decomposition of $C_G(N) = 1 \times \mathrm{PSL}(2,5) \times \mathrm{PSL}(2,5)$. At this point the algorithm returns the unique Remak decomposition

$$\mathcal{Q} := \{\mathrm{PSL}(2,5) \times 1 \times 1, 1 \times \mathrm{PSL}(2,5) \times 1, 1 \times 1 \times \mathrm{PSL}(2,5)\}.$$

These are pulled back to the set $\{H_1, H_2, H_3\}$ of subgroups in G.

Next the algorithm constructs a Remak G-decomposition of $\zeta_2(G)$. For that the algorithm constructs the bilinear map of commutation from $\zeta_2(G)/\zeta_1(G) \cong \mathbb{Z}_2^4$ into $\gamma_2(\zeta_2(G)) = \langle z_1, z_2\rangle \cong \mathbb{Z}_2^2$, i.e.

$$b := \mathsf{Bi}(\zeta_2(G)) : \mathbb{Z}_2^4 \times \mathbb{Z}_2^4 \to \mathbb{Z}_2^2$$

Below we have described the structure constants for b in a nice basis but remark that unless we already know the direct factors of $\zeta_2(G)$ it is unlikely to have such a natural form.

$$b(u,v) = u \begin{bmatrix} 0 & z_1 & & \\ -z_1 & 0 & & \\ & & 0 & z_2 \\ & & -z_2 & 0 \end{bmatrix} v^t, \qquad \forall u, v \in \mathbb{Z}_2^4. \tag{7.1}$$

A basis for the centroid of b is computed:

$$C(b) = \left\{ \left(\begin{bmatrix} a & 0 & & \\ 0 & a & & \\ & & b & 0 \\ & & 0 & b \end{bmatrix}, \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \right) : a, b \in \mathbb{Z}_2 \right\} \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2. \tag{7.2}$$

Next, the unique frame $\mathcal{E} = \{(I_2 \oplus 0_2, 1 \oplus 0), (0_2 \oplus I_2, 0 \oplus 1)\}$ of $C(b)$ is built and used to create the subgroups $\mathcal{K} := \{D_8 \times Z(Q_8), Z(D_8) \times Q_8\}$ in $\zeta_2(G)$. Here, using an arbitrary basis $\mathcal{X}$ for $\zeta_1(G) = \mathbb{Z}_2^2 \times \mathbb{Z}_4^2$, the algorithm `Merge`$(\mathcal{X}, \mathcal{K})$ constructs a Remak decomposition $\mathcal{A} := \{H, K, C_1, C_2\}$ of $\zeta_2(G)$ where $H \cong D_8$, $K \cong Q_8$, and $C_1 \cong C_2 \cong \mathbb{Z}_4$.

Finally, the algorithm `Merge`$(\mathcal{A}, \mathcal{H})$ returns a Remak decomposition of G. To explain the merging process we trace that algorithm through as well.

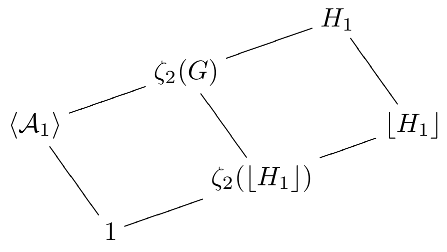
Let $R = \mathrm{SL}(d,q) \times 1 \times 1$ and $S = 1 \times (\mathrm{SL}(d,q) \circ \mathrm{SL}(d,q))$. These groups are directly indecomposable direct factors of G and serve as the hypothesized directions of for the direct chain used by `Merge`. Without loss of generality we index the H's so that $H_2 = R\zeta_2(G)$ and $H_1H_3 = S\zeta_2(G)$ and

$$G/\zeta_2(G) = \mathrm{PSL}(d,q) \times \mathrm{PSL}(d,q) \times \mathrm{PSL}(d,q) = H_2/\zeta_2(G) \times H_1/\zeta_2(G) \times H_3/\zeta_2(G).$$

Furthermore, $\zeta_2(H_i) = \zeta_2(G)$ for all $i \in \{1,2,3\}$. Therefore, $(\mathcal{A}, \mathcal{H})$ satisfies the hypothesis of Theorem 4.13.

