計算用紙

まい月

はじめに

　本書は著者が試してみたいと思うことをリストするという体でLaTeXとTikZを練習する文書である。

目次

第1章
ステーブルコインのようなもの

1.1 状況

ブロックチェーンで流通するトークン🍑を裏付けとして、ある価値🍎と連動したトークン🍃を発行したい。

1.2 ためしてみたい解決策

参加者の財産	DAOの財産	レバレッジ	🍃/🍑
• 現物1000🍑	0	0.5倍	1🍑
参加者は1000🍑を担保に10🍃鋳造する			
• 1000🍑を担保に10🍃鋳造したポジション • 現物10🍃	• 現物1000🍑	0.5倍	1🍑

Table 1.1: 鋳造する手続き

参加者の財産	DAOの財産	レバレッジ	🍃/🍑
• 1000🍑を担保に10🍃鋳造したポジション • 現物10🍃	• 現物1000🍑	0.5倍	1🍑
参加者は10🍃焚いて1000🍑を返してもらう			
• 現物1000🍑	0	0.5倍	1🍑

Table 1.2: 焚く手続き

参加者 の財産	DAO の財産	レバ レッジ	🍃/🍑
• 1000🍑を担保に 10🍃鋳造した ポジション • 現物10🍃	• 現物1000🍑	0.5倍	1🍑
1🍃の価格が100🍑になり、 1000🍑にレバレッジの0.5倍を乗じた額を超えて 🍃を鋳造している状態になり強制ロスカする			
• 現物10🍃	• 現物1000🍑	0.5倍	100🍑

Table 1.3: 強制決済

第2章
インポータンスの計算の効率化

2.1 目標と方針

この節は理想についての演説にほかならず中身がない。

まずSymbolやNEMではブロックを検証するのに所定の条件を満たした適格なアカウントであることを要求していた。これは検証するための抽選とは別に、その抽選に参加するために必要なものであった。

そこでSymbolのPoS+を参考にしつつSymbolのPoS+より計算の効率と開かれ具合をすっごく上げたい。特にSymbolで採用されているPoS+と今回やりたいことの違いは：

- 所持金の下限を撤廃
  - ブロックチェーンに載っていれば適格
- ブロック毎に計算できる効率
- 毎回すべての適格なアカウントのインポータンスを計算しなくていい

であり、そのための原材料は：

- 残高
- 検証したブロックの数
- 支払った手数料

であり、ここまではSymbolと同じで、ここからが肝心なところで、これらすべての適格なアカウントについての総和がブロックヘッダに残ること、そしてこれがいたって自然な活動であれば尚良い。

2.2 準備

ブロックiごと	アカウント$j \in \mathcal{A}$ごと	パラメータ
• アカウントの数$\#\mathcal{A}$ • 総供給量$S_1(i)$ • ブロックの高さ$S_2(i)$ • 手数料の合計$S_3(i)$	• 残高$s_1(i,j)$ • 検証した回数$s_2(i,j)$ • 手数料の合計$s_3(i,j)$	• 遡るブロックH • ハッシュの大きさN • 理想の間隔T • 重み$\alpha_1, \alpha_2, \alpha_3$

Table 2.1: オンチェーンの情報

Table2.1のとおりブロックチェーンに書き込むものとして、色々計算する。

2.3 計算

そしてみんなここにしか興味なしであろう計算の部分です。

$$\delta(u,v) \coloneqq \begin{cases} 1 & (u=v) \\ 0 & \text{otherwise} \end{cases}$$

$$\tilde{I}(i,j) \coloneqq \frac{\sum_{h=0}^{\min\{H,i\}-1} \frac{\sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\sum_{k=1}^{3} \alpha_k}}{\min\{H,i\}}$$

$$= \frac{1}{\min\{H,i\}} \sum_{h=0}^{\min\{H,i\}-1} \frac{1}{\sum_{k=1}^{3} \alpha_k} \sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}$$

$$M \coloneqq 3NH\#\mathcal{A}+1$$

$$I(i,j) \coloneqq \frac{1}{M} \left\lfloor M\tilde{I}(i,j) \right\rfloor$$

この$I(i,j)$、あるいはお好みの方法で減衰させたもの（e.g., $\tilde{I}(i,j)$の代わりに$\frac{\sum_{h=0}^{\min\{H,i\}-1} \gamma^h \frac{\sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\sum_{k=1}^{3} \alpha_k}}{\sum_{h=0}^{\min\{H,i\}-1} \gamma^h}$（$\gamma \in (0,1)$）を使う、すると適当な$M$も変わりうる）をインポータンス（i.e., 議決権）として使おうと企てている。

