

state transaction functions - (transaction) $\Upsilon \rightarrow$ (give rewards) $\Omega \rightarrow$ (block) Π

$\sigma \rightarrow \sigma_0 \rightarrow \sigma_P \rightarrow \sigma^* \rightarrow \sigma'$
 $\sigma \rightarrow \sigma'_1 \rightarrow \sigma_1 \rightarrow \sigma^{**} \rightarrow \sigma'$

Greek Alphabets		
α	stack items added	9.4.1, 9.5
α_w	!!!	9.4.2, 9.5
Δ	!!!	9.5
δ	stack items removed	9.4.1, 9.5
δ	number of stack items for a given operation	3, 9.4.2
δ_w	!!!	9.4.2, 9.5
ϵ	exponential difficulty symbol for difficulty bomb (or ice age)	4.3.4 (4.4.4)
$\Gamma()$	function that maps a block to its initiation state	11.4
μ	machine-state	3, 9.4.0, 9.4.1, 9.4.2, 9.5
μ'	resultant machine state	9.4.0(), 9.5
μ_g	!!!	9.4.0, 9.4.2
μ_i	!!!	9.4.0
μ'_i	!!!	9.5
μ_m	machine's memory	3, 9.4.0, 9.4.1
μ'_m	!!!	9.5
μ_o	!!!	9.4.0, 9.4.2()
μ_{pc}	!!! (\mathbb{N}_{256})	9.4.0, 9.4.1, 9.5
μ'_{pc}	!!!	9.5
μ_s	machine's stack	3, 9.4.0, 9.4.2, 9.5
μ'_s	!!!	9.5
μ_g	!!!	9.4.0()
μ'_g	remaining gas from the resultant machine state (μ')	9.4.0()
Λ	!!!	(6.2), 7
Λ_4	!!!	6.2()
$\Omega()$	block-finalisation state transition function that gives rewards	2.0, 11.3, 11.4
$\Pi()$	block-level state transition function	2.0, 4.3.2(4.4.2), 11.4
Φ	block transition function which maps an incom- plete block to a complete block	11.4
σ, σ_t	world-state, global state, sequence of accounts	2.0, 3, 4.3.2(4.4.2), 6.0, 7, 8, 9.3, 9.4.0, 9.4.2, 9.5, 11.3, 11.4
σ'	final (post-transactional) state	6.0, 6.2, 7, 8, 9.3, 9.4.0(), 9.5, 11.3
σ^*	pre-final state	6.2, 7
σ^{**}	resultant state	7, 8
σ_0	checkpoint state	6.2 , 11.4
σ_1	first transitional state	8
σ'_1	!!!	8
σ_i	!!!	11.4
σ_P	tuple of post-execution provisional state	6.2

$\sigma[a]$	account state	4.1, 6.2, 11.3(), 11.4()
$\sigma[a]_b$	balance, number of Wei owned by this address (\mathbb{N}_{256})	4.1, 6.2, 7, 8, 11.3
$\sigma[a]_c$	codeHash, hash of the EVM code of this address (\mathbb{N}_{32})	4.1, 7, 8, 11.3()
$\sigma[a]_n$	nonce, (non-contract account) number of trans- actions sent from this address or (contract ac- count) number of contract-creations made by this account (\mathbb{N}_{256})	4.1, 6.2, 7, 8(), 11.3()
$\sigma[a]_s$	storageRoot, root node hash of a Merkle Patri- cia tree that encodes the storage contents of this account (\mathbb{N}_{32})	4.1, 7(), 8(), 11.3()
ς_1	pre-Homestead difficulty parameter	(4.4.4)
ς_2	Homestead difficulty parameter	4.3.4(4.4.4)
Θ	!!!	8
Θ_3	!!!	(6.2)
Θ_4	!!!	6.2()
Υ	transaction-level state transition function	2.0, 3, 6.0, 11.4
$\Upsilon^g()$	total gas used in the execution of a transaction	6.0, 6.2, 11.4
$\Upsilon^l()$	log items created by this transaction	6.0, 6.2, 11.4
$\Upsilon^z()$	status code resulting from the transaction	6.0(), 6.2(), 11.4()
$\Xi(), \Xi_{\dots}()$	code execution function	7, 8, 9.3, 9.4.0

