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 $\textsc{List}(S)$ and $\textsc{Set}(S)$ be the sets of all finite lists and all finite sets of elements of S, respectively. Given $L \in \textsc{List}(S)$, 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 :

$$\forall S \in \textsc{Set}(B), \forall L \in \textsc{List}(B), L \subseteq S \Leftrightarrow \forall i \in \{1, \ldots, |L|\}, L[i] \in S$$

Definition 1. *A validation rule is a function* $V : \textsc{List}(\mathbf{B}) \to \textsc{Set}(\mathbf{B})$

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 $G \in \textsc{List}(\mathbf{B})$ be non-empty, and V be a validation rule. Then a list $L \in \textsc{List}(\mathbf{B})$ 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 $\textsc{Log}(G, V)$ be the set of validated chains with respect to (G, V).

Definition 3. *Let $G \in \textsc{List}(\mathbf{B})$ be non-empty, and V be a validation rule. Then $\textsc{Log}(G, V)$ is safe if for every $L \in \textsc{Log}(G, V)$ 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 $LOG_{G,V}$. Let $\mathcal{K}$ be the set of knowledge tree with respect to (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 $\preceq_{G,V}(t)$.*

Definition 7. *Let $K = (N, E)$ and $K' = (N', E')$ two knowledge trees we say that K and K' are $\equiv$ equivalent if :*

$$\equiv \text{ is an equivalent function}$$
$$\forall L \in \mathit{PATHS}(K), \exists L' \in \mathit{PATHS}(K'), \forall i \in [\![1, |L|]\!], L[i] \equiv L'[i]$$
$$\forall L' \in \mathit{PATHS}(K'), \exists L \in \mathit{PATHS}(K), \forall i \in [\![1, |L'|]\!], L[i] \equiv L'[i]$$

We denote $K^{\equiv}$ the set of knowledge equivalent to K.

Action, states, reward and game

Definition 8. *Considering a set of player P we denote $\mathcal{K}_P$ the set of function $K_P : P \to \mathcal{K}$ mapping a knowledge tree to each player. We denote $\mathcal{K}_P^{\equiv}$ the set of mapping where:*

$$\forall K_P, K'_P \in \mathcal{K}_P^{\equiv}, \forall p \in P, K_P(p) \text{ and } K'_P(p) \text{ are } \equiv \text{ equivalent}$$

Intuitively $\mathcal{K}_P$ represents the true knowledge of each player.

Definition 9. *We call action a for player $w \in P$ function $a_w^{\equiv} : \mathcal{K}_P \to \mathcal{K}_P^{\equiv}$ such that:*

$$\forall K_P \in \mathcal{K}_P, \forall K'_P \in a_w(K_P), \forall u \in P, K_P(u) \subseteq K'_P(u)$$
$$\forall K_P \in \mathcal{K}_P, \forall K'_P \in a_w(K_P), \forall u \in P, K'_P(u) \subseteq K'_P(w) \cup K_P(u)$$

Let $A_w^{\equiv}$ be the set of action for player w.

An action of player w is represented by a modification of w knowledge and a round of communication. We are dealing with equivalence knowledge in order to reduce the number of action possible.

Definition 10. *Let P be a set of player, $\mathcal{A}^{\equiv} = A_{p_1}^{\equiv} \times A_{p_1}^{\equiv} \ldots \times A_{p_{|P|}}^{\equiv}$*

Definition 11. *We call reward for player w a function $r_w : \mathcal{K}_P \times \mathcal{A}^{\equiv} \to \mathbb{R}^+$ such that:*

$$\forall A \in \mathcal{A}^{\equiv}, K_P \text{ equivalent } K'_P \Rightarrow r_w(K_P, A) = r_w(K'_P, A)$$

Intuitively $\mathcal{K}_P$ represent the knowledge of each player assumed by w

Definition 12. *Let P be a set of player, $\mathcal{R} = r_{p_1} \times r_{p_1} \ldots \times r_{p_{|P|}}$*

Definition 13. *Considering a set of player P, a set of function $\mathcal{K}_P$, the set of action $\mathcal{A}$, the set of reward $\mathcal{R}$, and a function $\mathcal{P} : \mathcal{K}_P \times \mathcal{A} \times \mathcal{K}_P^{\equiv} \to [0; 1]$ a transition probability ($\mathcal{P}(K_P, A, K_P^{1\equiv})$ is the probability of transitioning from K_P to an element of $K_P^{1\equiv}$ after joint action A). We define a infinite stochastic game such that:*

- *P is the set of player.*
- *$\mathcal{A}$ is the set of available action.*
- *$\mathcal{K}_P$ is the set of states.*
- *$\mathcal{R}$ the set of pay-off function.*
- *$\mathcal{P}$ is the transition probability function.*

Remark. *I don't like what i am doing with equivalence as it complicate stuff a lot. Moreover if i have to prove that my game is well defined as it is not immediate due to that. Finally we have an infinite number of states, equivalence states and finitely repeated game might be a solution.*

Definition 14. *We say that K'_P is reasonable regarding I and K_P if exists an action $a \in A$ associate to a player $w \in P$ in an equilibrium profile such that:*

$$K'_P(w) = a(K_P(w))$$

K'_P represent all the possible knowledge after one reasonable action (on belonging to a nash equilibrium) has happened with an optional comunication round from the winner ($K'_P(p) \subseteq K_P(p) \cup K'_P(w)$). (We may want to include comunication is A instead of here as it's part of the strategie ... so will influence I).

Definition 15. *We say that K^n_p is n reasonable regarding K_p and I if exists $(K^i_p), \forall i \leq n, K^i_p$ reasonable regarding K^{i-1}_p and I and $K^0_P = K_P$.*

Good to go we finally have a defintion of reasonable K and can define blockchain property which should be verified over all reasonable K.