4 Algorithm

We present two slightly different versions of the algorithm. The first version is easier to understand but analyzing its complexity needs more argument. Our initial plan was to present the first version algorithm. The reason for the first version of the algorithm, is also to respond to anonymous reviewers.

The beginning lines take care of some easy special cases - performing Preprocessing to ensure lists are (2,3)-consistent (see paragraph 3) , and separating $G \times_L H$ into weakly connected components. After line 3 we can assume $G \times_L H$ is a single weakly connected component.

Algorithm 1 RemoveMinority – Using Maltsev Property

1: **function** RemoveMinority(G, H, L)
2: PreProcessing(G, L) and if a list becomes empty return $\emptyset$.
3: We assume $G \times_L H$ is connected else we consider each connected component separately
4: **for all** $x \in V(G)$ s.t $|L(x) = 1|$ **do** $\psi(x) = L(x)$
5: **for all** $x \in V(G), a, b \in L(x)$ **do**
▷ $(x, a), (x, b)$ are in the same component of $G \times_L H$
6: $(G^x_{a,b}, L') =$ Sym-Dif(G, L, x, a, b).
7: $g^x_{a,b} =$ RemoveMinority$(G^x_{a,b}, H, L')$ ▷ Performing a small test
▷ $g^x_{a,b}$ is a L' homomorphism from $G^x_{a,b}$ to H
8: **for all** $x \in V(G)$, $a, b \in L(x)$ **do**
9: **if** $g^x_{a,b}$ is empty **then** remove a from $L(x)$.
10: PreProcessing (G, H, L).
11: set ψ to be an empty homomorphism.
12: **while** $\exists x \in V(G), a \neq b \in L(x)$ **do**
13: Remove b from $L(x)$
14: Pre-processing the lists L
15: **if** $\exists x \in V(G)$ with $L(x)$ is empty **then** return $\emptyset$.
16: **else**
17: **for all** $x \in V(G)$ **do**
18: $\psi(x) = L(x)$ ▷ in this case the lists are singletons
19: **return** ψ.

20: **function** Sym-Dif(G, L, x, a, b)
21: Initiate empty lists L'.
22: $L'(x) = \{a\}$.
23: $L'(y) = \{d \ |(a, d) \in L(x, y), (b, d) \notin L(x, y)\}$.
24: Let G' be the induced sub-digraph of G with vertices y such that $L'(y) \neq \emptyset$
25: Let V_1 be the set of vertices of $G \setminus G'$ with out-neighbor (in-neighbor) to a vertex in G'.
▷ The vertices at the end of rectangles; boundary vertices.
26: Add the vertices of V_1 into G' together with their connecting arcs.
27: $\forall\ y \in V_1$, set $L'(y) = \{d \ |(a, d) \in L(x, y)\}$.
28: **return** (G', L')

An example suggested by anonymous reviewer: Consider the following system of linear equations in Z_2.

This system of equation has a solution in Z_2. For $y = z = 1$, the system does not have a solution because of the parity reason and when $y = z = 0$ the system has several solutions. Each equation is considered as a vertex of the graph G depicted in Figure 4. Two vertices are adjacent if they share a