2.4 導出過程

この節では、みんな興味ないかもしれないけど、$M \coloneqq NH\#\mathcal{A}+1$とする理由を説明する。$\forall i=1,2,\ldots; \left|1-\sum_{j\in\mathcal{A}} I(i,j)\right| < \frac{1}{N}$を満たせばよいから、

$$\forall i=1,2,\ldots;$$
$$\left|1-\sum_{j\in\mathcal{A}} I(i,j)\right| < \frac{1}{N}$$

$$\Leftrightarrow \left| \sum_{j\in\mathcal{A}} \left(\frac{\sum_{h=0}^{\min\{H,i\}-1} \frac{\sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\sum_{k=1}^{3}\alpha_k}}{\min\{H,i\}} - \frac{1}{M} \left\lfloor M \frac{\sum_{h=0}^{\min\{H,i\}-1} \frac{\sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\sum_{k=1}^{3}\alpha_k}}{\min\{H,i\}} \right\rfloor \right) \right| < \frac{1}{N}$$

$$\Leftrightarrow \sum_{j\in\mathcal{A}} \left(\frac{\sum_{h=0}^{\min\{H,i\}-1} \frac{\sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\sum_{k=1}^{3}\alpha_k}}{\min\{H,i\}} - \frac{1}{M} \left\lfloor M \frac{\sum_{h=0}^{\min\{H,i\}-1} \frac{\sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\sum_{k=1}^{3}\alpha_k}}{\min\{H,i\}} \right\rfloor \right) < \frac{1}{N}$$

$$\Leftarrow \sum_{j\in\mathcal{A}} \sum_{h=0}^{\min\{H,i\}-1} \left(\frac{\sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\min\{H,i\}\sum_{k=1}^{3}\alpha_k} - \frac{1}{M} \left\lfloor M \frac{\sum_{k=1}^{3} \alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\min\{H,i\}\sum_{k=1}^{3}\alpha_k} \right\rfloor \right) < \frac{1}{N}$$

$$\Leftarrow \sum_{j\in\mathcal{A}} \sum_{h=0}^{\min\{H,i\}-1} \sum_{k=1}^{3} \left(\frac{\alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\min\{H,i\}\sum_{k=1}^{3}\alpha_k} - \frac{1}{M} \left\lfloor M \frac{\alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\min\{H,i\}\sum_{k=1}^{3}\alpha_k} \right\rfloor \right) < \frac{1}{N}$$

$$\Leftrightarrow \sum_{j\in\mathcal{A}} \sum_{h=0}^{\min\{H,i\}-1} \sum_{k=1}^{3} \left(\frac{M\alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\min\{H,i\}\sum_{k=1}^{3}\alpha_k} - \left\lfloor M \frac{\alpha_k \frac{s_k(h,j)+\delta(1,h)}{S_k(h)+\delta(1,h)}}{\min\{H,i\}\sum_{k=1}^{3}\alpha_k} \right\rfloor \right) < \frac{M}{N}$$

$$\Leftarrow \sum_{j\in\mathcal{A}} \sum_{h=0}^{\min\{H,i\}-1} \sum_{k=1}^{3} 1 < \frac{M}{N}$$

$$\Leftrightarrow 3N\#\mathcal{A}\min\{H,i\} < M$$

$$\Leftarrow 3NH\#\mathcal{A} < M$$

となる。例えば、ハッシュにはKeccak512を用い、アカウントは1000こあり、ブロックは100000ブロック前まで遡って良いものとして、$M = 3 \cdot 2^{512} \cdot 100000 \cdot 1000 + 1$となる。

第3章
難易度とヴァリデーターの抽選

3.1 目標

みんなにブロックチェーンの維持をお願いするために、やらなきゃいけないこと、それが「だれにしようかな」である。誰たちのなかからどうやってどんなひとを選ぶか、どのくらいのペースで条件がゆるくなるか、それを決めなければならない。