Mathematical Symbols

\equiv	defined to be	...
$\ \dots \ $	sequence length	...
$ \dots $	size of the code	7
\wedge	and	...
\vee	or	...
\bigvee	bitwise or	4.3.1(4.4.1), 4.3.2(4.4.2)
$()$	empty string	4.1, 6.1, 7, 9.4.0, 9.4.4
\emptyset	empty set	...
\top	!!!	6.2()
\neg	not	9.4.2()

Latin Alphabets

A	transaction substate	6.1, 6.2, 7, 8, 9.3(), 9.4.0, 9.5
A'	!!!	9.5
A^0	empty substate	6.1 , 9.4.0
A_1	log series, series of archived and indexable 'checkpoints'	6.1
A_r	refund balance which increases through using the SSTORE instruction in order to reset con- tract storage	6.1, 6.2
A_s	self-destruct (suicide) set: a set of accounts that <u>will be discarded</u>	6.1, 6.2
A_t	set of touched empty accounts which are deleted at the end of a transaction	6.1(), 6.2()

a	address of the new account	7
\mathbf{a}'	!!!	11.3()
\mathbf{a}^*	updated contract creation sender (see also s)	7
\mathbf{a}_1	!!!	8()
\mathbf{a}'_1	!!!	8()
\mathbf{a}_1	!!!	8()
\mathbf{a}'_1	!!!	8()
ADD	!!!	9.4.1
B	this block	2.0, 4.3.0(4.4.0), 4.3.2(4.4.2), 4.3.3(4.4.3), 6.2, 10, 11.3, 11.4
B_H	block header (see H)	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4), 11.1
B_{H_c}	beneficiary (H_c)	6.2
B_d	block's difficulty	10
\mathbf{B}_{H_c}	!!!	11.3
B_{H_g}	total gas used in the block (H_g)	11.2
B_{H_i}	!!! (H_i)	11.3
B_{H_1}	block's gasLimit (H_1)	6.2
B_R	transaction receipts in this block	4.3.1(4.4.1), 4.3.2(4.4.2), 6.2
B_T	a series of the transactions in this block	4.3.0(4.4.0), 4.3.2(4.4.2), 4.3.3(4.4.3), 11.4
B_t	total difficulty of block	10
B'_t	total difficulty of the parent block	10
B_U	list of ommer (uncle) block headers	4.3.0(4.4.0), 4.3.2(4.4.2), 4.3.3(4.4.3), 4.3.4(4.4.4), 11.1, 11.3
B'	parent block	10
B'	!!! complete block	11.4
B'_n	!!!	11.4
B'_m	!!!	11.4
B^*	!!!	11.4
$B_{\mathcal{N}}^*$!!!	11.4
B_r^*	!!!	11.4
$\mathbb{B}, \mathbb{B}_{number}$	byte sequence, of length $number$	3, 4.1, 4.2(4.3), 4.3.1(4.4.1), 4.3.3(4.4.3)
$\mathcal{B}_j(), \mathcal{B}_{a..b}()$	bit of index j or bits of indices in the range $[a, b]$ in the byte array	4.3.1 (4.4.1), 7
\mathbf{b}	EVM code	4.1
BLOCKHASH	!!!	12.2
$C, C_{\text{OPERATION}}$	cost function	3, 9.4.1
$C()$!!!	9.4.0, 9.4.2, 9.5
\mathbf{c}	!!!	9.4.3
c	code-deposit cost	7

