Single Family Algebra Operation on BDDs and ZDDs Leads to Exponential Blow-Up

Kengo Nakamura ✉ ⓘ
NTT Communication Science Laboratories, Kyoto, Japan

Masaaki Nishino ✉ ⓘ
NTT Communication Science Laboratories, Kyoto, Japan

Shuhei Denzumi ✉ ⓘ
NTT Communication Science Laboratories, Kyoto, Japan

─── **Abstract** ───

Binary decision diagram (BDD) and zero-suppressed binary decision diagram (ZDD) are data structures to represent a family of (sub)sets compactly, and it can be used as succinct indexes for a family of sets. To build BDD/ZDD representing a desired family of sets, there are many transformation operations that take BDDs/ZDDs as inputs and output BDD/ZDD representing the resultant family after performing operations such as set union and intersection. However, except for some basic operations, the worst-time complexity of taking such transformation on BDDs/ZDDs has not been extensively studied, and some contradictory statements about it have arisen in the literature. In this paper, we show that many transformation operations on BDDs/ZDDs, including all operations for families of sets that appear in Knuth's book, cannot be performed in worst-case polynomial time in the size of input BDDs/ZDDs. This refutes some of the folklore circulated in past literature and resolves an open problem raised by Knuth. Our results are stronger in that such blow-up of computational time occurs even when the ordering, which has a significant impact on the efficiency of treating BDDs/ZDDs, is chosen arbitrarily.

1 Introduction

Combinatorial problems, i.e., the problems dealing with combinations of a set, frequently arise in several situations such as operations research, network analysis, and LSI design. In solving such problems, it is often convenient to consider the set of combinations, i.e., the *family of (sub)sets*. For example, many combinatorial optimization problems can be formulated as selecting the best combination (subset) from the family of sets satisfying constraints. However, the number of sets in a family is possibly exponential, precluding us from explicitly retaining the family of sets.

To alleviate this issue, we can use *binary decision diagram* (*BDD*) [2] or *zero-suppressed binary decision diagram* (*ZDD*) [14] that is a variant of BDD. BDD and ZDD are data structures that compactly represent a Boolean function and a family of sets, respectively. Since a Boolean function f can be regarded as a family of sets by considering the set of assignments of input Boolean variables that evaluates f to *true*, BDD can also be regarded as a succinct representation of a family of sets. Moreover, they support many queries about the represented family of sets, e.g., counting the number of sets and performing linear optimization over the family. Thus, BDD and ZDD can be used as succinct indexes for a family of sets.

BDDs and ZDDs also support a number of transformation operations. For example, when we have two BDDs representing two families of sets, we can construct a BDD representing the set union of them without extracting each set from the input families. Using such operations, we can construct a BDD or a ZDD representing the desired family of sets. By collecting such transformation operations, Minato [15] considered an algebraic system called *unate cube set algebra*, whose element is a family of sets. After that, many operations were introduced, and now the system is widely called *family algebra*, whose name was given by Knuth [13]. With the algorithms performing operations on BDDs and ZDDs, every operation in the family algebra provides a useful way to construct a BDD or a ZDD representing the desired family of sets in many applications. Many of these operations have been implemented in standard BDD and ZDD manipulation packages [8, 18], and they are used in a wide range of applications, including formal verification of circuits [7, 10], analyses of power distribution networks [9, 19], and data mining [16].

However, the complexity of performing family algebra operations on BDDs and ZDDs has not been well studied, except for basic set operations. This is because some operations require complicated recursion procedures that make complexity analysis difficult. In particular, revealing *worst-case* time complexity is important to us. If the worst-case time complexity is large, it takes an unexpectedly long time to carry out even a *single* operation for certain kinds of input. If so, we should pay attention to the possibility of such input when we use BDDs and ZDDs as a way to implement the manipulation of families of sets. Therefore, we investigated the worst-case time complexity of executing a single family algebra operation on BDDs and ZDDs. Since it is known that, as described later, the sizes of a BDD and a ZDD representing the same family of sets differ in only a linear factor, this paper mainly focused on the complexity of ZDDs. After that, we mention the complexity on BDDs.

1.1 Related Work

Since the invention of ZDD [14], many family algebra operations have been proposed. Table 1 lists basic operations. As related work, we first describe the origins of these operations.

