bool_list(constraints)
    constraints_list = super(Constraints,
self).__add__(constraints_list)
    return Constraints(constraint_list=constraints_list)
```

Dados de debug:

```Unset
constraints_list =
[Or(Not(ULE(balance[1004753105490295263244812946565948198177742958590],
           call_value1)),
   balance[1004753105490295263244812946565948198177742958590] ==
   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,
                       51421440056055728346017419001665401074216449311,

balance[51421440056055728346017419001665401074216449311] +
                       call_value1),
                 1004753105490295263244812946565948198177742958590,
                 balance[1004753105490295263244812946565948198177742958590]
+

115792089237316195423570985008687907853269984665640564039457584007913129639
35*
                 call_value1)[sender_2],
           call_value2)),
   Store(Store(balance,
               514214400560557283460174190016654010742164493111,
               balance[514214400560557283460174190016654010742164493111] +
               call_value1),
         1004753105490295263244812946565948198177742958590,
         balance[1004753105490295263244812946565948198177742958590] +

115792089237316195423570985008687907853269984665640564039457584007913129639935*
         call_value1)[sender_2] ==
   call_value2), Or(sender_2 ==
   1004753105490295263244812946565948198177742958590,
   sender_2 ==
   1271270613000041655817448348132275889066893754095,
   sender_2 ==
   974334424887268612135789888477522013103955028650), 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 <=
115792089237316195423570985008687907853269984665640564039457584007913129639932 +
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)]
constraints_list
na segunda atribuição:
[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[1004753105490295263244812946565948198177742958590],
           call_value1)),
   balance[1004753105490295263244812946565948198177742958590] ==
   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,
                       514214400560557283460174190016654010742164493311,

balance[514214400560557283460174190016654010742164493311] +
                       call_value1),
                 1004753105490295263244812946565948198177742958590,
                 balance[1004753105490295263244812946565948198177742958590]
+

115792089237316195423570985008687907853269984665640564039457584007913129639935*
                 call_value1)[sender_2],
           call_value2)),
   Store(Store(balance,
               514214400560557283460174190016654010742164493311,
               balance[514214400560557283460174190016654010742164493311] +
               call_value1),
         1004753105490295263244812946565948198177742958590,
         balance[1004753105490295263244812946565948198177742958590] +

115792089237316195423570985008687907853269984665640564039457584007913129639935*
         call_value1)[sender_2] ==
   call_value2), Or(sender_2 ==
   1004753105490295263244812946565948198177742958590,
   sender_2 ==
   1271270613000041655817448348132275889066893754095,
   sender_2 ==
   974334424887268612135789888477522013103955028650), 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 <=
11579208923731619542357098500868790785326998466564056403945758400791312963993
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)]
```

Fluxo de Execução:

1. Recebe uma lista de **constraints** para adicionar.
2. Chama `_get_smt_bool_list` para converter as constraints para o formato SMT (**Código 3**).
3. Usa a implementação base de `list.__add__` para concatenar as listas.
4. Cria e retorna um novo objeto `Constraints` com a lista concatenada.

Código 3: Método `_get_smt_bool_list`

```python
@staticmethod
def _get_smt_bool_list(constraints: Iterable[Union[bool,
Bool]]) -> List[Bool]:
    return [
        (constraint if isinstance(constraint, Bool) else
symbol_factory.Bool(constraint))
        for constraint in constraints
    ]
```

Fluxo de Execução:

1. Recebe uma lista de **constraints** que podem ser:
    - Booleanos Python.
    - Objetos **`Bool`** do SMT.
2. Para cada **constraint**:
    - **Se já for um `Bool` SMT**, mantém como está.
    - **Se for um booleano Python**, converte para `Bool` SMT usando **`symbol_factory`**.

Código 4: Classe `Constraints`

```python
class Constraints(list):
    def __init__(self, constraint_list: Optional[List[Bool]] =
None) -> None:
        constraint_list = constraint_list or []
        constraint_list =
self._get_smt_bool_list(constraint_list)
        super(Constraints, self).__init__(constraint_list)
```

Fluxo de Execução:

1. Inicializa uma nova lista de **constraints**.
2. Se nenhuma lista for fornecida, usa uma lista vazia.
3. Converte todas as **constraints** para o formato SMT.
4. Inicializa a classe base **`list`** com as **constraints** convertidas.

Propósito Final:

Todo este fluxo serve para:

1. **Coletar todas as constraints relevantes** do estado atual e global.
2. **Garantir que todas as constraints estejam no formato correto** (`Bool` SMT).
3. **Combinar todas as constraints em uma única lista consistente**.
4. Fornecer estas **constraints** para o solver SMT que irá:
   - Verificar se existe alguma sequência de transações que **satisfaz todas as constraints**.
   - **Identificar possíveis vulnerabilidades** no contrato.
   - **Gerar casos de teste** que demonstrem as vulnerabilidades encontradas.