
Algorithm 1: Handling Inconsistent Blockchain Views (G)

Input: Received Blockchain graph G
Output: Authoritative view A_v
// Collect all valid sibling groups to S

```
1  $S \leftarrow \emptyset$ 
2 foreach  $B \in G$  do
3    $s \leftarrow \text{SIBLING}(B)$ 
4    $S.\text{APPEND}(s)$ 
5 end
  // Initialize weights of each sibling groups
6 foreach  $s \in S$  do
7    $W(s) \leftarrow 0$ 
8 end
  // Calculate weights of all sibling groups
9 foreach  $B \in G$  do
10  foreach  $s \in S$  do
11    foreach  $v \in \text{SUBVIEWSBEGINWITH}(s)$  do
12      if  $B \in v$  then
13         $W(s) \leftarrow W(s) + \text{MININGPOWER}(B)$ 
14        break
15      end
16    end
17  end
18 end
  // Find the authoritative view
19  $s \leftarrow \text{SIBLINGS}(B^0)$  //  $B^0$  stands for genesis block
20  $A_v \leftarrow \emptyset$ 
21  $A_v.\text{APPEND}(s)$ 
22 if  $\text{CHILDSIBLINGGROUPS}(s) = \emptyset$  then
23   return  $s$ 
24 else
25   Update  $s \leftarrow \arg \max_{s' \in \text{CHILDSIBLINGGROUPS}(s)} W(s')$ 
26   Go to line 21
27 end
```

Function: Subviews Begin with

```
// Function to find all subviews that begin with s
1 Function SUBVIEWSBEGINWITH( $s$ ):
2    $V \leftarrow \emptyset$ 
3   if  $CHILDSIBLINGGROUPS(s) = \emptyset$  then
4      $v \leftarrow \text{CONSTRUCTSUBVIEW}(s)$ 
5      $V \leftarrow V.\text{APPEND}(v)$ 
6     return  $V$ 
7   else
8     foreach  $c \in CHILDSIBLINGGROUPS(s)$  do
9       foreach  $v \in \text{SUBVIEWSBEGINWITH}(c)$  do
10         $V.\text{APPEND}(v.\text{APPEND}(s))$ 
11      end
12    end
13  end
14 return  $V$ 
```

Function: Child Sibling Groups

```
// Function to find child siblings
1 Function CHILDSIBLINGGROUPS( $s$ ):
2    $C \leftarrow \emptyset$ 
3   foreach  $s' \in S$  do
4     foreach  $B' \in s'$  do
5       foreach  $B \in s$  do
6         if  $B \in B'.\text{predecessors}$  then
7            $C.\text{APPEND}(s')$ 
8         end
9       end
10    end
11  end
12 return  $C$ 
```