The first four operations in Table 1 are the most fundamental set operations set described by Minato [14]. The join, quotient, and remainder operations appeared in Minato's next paper [15], where the join operation is called "product" because a join can be considered to be the multiplication of two families when we view the union operation as an addition operation. These operations are peculiar to the families of sets and also fundamental in defining other family algebra operations. Later, the disjoint join and joint join operations were proposed by Kawahara et al. [12] through an extension of the join; their usage is to implicitly enumerate all of the subgraphs having a particular shape.

Restrict and permit operations were originally proposed by Coudert et al. [5], where they were called SupSet and SubSet and used for solving set cover problems or performing logic circuit minimization. The names "restrict" and "permit" come from a study by Okuno et

Table 1 List of operations on family algebra.

Operation	Definition	Is polytime in DD sizes?
Union $\mathcal{F} \cup \mathcal{G}$	$\{S \mid S \in \mathcal{F} \vee S \in \mathcal{G}\}$	Yes [14]
Intersection $\mathcal{F} \cap \mathcal{G}$	$\{S \mid S \in \mathcal{F} \wedge S \in \mathcal{G}\}$	Yes [14]
Difference $\mathcal{F} \setminus \mathcal{G}$	$\{S \mid S \in \mathcal{F} \wedge S \notin \mathcal{G}\}$	Yes [14]
Symmetric difference $\mathcal{F} \oplus \mathcal{G}$	$(\mathcal{F} \setminus \mathcal{G}) \cup (\mathcal{G} \setminus \mathcal{F})$	Yes [14]
Join $\mathcal{F} \sqcup \mathcal{G}$	$\{F \cup G \mid F \in \mathcal{F}, G \in \mathcal{G}\}$	**No** (Theorem 7)*
Disjoint join $\mathcal{F} \bowtie \mathcal{G}$	$\{F \cup G \mid F \in \mathcal{F}, G \in \mathcal{G}, F \cap G = \emptyset\}$	**No** (Theorem 7)
Joint join $\mathcal{F} \bowtie \mathcal{G}$	$\{F \cup G \mid F \in \mathcal{F}, G \in \mathcal{G}, F \cap G \neq \emptyset\}$	**No** (Theorem 7)
Meet $\mathcal{F} \sqcap \mathcal{G}$	$\{F \cap G \mid F \in \mathcal{F}, G \in \mathcal{G}\}$	**No** (Theorem 7)*
Delta $\mathcal{F} \boxplus \mathcal{G}$	$\{F \oplus G \mid F \in \mathcal{F}, G \in \mathcal{G}\}$	**No** (Theorem 7)*
Quotient $\mathcal{F} / \mathcal{G}$	$\{S \mid \forall G \in \mathcal{G} : S \cup G \in \mathcal{F} \wedge S \cap G = \emptyset\}$	**No** (Theorem 9)
Remainder $\mathcal{F} \% \mathcal{G}$	$\mathcal{F} \setminus (\mathcal{G} \sqcup (\mathcal{F} / \mathcal{G}))$	**No** (Theorem 9)
Restrict $\mathcal{F} \triangle \mathcal{G}$	$\{F \in \mathcal{F} \mid \exists G \in \mathcal{G} : G \subseteq F\}$	**No** (Theorem 10)*
Permit $\mathcal{F} \oslash \mathcal{G}$	$\{F \in \mathcal{F} \mid \exists G \in \mathcal{G} : F \subseteq G\}$	**No** (Theorem 10)
Nonsuperset $\mathcal{F} \searrow \mathcal{G}$	$\{F \in \mathcal{F} \mid \forall G \in \mathcal{G} : G \not\subseteq F\}$	**No** (Theorem 10)
Nonsubset $\mathcal{F} \nearrow \mathcal{G}$	$\{F \in \mathcal{F} \mid \forall G \in \mathcal{G} : F \not\subseteq G\}$	**No** (Theorem 10)
Maximal $\mathcal{F}^{\uparrow}$	$\{F \in \mathcal{F} \mid \forall F' \in \mathcal{F} : F \subseteq F' \Rightarrow F = F'\}$	**No** (Theorem 11)
Minimal $\mathcal{F}^{\downarrow}$	$\{F \in \mathcal{F} \mid \forall F' \in \mathcal{F} : F' \subseteq F \Rightarrow F = F'\}$	**No** (Theorem 11)
Minimal hitting set $\mathcal{F}^{\sharp}$	$\{S \mid \forall F \in \mathcal{F} : S \cap F \neq \emptyset\}^{\downarrow}$	**No** (Theorem 12)
Closure $\mathcal{F}^{\cap}$	$\{\bigcap_{S \in \mathcal{F}'} S \mid \mathcal{F}' \subseteq \mathcal{F}\}$	**No** (Theorem 12)

