Machine State

Notation : `machine_state`

Description : The machine state is a tuple consisting of five elements:

1. `gas_available`
2. `program_counter`
3. `memory_contents` A series of zeroes of size 2^{256}
4. `memory_words.count`
5. `stack_contents`

There is also, [to_execute]: the current operation to be executed

0.0.1 Exceptional Halting

An exceptional halt may be caused by a handful of boolean values:

```
forall instruction.x
if gas_empty = true
then signal halt
elif instruction.x = fake
then signal halt
elif stack = terse
then signal halt
elif jumpdest = bad
then signal halt
else exec instruction.x

forall instruction.y
[...]
[...]
[...]
[...]
```

```
forall instruction.z
[...]
[...]
[...]
[...]

then signal controlled_halt
```

No instruction can, through its execution, cause an exceptional halt. They can only happen if some instruction, for whatever reason, fails to execute.

- The amount of remaining gas in each transaction is extracted from information contained in the `machine_state`
- A simple iterative recursive loop[1] with a boolean value:

  true indicating that in the run of computation, an exception was signaled

  false indicating in the run of computation, exceptions were signaled. If this value remains false for the duration of the execution until the set of transactions becomes a series (rather than an empty set.) This means that the machine has reached a controlled halt.

References

[1] D. G. Wood, *Ethereum: A secure decentralised generalised transaction ledger*, `https://github.com/ethereum/yellowpaper`, 2017.