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

# 1 BlockChain

## 1.1 Lists and their validation

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

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

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

A list $L \in \text{FLIST}(\mathbf{B})$ is simple if for every $i, j \in \{1, \ldots, |L|\}$ such that $i \neq j$, it holds that $L[i] \neq L[j]$. Moreover, $L$ is said to be a genesis list of $V$ if $L$ is non-empty, simple, $L[1] \in V([\,])$ and $L[i+1] \in V(L[1, i])$, for every $i \in \{1, \ldots, |L| - 1\}$.

**Definition 2.** *Let $V$ be a validation rule and $G$ be a genesis list of $V$. Then a list $L \in \text{FLIST}(\mathbf{B})$ is valid with respect to $(G, V)$ if:*

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

The role of $G$ in this definition is to provide the blocks to startup the system. Let $\text{LOG}(G, V)$ be the set of valid lists with respect to $(G, V)$.

**Definition 3.** *Let $V$ be a validation rule and $G$ be a genesis list of $V$. Then $\text{LOG}(G, V)$ is safe if for every $L \in \text{LOG}(G, V)$, with $|L| = k$, and every $b_1, b_2 \in \mathbf{B}$ such that $b_1 \neq b_2$, it holds that*:

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

## 1.2 Knowledge

Given a a validation rule $V$, a genesis list $G$ with respect to $V$ and a subset $K$ of $\text{LOG}(G, V)$, there is a natural way to visualize $K$ as a graph $\mathcal{G}(K)$. The set of nodes of $\mathcal{G}(K)$ is the set of blocks occurring in the lists in $K$, and there is an edge from a block $b_1$ to a block $b_2$ if there exists a list $L \in K$ such that $b_1 = L[i]$ and $b_2 = L[i+1]$, where $i \in \{1, \ldots, |L| - 1\}$.

**Lemma 1.** *Assume that $\text{LOG}(G, V)$ is safe. The for every subset $K$ of $\text{LOG}(G, V)$, it holds that $\mathcal{G}(K)$ is a tree rooted at $G[1]$.*

Thus, assuming that $\text{LOG}(G, V)$ is safe, from now on we refer to every non-empty finite subset $K$ of $\text{LOG}(G, V)$ as a *knowledge tree* of $(G, V)$. Moreover, we define $\mathcal{K}(G, V)$ as the set of all knowledge trees of $(G, V)$.

## 1.3 Block chain and protocols

**Definition 4.** *A relation $\preceq$ on $\text{LOG}(G, V)$ is said to be a knowledge order over $(G, V)$ if the following conditions are satisfied:*

1. *$\preceq$ is a total preorder, that is, $\preceq$ is reflexive, transitive and total.*

2. *For every $K_1, K_2 \in \mathcal{K}(G,V)$ such that $K_1 \subsetneq K_2$, it holds that $K_1 \preceq K_2$ and $K_2 \not\preceq K_1$.*

*Moreover, a blockchain protocol over $(G,V)$ is a sequence $\{\preceq_t\}_{t\in\mathbb{N}}$ such that each $\preceq_i$ $(i \in \mathbb{N})$ is a knowledge order over $(G,V)$.*

**Definition 5.** *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 6.** *Let $K = (N,E)$ and $K' = (N',E')$ two knowledge trees we say that $K$ and $K'$ are $\equiv$ equivalent if :*

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

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

**Action, states, reward and game**

**Definition 7.** *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 8.** *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 9.** *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 10.** *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 11.** *Let $P$ be a set of player, $\mathcal{R} = r_{p_1} \times r_{p_1} \ldots \times r_{p_{|P|}}$*

# 2 Etienne 's modification space

**Definition 12.** *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 $\Gamma$ 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.*

**Stationary Nash equilibrium, Reasonable knowledge**

We know that stochastic games with finite number of states have a discounted nash equilibrium and it is computable : (Nonlinear program with linear constraints). 2 solutions: either we are able to define equivalence classe over states to create a finite number state stochastic game or our game fit in the "On a new class of nonzero-sum discounted stochastic games having stationary Nash equilibrium points", have to understand this paper ...

**Definition 13.** *We say that* $K'_P$ *is reasonable regarding a game* $\Gamma$ *if exists a strategy*