*Previous studies [17, 13] stated that they can be performed in worst-case polynomial time.

al. [17]. Later, nonsuperset, nonsubset, maximal, and minimal operations were introduced by Coudert [4] to solve various optimization problems on graphs. Furthermore, meet, delta, minimal hitting set, and closure operations were introduced by Knuth [13, §7.1.4 Ex.203,236,243] to solve various graph problems. Table 1 contains all of the transformation operations for families of sets that appeared in Knuth's book [13, §7.1.4 Ex. 203,204,236,243].

Compared to the operations themselves, the time complexity of performing them on ZDDs has not been well investigated. Minato [14] proved that the first four operations in Table 1 can be performed in polynomial time with respect to the size of input ZDDs. However, the complexity of a join operation, the most basic one among the rest, has not been fully clarified. Knuth [13, §7.1.4 Ex. 206] claimed that join, as well as meet and delta, can be performed in worst-case polynomial time, but this claim lacks proof. Conversely, Kawahara et al. [12] suggested that join, as well as disjoint join and joint join, take worst-case exponential time, again without proof. In addition to those reports, Okuno et al. [17] claimed that restrict can be performed in polynomial time, but they used the unproven proposition that join can be performed in polynomial time. Furthermore, Knuth [13, §7.1.4 Ex. 206] stated that the worst-case complexity of the quotient operation was an open problem.

1.2 Our Contribution

In this paper, we prove that, for the operations in Table 1 aside from the first four operations, there exist polynomial-sized ZDDs such that after taking the operation, the ZDD size becomes exponential. For example, for the join operation, we prove that there exist sequences of families of sets $\{\mathcal{F}_m\}$ and $\{\mathcal{G}_m\}$ such that the ZDD sizes representing $\mathcal{F}_m$ and $\mathcal{G}_m$ are polynomial in m, while the ZDD size representing $\mathcal{F}_m \sqcup \mathcal{G}_m$ is exponential in m. This result implies that these operations cannot be performed in worst-case polynomial time with respect to the size of input ZDDs. Thus, we refute the statement raised by Knuth [13] and Okuno et al. [17] that join, meet, delta, and restrict can be performed in worst-case polynomial time.

We also resolve the worst-case complexity of the quotient operation. Moreover, we also prove that the operations in Table 1, except for the first four operations, cannot be performed in polynomial time even when families are represented by BDDs. Since Table 1 contains all the family algebra operations raised by Knuth [13], this paper concludes what kind of family algebra operations can be performed in polynomial time on BDDs and ZDDs.

Our result is stronger in that the resultant BDD/ZDD's size remains exponential for any *order of elements*. BDD/ZDD structures follow a total order of the elements in the base set, and it is known that this element order has a significant impact on the BDD/ZDD size. For example, it is known that a multiplexer function can be represented in linear-sized BDD by managing the ordering while its size becomes exponential when the ordering is terrible [13, p.235]. However, we also prove that for the sequences used in proving the above, the resultant BDD/ZDD's size is exponential in m regardless of the order of elements. This suggests that we cannot shrink the BDD/ZDD size after taking an operation by managing the element order. Some famous BDD manipulation packages such as CUDD [18] implemented dynamic reordering, the reordering of elements after executing operations to shrink the BDD/ZDD size and thus increase the efficiency of BDD/ZDD manipulations. Nevertheless, our results suggest that the worst-case complexity of carrying out operations cannot be polynomial, even if we employ dynamic reordering.

Note that this follows the research line of Bollig [1] as follows. Yoshinaka et al. [20] refuted Bryant's conjecture, which is about the complexity of performing operations on BDDs, but their counterexample was somewhat weak in that the order of elements they used was unfavorable for BDD representations. Bollig [1] later resolved this issue by proposing simpler counterexamples. Similar to this, our results imply that the exponential blow-up in taking an operation on BDDs/ZDDs occurs not only when the order of elements is unfavorable but also when it is good for BDD/ZDD representations.

