An attempt to give the most abstract possible definition of a blockchain.

1 BlockChain

Chains as lists and their validation

Given a set S, let $\text{LIST}(S)$ and $\text{SET}(S)$ be the sets of all finite lists and all finite sets of elements of S, respectively. Given $L \in \text{LIST}(S)$, we use notation $|L|$ to refer to the number of elements in L, and notation $L[i]$ to refer to the i-th element in L, where $i \in \{1, \ldots, |L|\}$. From now on, assume that Σ is a finite alphabet, and that $\mathbf{B} \subseteq \Sigma^*$ is the set of all possible blocks. Moreover we extend the definition of $\subseteq$ such that :

$$\forall S \in \text{SET}(B), \forall L \in \text{LIST}(B), L \subseteq S \Leftrightarrow \forall i \in \{1, \ldots, |L|\}, L[i] \in S$$

Definition 1. *A validation rule is a function* $V : \text{LIST}(\mathbf{B}) \to \text{SET}(\mathbf{B})$

Intuitively V is a function taking a list L of block as input, and returning the set of blocks that could be added to L to produce a valid blockchain.

Definition 2. *Let* $G \in \text{LIST}(\mathbf{B})$ *be non-empty, and* V *be a validation rule. Then a list* $L \in \text{LIST}(\mathbf{B})$ *is a validated chain with respect to* (G, V) *if:*

1. $|G| \leq |L|$ *and* $L[i] = G[i]$*, for every* $i \in \{1, \ldots, |G|\}$.
2. $L[1] \in V([\,])$ *and* $L[i+1] \in V([L[1], \ldots, L[i]])$*, for every* $i \in \{1, \ldots, |L| - 1\}$.

List L in this definition is a valid chain according to the validation rule V and the lists G of genesis blocks (whose role is to provide the blocks to startup the system). Let $\text{LOG}(G, V)$ be the set of validated chains with respect to (G, V).

Definition 3. *Let* $G \in \text{LIST}(\mathbf{B})$ *be non-empty, and* V *be a validation rule. Then* $\text{LOG}(G, V)$ *is safe if for every* $L \in \text{LOG}(G, V)$ *such that every* $b_1, b_2 \in \mathbf{B}$ *such that* $b_1 \neq b_2$*:*

$$V([L[1], \ldots, L[|L|], b_1]) \cap V([L[1], \ldots, L[|L|], b_2]) \quad = \quad \emptyset$$

Intuitively, in order to be secured V should depend on the last block b that is included in the blockchain.

Knowledge

Definition 4. *A* knowledge tree K *is a tree* $K = (N, E)$ *with* $N \subseteq \mathbf{B}$ *and such that every path in* K *from its root to a leaf belongs to* $LOG_{G,V}$.

Intuitively, the knowledge tree represents all the blockchain information we know. Abusing notation, we say that a block B is in a knowledge tree $K = (N, E)$ if $B \in N$. (this is informal) We use PATHS(K) to denote the set of all lists of blocks made out of a path in K from its root to a leaf.

Block chain, protocols

Definition 5. *Let* $\preceq_{G,V,t}$ *be a total preorder over* $LOG_{G,V}$*:*

$$\forall L_1, L_2, L_3 \in LOG_{G,V}, L_1 \preceq_{G,V,t} L_2 \wedge L_2 \preceq_{G,V,t} L_3 \implies L_1 \preceq_{G,V,t} L_3$$
$$\forall L_1, L_2 \in LOG_{G,V}, L_1 \preceq_{G,V,t} L_2 \vee L_2 \preceq_{G,V,t} L_1$$

A block chain protocol over $LOG_{G,V}$ *is a function noted* $\preceq_{G,V}$ *such that:*

$$\forall t \in \mathbb{N}, \preceq_{G,V}(t) = \preceq_{G,V,t}$$

where $\preceq_{G,V,t}$ *is a total preorder over* $LOG_{G,V}$

Definition 6. *Let* $t \in \mathbb{N}$*,* $\preceq_{G,V}$ *a block chain protocol and* K *a knowledge tree. A block chain of* K *with respect to* $\preceq_{G,V}$ *in* t *is any minimal element in PATHS*(K) *with respect to* $\preceq_{G,V}(t)$.