Algorithm 1: Handling Inconsistent Blockchain Views (G_0, B_0)

Input: Blockchain graph G_0, genesis block B_0
Output: Main View v_0

1   $S_0 \leftarrow \varnothing$
2   **foreach** $B \in G$ **do**
3     $s \leftarrow \text{BLOCKTOSUBGROUP}(B)$
4   **if** $B' \in S_0$ **then**
5   s
6   **else**
7   d
8   **end**
9   **end**
   // Find all subgroups
10 **foreach** $B \in G$ **do**
11   **foreach** $s \in S_0$ **do**
12      $W(s) \leftarrow 0$
13   **foreach** $v \in \text{SUBVIEWSBEGINWITH}(s)$ **do**
14   **if** $B \in v$ **then**
15          $W(s) \leftarrow W(s) + \text{MININGPOWER}(B)$
16   **continue**
17   **end**
18   **end**
19   **end**
20 **end**
21 $s \leftarrow \text{BLOCKTOSUBGROUP}(B_0)$
22 **if** $\text{CHILDSUBROUPS}(s) = \varnothing$ **then**
23   **return** s
24 **else**
25   Update $s \leftarrow \underset{s' \in \text{CHILDSUBROUPS}(s)}{\arg\max} W(s')$
26   Go to line 15
27 **end**
28 **Function** $\text{FINDSUBGROUPS}(G_0)$**:**
29    $v \leftarrow v \cup \mathcal{S}$
30    $U \leftarrow U \cup \{v\}$
31   **if** $\mathcal{S}$ *is genesis* **then**
32   **return** v, U
33   **else**
34      $\text{ADDPRE}(\mathcal{S}.predecessors, G, v, U)$
35   **end**