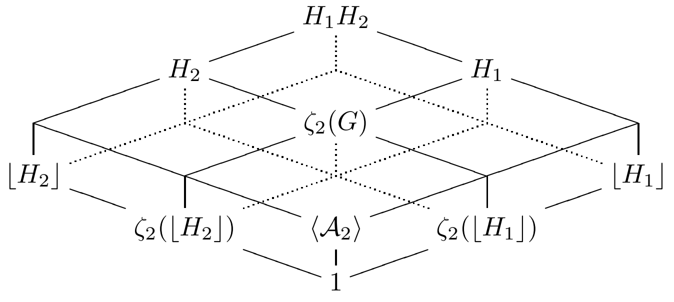
The loop in `Merge` begins with $\mathcal{K}_0 = \mathcal{A}$ and seeks to extend $\mathcal{A}$ to H_1 by selecting an appropriate subset $\mathcal{A}_1 \subseteq \mathcal{K}_0 = \mathcal{A}$ and finding a complement $\lfloor H_1 \rfloor \leq H_1$ such that $\mathcal{K}_1 = \mathcal{A}_1 \sqcup \{\lfloor H_1 \rfloor\}$ is a direct decomposition of H_1. The configuration at this stage is seen in Figure 4. By Theorem 4.10, we have $H, K \in \mathcal{A}_1$ (as those lie outside the center) and one of the C_i's (though no unique choice exists there).

In the second loop iteration we extend $\mathcal{K}_1$ to a $\mathfrak{N}_2$-refined direct decomposition if H_1H_2. This selects a subset $\mathcal{A}_2 \subseteq \mathcal{K}_1 \cap \zeta_2(G)$. Also H_1 and H_2 are in different directions, specifically $H_2 = R\zeta_2(G)$ and $H_1 \leq S\zeta_2(G)$, so the algorithm is forced to include $\lfloor H_1 \rfloor \in \mathcal{K}_2$ (cf. Theorem 4.10(iii)) and then creates a complement $\lfloor H_2 \rfloor \cong \mathrm{SL}(2,5)$ to $\langle \mathcal{A}_2, \lfloor H_1 \rfloor \rangle$. The configuration is illustrated in Figure 5. As



H_1
$\zeta_2(G)$
$\langle \mathcal{A}_1 \rangle$
$\lfloor H_1 \rfloor$
$\zeta_2(\lfloor H_1 \rfloor)$
1

FIGURE 4. The lattice encountered during the first iteration of the loop in the algorithm $\texttt{Merge}(\mathcal{A}, \{H_1, H_2, H_3\})$.



H_1H_2
H_2
H_1
$\zeta_2(G)$
$\lfloor H_2 \rfloor$
$\lfloor H_1 \rfloor$
$\zeta_2(\lfloor H_2 \rfloor)$
$\langle \mathcal{A}_2 \rangle$
$\zeta_2(\lfloor H_1 \rfloor)$
1

FIGURE 5. The lattice encountered during the second iteration of the loop in the algorithm $\texttt{Merge}(\mathcal{A}, \{H_1, H_2, H_3\})$.

before, we have $H, K \in \mathcal{K}_2$ as well, but the cyclic groups are now gone as the centers of $\lfloor H_i \rfloor$, $i \in \{1, 2\}$, fill out a direct decomposition of $\zeta_2(G)$.

Finally, in the third loop iteration, the direction is back towards S and so the extension $\mathcal{K}_3$ of $\mathcal{K}_2$ to $H_1H_2H_3$ contains $\lfloor H_2 \rfloor$ and is $\mathfrak{N}_2$-refined. However, the group $\lfloor H_1 \rfloor$ is not a direct factor of G as it is one term in nontrivial central product. Therefore that group is replaced by a subgroup $\lfloor H_1H_3 \rfloor \cong \mathrm{SL}(d, q) \circ \mathrm{SL}(d, q)$. The final configuration is illustrated in Figure 6. $\mathcal{K}_3$ is a Remak decomposition of G.