From the viewpoint of applications, BDDs/ZDDs are usually built by applying multiple family algebra operations in combination with some direct construction methods such as Simpath [13] and frontier-based search [11], which are fixed-parameter tractable algorithms with pathwidth. However, the number of required operations stays constant in many applications. If every operation can be performed in polynomial time, we can enjoy the polynomial time complexity in BDD/ZDD sizes even for these applications. However, our results suggest this is not the case except for the first four operations. In addition, although we rely on specific input examples to prove non-polynomial lower bounds, we later discuss that such blow-up may occur for other input; the detailed discussions are in Section 3.5. Therefore, our theoretical results have practical importance.

2   Preliminaries

2.1   Zero-suppressed Binary Decision Diagram

A *zero-suppressed binary decision diagram* (*ZDD*) [14] is a rooted directed acyclic graph (DAG)-shaped data structure for representing a family of sets. First, we describe the structure of ZDD. ZDD `Z` consists of node set `N` and arc set `A`, where the node set contains *terminal* nodes $\top, \bot$ and other internal nodes. Terminal nodes have no outgoing arcs, while every internal node has two outgoing arcs called *lo-arc* and *hi-arc*. The nodes pointed by the lo-arc and the hi-arc outgoing from a node `n` are called *lo-child* `lo(n)` and *hi-child* `hi(n)` of `n`. Every internal node `n` is associated with an element called *label* that is denoted by `lb(n)`. ZDDs must follow the *ordered property*: Given a total order of elements $<$, the label of the parent
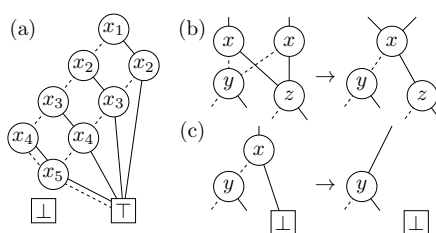
Figure 1 (a) Example of a ZDD representing the family of subsets of $\{x_1, \ldots, x_5\}$ such that the cardinality is less than 3. (b) Schematic of node sharing. (c) Schematic of zero suppression.

node must precede that of the child node, i.e., $\mathsf{lb}(\mathtt{n}) < \mathsf{lb}(\mathsf{lo}(\mathtt{n}))$ and $\mathsf{lb}(\mathtt{n}) < \mathsf{lb}(\mathsf{hi}(\mathtt{n}))$ must hold for every internal node $\mathtt{n}$. Note that the child node is always allowed to be a terminal node. Finally, the size of a ZDD is defined by its number of nodes.

Next, we describe the semantics of ZDD.

▶ **Definition 1.** *For ZDD node* $\mathtt{n}$*, the family* $\mathcal{F}_\mathtt{n}$ *of sets represented by* $\mathtt{n}$ *is defined as follows.* *(i) If* $\mathtt{n} = \top$*, then* $\mathcal{F}_\mathtt{n} = \{\emptyset\}$*.* *(ii) If* $\mathtt{n} = \bot$*, then* $\mathcal{F}_\mathtt{n} = \emptyset$*.* *(iii) Otherwise,* $\mathcal{F}_\mathtt{n} = \mathcal{F}_{\mathsf{lo}(\mathtt{n})} \cup (\{\{\mathsf{lb}(\mathtt{n})\}\} \sqcup \mathcal{F}_{\mathsf{hi}(\mathtt{n})})$*. Furthermore, the family of sets represented by* $\mathtt{Z}$ *is that represented by root node* $\mathtt{r}$*, where the root node is the only node having no incoming arcs.*

Note that $\{\emptyset\}$ and $\emptyset$ are different families; the former is the family consisting of only an empty set, while the latter is the family containing no set. For example, Figure 1a is the ZDD representing the family of subsets of $\{x_1, \ldots, x_5\}$ whose cardinality is less than 3. Solid and dashed lines represent hi- and lo-arcs, and the element inside a circle indicates its label.

Without restrictions on the structure, there exist many ZDDs representing the same family of sets. However, by imposing restrictions, we can obtain a *canonical* ZDD, i.e., an identical ZDD structure, for every family of subsets. This canonical form is called *reduced ZDD*, and a reduced ZDD can be obtained from any ZDD by repetitively applying the following two rules. The first rule is *node sharing*: If there exist two nodes $\mathtt{n}$ and $\mathtt{m}$ whose lo-child, hi-child, and label are equal, we merge these two nodes into one (Figure 1b). The second rule is *zero suppression*: If there exists a node $\mathtt{n}$ whose hi-child is $\bot$, we eliminate $\mathtt{n}$ and let all of the arcs pointed to $\mathtt{n}$ also point to $\mathsf{hi}(\mathtt{n})$ (Figure 1c). In the reduced ZDD, no node can be eliminated by applying the above two rules. Since applying these rules strictly decreases the size of ZDD, i.e., the number of nodes, we can deduce that the reduced ZDD of a family $\mathcal{F}$ is the smallest ZDD representing $\mathcal{F}$ given the total order $<$ of elements. The size of the reduced ZDD of the family $\mathcal{F}$, given the total order $<$, is denoted by $Z_<(\mathcal{F})$. If it is clear from the context, we omit $<$ and simply write it as $Z(\mathcal{F})$.

