to the aborted contract's address when we are out-of-gas, thus the contract's code is not stored.

If such an exception does not occur, then the remaining gas is refunded to the originator and the now-altered state is allowed to persist. Thus formally, we may specify the resultant state, gas, accrued substate and status code as $(\boldsymbol{\sigma}', g', A', z)$ where:

(105)
$$g' \equiv \begin{cases} 0 & \text{if} \quad F \\ g^{**} - c & \text{otherwise} \end{cases}$$

(106)
$$\boldsymbol{\sigma}' \equiv \begin{cases} \boldsymbol{\sigma} & \text{if} \quad F \ \vee \ \boldsymbol{\sigma}^{**} = \varnothing \\ \boldsymbol{\sigma}^{**} \quad \text{except:} & \\ \quad \boldsymbol{\sigma}'[a] = \varnothing & \text{if} \quad \mathtt{DEAD}(\boldsymbol{\sigma}^{**}, a) \\ \boldsymbol{\sigma}^{**} \quad \text{except:} & \\ \quad \boldsymbol{\sigma}'[a]_{\mathrm{c}} = \mathtt{KEC}(\mathbf{o}) & \text{otherwise} \end{cases}$$

(107)
$$A' \equiv \begin{cases} A^{*} & \text{if} \quad F \ \vee \ \boldsymbol{\sigma}^{**} = \varnothing \\ A^{**} & \text{otherwise} \end{cases}$$

(108)
$$z \equiv \begin{cases} 0 & \text{if} \quad F \ \vee \ \boldsymbol{\sigma}^{**} = \varnothing \\ 1 & \text{otherwise} \end{cases}$$

where

(109)
$$\begin{aligned} F \equiv & \big(\boldsymbol{\sigma}[a] \neq \varnothing \ \wedge \ \big(\boldsymbol{\sigma}[a]_{\mathrm{c}} \neq \mathtt{KEC}\big(()\big) \vee \boldsymbol{\sigma}[a]_n \neq 0\big)\big) \quad \vee \\ & (\boldsymbol{\sigma}^{**} = \varnothing \ \wedge \ \mathbf{o} = \varnothing) \quad \vee \\ & g^{**} < c \quad \vee \\ & \|\mathbf{o}\| > 24576 \end{aligned}$$

The exception in the determination of $\boldsymbol{\sigma}'$ dictates that $\mathbf{o}$, the resultant byte sequence from the execution of the initialisation code, specifies the final body code for the newly-created account.

Note that intention is that the result is either a successfully created new contract with its endowment, or no new contract with no transfer of value. In addition, observe that if the execution of the initialising code reverts ($\boldsymbol{\sigma}^{**} = \varnothing \ \wedge \ \mathbf{o} \neq \varnothing$), the resultant gas g' is not depleted (provided there was no other exception), but no new account is created.

7.1. Subtleties.

Note that while the initialisation code is executing, the newly created address exists but with no intrinsic body code[5]. Thus any message call received by it during this time causes no code to be executed. If the initialisation execution ends with a SELFDESTRUCT instruction, the matter is moot since the account will be deleted before the transaction is completed. For a normal STOP code, or if the code returned is otherwise empty, then the state is left with a zombie account, and any remaining balance will be locked into the account forever.

8. Message Call

In the case of executing a message call, several parameters are required: sender (s), transaction originator (o), recipient (r), the account whose code is to be executed (c, usually the same as recipient), available gas (g), value (v) and gas price (p) together with an arbitrary length byte array, $\mathbf{d}$, the input data of the call, the present depth of the message-call/contract-creation stack (e) and finally the permission to make modifications to the state (w).

Aside from evaluating to a new state and accrued transaction substate, message calls also have an extra component—the output data denoted by the byte array $\mathbf{o}$. This is ignored when executing transactions, however message calls can be initiated due to VM-code execution and in this case this information is used.

(110)
$$(\boldsymbol{\sigma}', g', A', z, \mathbf{o}) \equiv \Theta(\boldsymbol{\sigma}, A, s, o, r, c, g, p, v, \tilde{v}, \mathbf{d}, e, w)$$

Note that we need to differentiate between the value that is to be transferred, v, from the value apparent in the execution context, $\tilde{v}$, for the DELEGATECALL instruction.

We define $\boldsymbol{\sigma}_1$, the first transitional state as the original state but with the value transferred from sender to recipient:

$$\boldsymbol{\sigma}_1[r]_{\mathrm{b}} \equiv \boldsymbol{\sigma}[r]_{\mathrm{b}} + v \quad \wedge \quad \boldsymbol{\sigma}_1[s]_{\mathrm{b}} \equiv \boldsymbol{\sigma}[s]_{\mathrm{b}} - v \tag{111}$$

unless $s = r$.

Throughout the present work, it is assumed that if $\boldsymbol{\sigma}_1[r]$ was originally undefined, it will be created as an account with no code or state and zero balance and nonce. Thus the previous equation should be taken to mean:

$$\boldsymbol{\sigma}_1 \equiv \boldsymbol{\sigma}_1' \quad \text{except:} \tag{112}$$

$$\boldsymbol{\sigma}_1[s] \equiv \begin{cases} \varnothing & \text{if } \boldsymbol{\sigma}_1'[s] = \varnothing \ \wedge \ v = 0 \\ \mathbf{a}_1 & \text{otherwise} \end{cases} \tag{113}$$

$$\mathbf{a}_1 \equiv \big(\boldsymbol{\sigma}_1'[s]_{\mathrm{n}}, \boldsymbol{\sigma}_1'[s]_{\mathrm{b}} - v, \boldsymbol{\sigma}_1'[s]_{\mathbf{s}}, \boldsymbol{\sigma}_1'[s]_{\mathrm{c}}\big) \tag{114}$$

$$\text{and} \quad \boldsymbol{\sigma}_1' \equiv \boldsymbol{\sigma} \quad \text{except:} \tag{115}$$

(116)
$$\begin{cases} \boldsymbol{\sigma}_1'[r] \equiv (0, v, \mathtt{TRIE}(\varnothing), \mathtt{KEC}(())) & \text{if} \quad \boldsymbol{\sigma}[r] = \varnothing \wedge v \neq 0 \\ \boldsymbol{\sigma}_1'[r] \equiv \varnothing & \text{if} \quad \boldsymbol{\sigma}[r] = \varnothing \wedge v = 0 \\ \boldsymbol{\sigma}_1'[r] \equiv \mathbf{a}_1' & \text{otherwise} \end{cases}$$

$$\mathbf{a}_1' \equiv (\boldsymbol{\sigma}[r]_{\mathrm{n}}, \boldsymbol{\sigma}[r]_{\mathrm{b}} + v, \boldsymbol{\sigma}[r]_{\mathbf{s}}, \boldsymbol{\sigma}[r]_{\mathrm{c}}) \tag{117}$$

The account's associated code (identified as the fragment whose Keccak-256 hash is $\boldsymbol{\sigma}[c]_{\mathrm{c}}$) is executed according to the execution model (see section 9). Just as with contract creation, if the execution halts in an exceptional fashion (i.e. due to an exhausted gas supply, stack underflow, invalid jump destination or invalid instruction), then no gas is refunded to the caller and the state is reverted to the point immediately prior to balance transfer (i.e. $\boldsymbol{\sigma}$).

[5]During initialization code execution, `EXTCODESIZE` on the address should return zero, which is the length of the code of the account while `CODESIZE` should return the length of the initialization code (as defined in H.2).