c	account whose code is to be executed (the same as message call recipient)	8
CALL	!!!	9.2, 9.3, 9.4.2()
CALLCODE	!!!	9.2, 9.4.2()
CREATE	!!!	9.2, 9.3, 9.4.2()
$D()$	canonical difficulty of a block of the given header (see H_d)	4.3.4 (4.4.4)
$D()$	function to determine the set of valid jump destinations given the code that is being run	9.4.2, 9.4.3
$D_J()$!!!	9.4.3
D_0	difficulty of the genesis block	4.3.4 (4.4.4)
\mathfrak{D}	Ether, ETH	2.1
\mathbf{d}	current DAG for computing the mix-hash	4.3.4(4.4.4), 11.4, 11.5.0
\mathbf{d}	input data of the call	8
DEAD()	whether an account is dead, when its account state is non-existent or empty (EMPTY())	(4.1 , 6.2, 7)
DELEGATECALL	!!!	8
e	present depth of the message-call/contract-creation stack	7, 8
EMPTY()	whether an account is empty, when it has no code, zero nonce and zero balance	(4.1)
F	out-of-gas exception condition	(7)
G_{\dots}	specific gas cost	6.2
g	!!! (μ_g)	9.4.1
g	amount of gas remaining after deducting the basic amount, $T_g - g_0$	6.2 , 7, 8, 9.3, 9.4.0
g'	remaining gas	6.2, 7 , 8, 9.3
g^*	amount of gas to be refunded	6.2
g^{**}	available gas remaining	7
g_0	intrinsic gas, the amount of gas this transaction requires to be paid prior to execution	6.2
$H()$	function which specifies the output data of the instruction if and only if the present state is a normal halting state of the machine (cf. $Z()$)	9.4.0, 9.4.1, 9.4.4
H	block header	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4), 11.5.0
H_b	logsBloom, Bloom filter for each log entry from the each transaction receipt	4.3.0(4.4.0), 4.3.2(4.4.2), 4.3.3(4.4.3)
H_c	beneficiary, address to which all fees collected be transferred	4.3.0(4.4.0), 4.3.3(4.4.3), 11.3
H_d	difficulty, difficulty level of this block	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4), 11.4, 11.5.0
H_e	receiptRoot, root node hash of each transaction receipt	4.3.0(4.4.0), 4.3.1(4.4.1), 4.3.2(4.4.2), 4.3.3(4.4.3)

H_g	gasUsed, total gas used in transactions in this block	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4)
H_i	number, number of ancestor blocks, block number	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4), (6.2), (7), 11.3
H'_i	fake block number to delay the ice age	(4.3.4)
H_l	gasLimit, current limit of gas expenditure per block	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4)
H_m	mixHash, hash which proves a sufficient amount of computation	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4), 11.5.0
H_n	nonce, value which proves a sufficient amount of computation	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4), 11.5.0
$H_{\mathfrak{H}}$	block's header without the nonce (H_n) and mix-hash (H_m) components	4.3.4(4.4.4), 11.5.0
H_o	ommersHash, hash of the ommers (uncles) list	4.3.0(4.4.0), 4.3.2(4.4.2), 4.3.3(4.4.3)
H_p	parentHash, hash of the parent block's header	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4)
H_r	stateRoot, root node hash of the state trie	4.3.0(4.4.0), 4.3.2(4.4.2), 4.3.3(4.4.3), 11.4
H_s	timestamp, block creation time in seconds	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4)
H_t	transactionsRoot, root node hash of each transaction	4.3.0(4.4.0), 4.3.2(4.4.2), 4.3.3(4.4.3)
H_x	extraData, arbitrary data relevant to this block	4.3.0(4.4.0), 4.3.3(4.4.3), 4.3.4(4.4.4)
H_{RETURN}	special function for RETURN and REVERT	9.4.4
I	parameters of the execution environment	7, 8, 9.3, 9.4.0, 9.4.2, 9.5
I_a	address of the account which owns the code that is executing (a)	7, 8, 9.3
I_b	byte array that is the machine code to be executed (i)	7, 8, 9.3, 9.4.1, 9.4.2
I_d	!!! empty tuple as there is no input data to this call ($()$), byte array that is the input data to this execution; if the execution agent is a transaction, this would be the transaction data	7, 8, 9.3
I_e	depth of the present message-call or contract-creation (e)	7, 8, 9.3
I_H	block header of the present block	7, 9.3