We briefly compare ZDDs with BDDs. BDD [2] has the same structure (syntax) as ZDD, although its semantics is slightly different. BDDs also follow the ordered property and have the smallest canonical form called *reduced BDD*. Given the total order $<$ of elements, the size of the reduced BDD of the family $\mathcal{F}$ is denoted by $B_<(\mathcal{F})$. The following is a famous result.

▶ **Lemma 2** ([13, Eq. (126)])**.** *For any family* $\mathcal{F}$ *of subsets of a set of* n *elements and any order* $<$ *of elements,* $B_<(\mathcal{F}) = O(nZ_<(\mathcal{F}))$ *and* $Z_<(\mathcal{F}) = O(nB_<(\mathcal{F}))$*.*

2.2 Family Algebra Operations on ZDDs

In this section, we explain how the family algebra operations are performed using ZDDs and point out what makes the difference between the basic set operations (union, intersection, difference, and symmetric difference) and the other operations.

As explained in Section 2.1, ZDD represents a family of sets in a recursive manner. Let us consider the situation in which there are two ZDDs whose root nodes are $\mathtt{n}$ and $\mathtt{m}$ and $\mathsf{lb}(\mathtt{n}) = \mathsf{lb}(\mathtt{m}) = x$. Then, the family of sets represented by them are $\mathcal{F}_{\mathtt{n}} = \mathcal{F}_{\mathsf{lo(n)}} \cup (\{\{x\}\} \sqcup \mathcal{F}_{\mathsf{hi(n)}})$ and $\mathcal{F}_{\mathtt{m}} = \mathcal{F}_{\mathsf{lo(m)}} \cup (\{\{x\}\} \sqcup \mathcal{F}_{\mathsf{hi(m)}})$. The union of them is

$$\mathcal{F}_{\mathtt{n}} \cup \mathcal{F}_{\mathtt{m}} = [\mathcal{F}_{\mathsf{lo(n)}} \cup \mathcal{F}_{\mathsf{lo(m)}}] \cup [\{\{x\}\} \sqcup (\mathcal{F}_{\mathsf{hi(n)}} \cup \mathcal{F}_{\mathsf{hi(m)}})]. \tag{1}$$

This means that the ZDD representing $\mathcal{F}_{\mathtt{n}} \cup \mathcal{F}_{\mathtt{m}}$ can be described as follows: The root node's label is x, its lo-child represents $\mathcal{F}_{\mathsf{lo(n)}} \cup \mathcal{F}_{\mathsf{lo(m)}}$, and its hi-child represents $\mathcal{F}_{\mathsf{hi(n)}} \cup \mathcal{F}_{\mathsf{hi(m)}}$. If $\mathsf{lb}(\mathtt{n}) < \mathsf{lb}(\mathtt{m})$, we have a simpler recursion:

$$\mathcal{F}_{\mathtt{n}} \cup \mathcal{F}_{\mathtt{m}} = [\mathcal{F}_{\mathsf{lo(n)}} \cup \mathcal{F}_{\mathtt{m}}] \cup [\{\{\mathsf{lb}(\mathtt{n})\}\} \sqcup (\mathcal{F}_{\mathsf{hi(n)}} \cup \mathcal{F}_{\mathtt{m}})]. \tag{2}$$

The case of $\mathsf{lb}(\mathtt{m}) < \mathsf{lb}(\mathtt{n})$ can be handled in the same way. By recursively expanding $\mathcal{F}_{\mathtt{n}} \cup \mathcal{F}_{\mathtt{m}}$ by (1) and (2), we eventually reach terminal nodes where the union is trivial, e.g., $\mathcal{F}_{\perp} \cup \mathcal{F}_{\top} = \{\emptyset\}$. Therefore, by caching the resultant ZDD nodes of $\mathcal{F}_{\mathtt{n}'} \cup \mathcal{F}_{\mathtt{m}'}$, where $\mathtt{n}'$ and $\mathtt{m}'$ are the child nodes of $\mathtt{n}$ and $\mathtt{m}$, respectively, we can efficiently compute the ZDD representing $\mathcal{F}_{\mathtt{n}} \cup \mathcal{F}_{\mathtt{m}}$. With the cache, one can show that we can build a ZDD representing the union of two ZDDs in a time proportional to the product of input ZDD sizes. The intersection, difference, and symmetric difference operations can be handled in almost the same way.