3.2 具体的な計算

現在の時刻をt、出力の大きさNのハッシュを$h:\{0,1\}^{*}\to\{1,2,\dots,N\}$、$i$番目のブロック$b_i$の難易度$d(i)$を、ブロック生成の時刻を$t(i)$、難易度を導出するために遡るブロックの高さを$D$、目標のブロック生成の間隔を$T$、アカウント$j\in\mathcal{A}$の議決権を全体の$I(i,j)$倍とすると、

$$d(i+1):=\frac{T}{t(i)-t(i-D)}\sum_{g=1}^{\min\{i,D\}}d(i)$$

と更新すると合理的な感じがして、$p(s):=\frac{h(s)}{N}$とすると、このpとある定数$C>1$を用いて、それっぽい式$p\left(b_{i-1}\parallel j\right)<C^{-\frac{t-t(i-1)}{d(i)}I(i,j)}$を変形して、

$$\begin{aligned}
&p\left(b_{i-1}\parallel j\right)>C^{-\frac{t-t(i-1)}{d(i)}I(i,j)}\\
\Leftrightarrow&\log_C p\left(b_{i-1}\parallel j\right)>-\frac{t-t(i-1)}{d(i)}I(i,j)\\
\Leftrightarrow&\log_C\frac{h\left(b_{i-1}\parallel j\right)}{N}>-\frac{t-t(i-1)}{d(i)}I(i,j)\\
\Leftrightarrow&-d(i)\log_C\frac{h\left(b_{i-1}\parallel j\right)}{N}<(t-t(i-1))I(i,j)\\
\Leftrightarrow&d(i)\left(\log_C N-\log_C h\left(b_{i-1}\parallel j\right)\right)<(t-t(i-1))I(i,j)
\end{aligned}$$

を得る。ここで$C=2$とhとしてKeccak512の値を整数と思って1を足したものを選ぶと$N=2^{512}$となり、$\exists M\in\mathbb{Z}\text{s.t.},\forall i,j;M\cdot I(i,j)\in\mathbb{Z}$なる$I$を用いてきりのいいところで端数処理を行うと、

$$\begin{aligned}
&d(i)\left(512-\log_2 h\left(b_{i-1}\parallel j\right)\right)<(t-t(i-1))I(i,j)\\
\Leftrightarrow&d(i)\left(512M-M\log_2 h\left(b_{i-1}\parallel j\right)\right)<M(t-t(i-1))I(i,j)\\
\Leftarrow&d(i)\left(512M-\left\lceil M\log_2 h\left(b_{i-1}\parallel j\right)\right\rceil\right)<M(t-t(i-1))I(i,j)\\
\Leftarrow&d(i)\left(512M-M\left\lceil\log_2 h\left(b_{i-1}\parallel j\right)\right\rceil\right)<M(t-t(i-1))I(i,j)
\end{aligned}$$

を得る。すなわち議決権が大きくハッシュの値の桁が大きいほど当たりやすい。やったね✨

第4章
データ構造

4.1 Merkle木

あとの方の端っこでいっしょにしてハッシュを計算する相手が足りなくなるが、その度に余分にハッシュを計算するのが気に食わないのでFigure4.1のようにする。

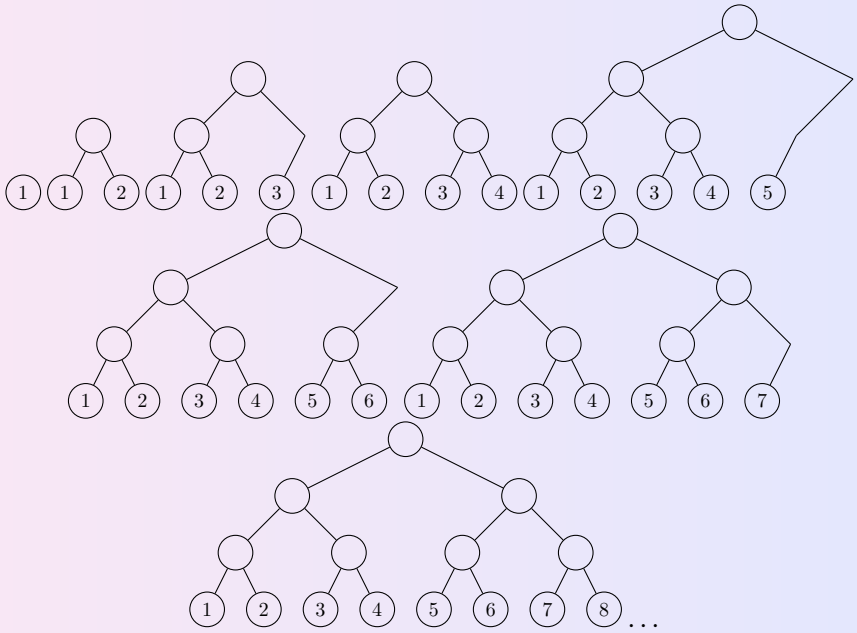

Figure 4.1: Merkle木