I_o	sender address of the transaction that originated this execution (o)	7, 8, 9.3
I_p	price of gas in the transaction that originated this execution (p)	7, 8, 9.3
I_s	address of the account which caused the code to be executing; if the execution agent is a transaction, this would be transaction sender (s)	7, 8, 9.3
I_v	value in Wei passed to this account as part of the same procedure as execution (v)	7, 8, 9.3
I_w	permission to make modifications to the state (w)	(7, 8, 9.3, 9.4.2)
I'	dropped fourth element from $X()$	9.4.0()
\mathbf{i}	initialisation EVM code for creating a contract	7
i	!!! (μ_i)	9.4.1
$J, J_{\text{JUMP}}, J_{\text{JUMPI}}$!!!	9.5
$J_{\text{cacheinit}}, J_{\text{datasetinit}}, J_{\text{epoch}}$!!!	11.5.1
JUMP	!!!	9.4.2
JUMPDEST	!!!	9.4.3
JUMPI	!!!	9.4.2
$k()$	check “is-kin” property	11.1
KEC()	Keccak-256 hash function, plain Keccak	3, 4.1, 4.3.1(4.4.1), 4.3.2(4.4.2), 4.3.4(4.4.4), 7, 8
KEC512()	Keccak 512 hash function	3
$L_B()$	preparation function for a block	4.3.3(4.4.3)
$L_H()$	preparation function for a block header	4.3.3(4.4.3)
L_H^*	!!!	4.3.2(4.4.2), 4.3.3(4.4.3)
$L_I()$	collapse function for a key/value pair in the trie, $L_I((k, v)) \equiv (\text{KEC}(k), \text{RLP}(v))$	4.1
L_I^*	element-wise transformation of L_I (see $\sigma[a]_s$)	4.1
$L_R()$	function that converts a transaction receipt into an RLP-serialised byte array	4.3.1(4.4.1), 4.3.2(4.4.2)
$L_S()$	world-state collapse function	4.1, (4.4.1), 4.3.2(4.4.2), 11.4
$L_T()$!!!	4.2(4.3), 4.3.2(4.4.2)
L_T^*	!!!	4.3.3(4.4.3)
\mathbf{l}	log series (A_l)	9.3
$\ell()$	last item in the sequence	3, 11.2, 11.4
LOG0	!!!	9.4.2()
LOG4	!!!	9.4.2()
$M()$	Bloom filter function to reduce a log entry into a single 256-byte hash	4.3.1(4.4.1)
$M_{3:2048}()$	specialised Bloom filter that sets three bits out of 2048	4.3.1(4.4.1)
\mathbf{m}	!!! (μ_m)	9.4.1
m	beneficiary (miner, H_c)	6.2
$m()$	fetch the given index after hashing	4.3.1(4.4.1)
$N()$	next valid instruction position in the code	9.4.3, 9.5

$\mathbb{N}, \mathbb{N}_{number}$	positive integer, smaller than 2^{number}	3(), 4.1() 4.2() , 4.3.1(), 4.3.3()
N_H	block number between the Frontier and Homestead phases (1, 150,000)	(4.2), 4.3.4(4.4.4), (6.2), (7)
n	transaction nonce (see T_n)	3
$O()$	iterator function which defines the result of a single cycle of the state machine	9.4.0, 9.4.1, 9.5
O, O_{number}	log entry created through execution of the transaction (see R_1)	4.3.1(4.4.1)
O_a	logger's address (\mathbb{B}_{20})	4.3.1(4.4.1)
O_d	number of bytes of data (\mathbb{B})	4.3.1(4.4.1)
O_t	series of log topics	4.3.1(4.4.1)
$O_{t_{number}}$	log topic (\mathbb{B}_{32})	4.3.1(4.4.1)
o	output data of a message call or created contract's code (the resultant byte sequence from the execution of the initialisation code)	3, 7(), 8, 9.3
o	(contract creation or transaction) originator (see also T_o)	7, 8
$\mathbb{P}, \mathbb{P}_{number}$	positive integer, smaller than 2^{number}	(3, 4.1, 4.3 , 4.4.1, 4.4.3)
$P(B_H), P(H)$	parent block of B (see H_p)	4.3.2(4.4.2), 4.3.4 (4.4.4), 10, 11.1, 11.4
p	(contract creation or message call) gas price	7, 8
$p()$	per-account collapse function (see $L_S()$)	4.1
$p()$	pairwise RLP transformation	4.3.2(4.4.2)
pc	!!! (μ_{pc})	9.4.1
PoW	proof-of-work function	4.3.4(4.4.4), 11.5.0
PUSH1	!!!	9.4.3
PUSH32	!!!	9.4.3
R	!!!	11.2
R	transaction receipt	4.3.1(4.4.1)
R	!!!	11.3()
R_σ	post-transaction state	(4.4.1, 11.4)
$R[n]_\sigma$!!! ($\sigma[n]$)	(11.4)
R_b	Bloom filter composed from information in those logs (\mathbb{B}_{256})	4.3.1(4.4.1), 11.4
R_b	!!!	(11.3)
R_{block}	block's beneficiary account	11.3()
R_1	a series of log entries (O) created through execution of the transaction	4.3.1(4.4.1), 11.4
R_u	cumulative gas used immediately after the transaction has happened (\mathbb{N})	4.3.1(4.4.1), 11.2, 11.4
R_z	status code of the transaction (\mathbb{N})	4.3.1(), 11.4()
r	message call recipient	8
r	refunds (A_r)	9.3
$r()$!!!	11.4
r_b	Bloom filter of a receipt (see O)	4.3.2(4.4.2)
RETURN	!!!	9.4.4
RETURNDATACOPY!!!		9.4.2()