8. Closing remarks

Historically the problem of finding a Remak decomposition focused on groups given by their multiplication table since even there there did not seem to be a polynomial-time solution. It was known that a Remak decomposition could be found by checking all partitions of all minimal generating sets of a group G and so the problem had a sub-exponential complexity of $|G|^{\log |G| + O(1)}$. That placed it in the company of other interesting problems including testing for an isomorphism between two groups [21]. Producing an algorithm that is polynomial-time in the size of the group's multiplication table (i.e. polynomial in $|G|^2$) was progress, achieved independently in [14] and [34]. Evidently, Theorem 1.1 provides a polynomial-time solution for groups input in this way (e.g. use a regular representation). With a few observations we sharpen Theorem 1.1 in that specific context to the following:

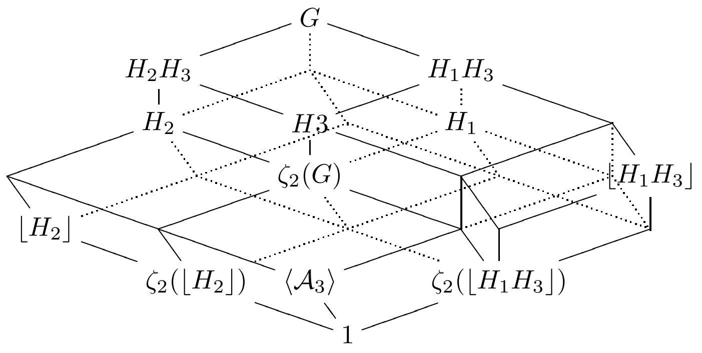

G
H_2H_3
H_1H_3
H_2
$H3$
H_1
$\zeta_2(G)$
$\lfloor H_1H_3 \rfloor$
$\lfloor H_2 \rfloor$
$\zeta_2(\lfloor H_2 \rfloor)$
$\langle \mathcal{A}_3 \rangle$
$\zeta_2(\lfloor H_1H_3 \rfloor)$
1

FIGURE 6. The lattice of encountered during the third iteration of the loop in the algorithm $\texttt{Merge}(\mathcal{A}, \{H_1, H_2, H_3\})$.

Theorem 8.1. *There is a deterministic nearly-linear-time algorithm which, given a group's multiplication table, returns a Remak decomposition of the group.*

Proof. The algorithm for Theorem 6.4 is polynomial in $\log|G|$. As the input length here is $|G|^2$, it suffices to show that the problems listed in Section 2.6 have $O(|G|^2 \log^c |G|)$-time or better solutions. Evidently, ORDER, MEMBER, SOLVE each have brute-force linear-times solutions. PRESENTATION can be solved in linear-time by selecting a minimal generating set $\{g_1, \ldots, g_\ell\}$ (which has size $\log|G|$) and acting on the cosets of $\{\langle g_i, \ldots, g_\ell\rangle : 1 \leq i \leq \ell\}$ produce defining relations of the generators in fashion similar to [32, Exercise 5.2]. For MINIMAL-NORMAL, begin with an element and takes it normal closure. If this is a proper subgroup recurse, otherwise, try an element which is not conjugate to the first and repeat until either a proper normal subgroup is discovered or it is proved that group is simple. That takes $O(|G|^2)$-time. The remaining algorithms PRIMARY-DECOMPOSITION, IRREDUCIBLE, and FRAME have brute force linear-time solutions. Thus, the algorithm can be modified to run in times $O(|G|^2 \log^c |G|)$. $\square$

Section 3 lays out a framework which permits for a local view of the direct products of group. We have some lingering questions in this area.

(1) What is the best series of subgroups to use for the algorithm of Theorem 6.4?

Corollaries 3.14 and 3.21 offer alternatives series to use in the algorithm. There is an option for a top-down algorithm based on down graders. That may allow for a black-box algorithm since verbal subgroups can be constructed in black-box groups; see [32, Section 2.3.4].

(2) Is their a parallel NC solution for REMAK-Ω-DECOMPOSITION?

We can speculate how this may proceed. First, select an appropriate series $1 \leq G_1 \leq \cdots \leq G_n = G$ for G and distribute and use parallel linear algebra methods to find Remak decompositions $\mathcal{A}_{i0}$ of each G_{i+1}/G_i, for $1 \leq i < n$. Then for $0 \leq j \leq \log n$, for each $1 \leq i \leq n/2^j$ in parallel compute $\mathcal{A}_{i(j+1)} := \text{MERGE}(\mathcal{A}_{ij}, \mathcal{A}_{(i+1)j})$. When $j = \lfloor \log n \rfloor$ we have a direct decomposition $\mathcal{A}_{1\log n}$ of G and have used poly-logarithmic time.