REVERT	!!!	9.4.0(), 9.4.4()
RLP()	!!! RLP transformation	4.1, 4.3.2(4.4.2), 4.3.3(4.4.3), 4.3.4(4.4.4), 7
$S(T)$	transaction sender	4.2(4.3), 6.2
\mathbf{s}	suicides set ($A_{\mathbf{s}}$)	9.3
\mathbf{s}	!!! ($\mu_{\mathbf{s}}$)	9.4.1
s	(contract creation or message call) sender	7, 8
$s()$	check “is-sibling” property	11.1
SELFDESTRUCT	!!!	6.1(), 9.4.2(), 9.4.4()
SSTORE	!!!	6.1, 9.4.2()
STOP	!!!	6.1, 9.4.1, 9.4.4
SUICIDE	!!!	(6.1, 9.4.4)
\mathbf{T}	transactions in the given block	4.3.0(4.4.0)
T, T_{number}	a transaction	2.0, 4.2(4.3), 6.0, 6.2
$T_{\mathbf{d}}$	data, input data of the message call (\mathbb{B})	4.2(4.3), 6.2
T_g	gasLimit, maximum amount of gas used in this transaction (\mathbb{N}_{256})	4.2(4.3), 6.2
$T_{\mathbf{i}}$	init, EVM-code for the account initialisation procedure, executed only once at account creation (\mathbb{B})	4.2(4.3), 6.2
$T_{\mathbf{n}}$	transaction nonce, number of transactions sent by the sender (\mathbb{N}_{256})	3, 4.2(4.3)
$T_{\mathbf{o}}$	original transactor, which can differ from the sender ($S(T)$)	6.2
$T_{\mathbf{p}}$	gasPrice, number of Wei to be paid per unit of gas (\mathbb{N}_{256})	4.2(4.3), 6.2
$T_{\mathbf{r}}$	one of transaction signature values (\mathbb{N}_{256})	4.2(4.3)
$T_{\mathbf{s}}$	one of transaction signature values (\mathbb{N}_{256})	4.2(4.3)
$T_{\mathbf{t}}$	to, message call’ recipient address (\mathbb{B}_{20}) or (contract creation transaction) \emptyset (\mathbb{B}_0)	4.2(4.3)
$T_{\mathbf{v}}$	value, number of Wei to be transferred (\mathbb{N}_{256})	4.2(4.3), 6.2
$T_{\mathbf{w}}$	one of transaction signature values (\mathbb{N}_5)	4.2(4.3)
\mathbf{t}	touched accounts ($A_{\mathbf{t}}$)	8(), 9.3()
TRIE()	!!!	4.1, (4.4.1), 4.3.2(4.4.2), 7, 8, 11.4
\mathbf{U}	set of uncle block headers	4.3.0(4.4.0)
$V()$	block header validity function	4.3.4(4.4.4), 11.1
$v()$	account validity function	4.1
v	(contract creation) endowment or (message call) value	7, 8
\tilde{v}	value to be transferred for the DELEGATECALL instruction	8
v'	account’s pre-existing value	7
v_0	up-front cost	6.2
W	!!!	9.4.2()
w	!!!	7(), 8(), 9.4.0(), 9.4.1, 9.4.2, 9.5
$X()$!!!	9.4.0

$X_{0,1,2,4}()$!!!	(9.4.0)
x	difficulty adjustment factor	4.3.4 (4.4.4), 11.4
$Z()$	function which check an exceptional halting state of the machine (cf. $H()$)	9.4.0, 9.4.1, 9.4.2
z	status code !!!	6.2(), 7(), 8()