The other operations can also be performed in a recursive manner. However, the recursion becomes more complicated. Let us consider, for example, the join operation. When $\mathsf{lb}(\mathtt{n}) = \mathsf{lb}(\mathtt{m}) = x$, the join becomes

$$
\begin{aligned}
\mathcal{F}_{\mathtt{n}} \sqcup \mathcal{F}_{\mathtt{m}} = & [\mathcal{F}_{\mathsf{lo(n)}} \cup (\{\{x\}\} \sqcup \mathcal{F}_{\mathsf{hi(n)}})] \sqcup [\mathcal{F}_{\mathsf{lo(m)}} \cup (\{\{x\}\} \sqcup \mathcal{F}_{\mathsf{hi(m)}})] \\
= & [\mathcal{F}_{\mathsf{lo(n)}} \sqcup \mathcal{F}_{\mathsf{lo(m)}}] \cup [\mathcal{F}_{\mathsf{lo(n)}} \sqcup (\{\{x\}\} \sqcup \mathcal{F}_{\mathsf{hi(m)}})] \cup \\
& [(\{\{x\}\} \sqcup \mathcal{F}_{\mathsf{hi(n)}}) \sqcup \mathcal{F}_{\mathsf{lo(m)}}] \cup [(\{\{x\}\} \sqcup \mathcal{F}_{\mathsf{hi(n)}}) \sqcup (\{\{x\}\} \sqcup \mathcal{F}_{\mathsf{hi(m)}})] \\
= & [\mathcal{F}_{\mathsf{lo(n)}} \sqcup \mathcal{F}_{\mathsf{lo(m)}}] \cup [\{\{x\}\} \sqcup (\mathcal{F}_{\mathsf{lo(n)}} \sqcup \mathcal{F}_{\mathsf{hi(m)}})] \cup \\
& [\{\{x\}\} \sqcup (\mathcal{F}_{\mathsf{hi(n)}} \sqcup \mathcal{F}_{\mathsf{lo(m)}})] \cup [\{\{x\}\} \sqcup (\mathcal{F}_{\mathsf{hi(n)}} \sqcup \mathcal{F}_{\mathsf{hi(m)}})] \\
= & [\mathcal{F}_{\mathsf{lo(n)}} \sqcup \mathcal{F}_{\mathsf{lo(m)}}] \cup [\{\{x\}\} \sqcup ((\mathcal{F}_{\mathsf{lo(n)}} \sqcup \mathcal{F}_{\mathsf{hi(m)}}) \cup (\mathcal{F}_{\mathsf{hi(n)}} \sqcup \mathcal{F}_{\mathsf{lo(m)}}) \cup (\mathcal{F}_{\mathsf{hi(n)}} \sqcup \mathcal{F}_{\mathsf{hi(m)}}))].
\end{aligned}
\tag{3}
$$

Here, the second equality holds because join distributes over the union. This means that we should build a ZDD where the root node's lo-child represents $\mathcal{F}_{\mathsf{lo(n)}} \sqcup \mathcal{F}_{\mathsf{lo(m)}}$ and its hi-child represents $(\mathcal{F}_{\mathsf{lo(n)}} \sqcup \mathcal{F}_{\mathsf{hi(m)}}) \cup (\mathcal{F}_{\mathsf{hi(n)}} \sqcup \mathcal{F}_{\mathsf{lo(m)}}) \cup (\mathcal{F}_{\mathsf{hi(n)}} \sqcup \mathcal{F}_{\mathsf{hi(m)}})$. Thus, in the recursion, we should also compute the union $\cup$ of families, which also needs a recursion like that above. Another example is the restrict operation. Restrict can be computed as

$$\mathcal{F}_{\mathtt{n}} \bigtriangleup \mathcal{F}_{\mathtt{m}} = [\mathcal{F}_{\mathsf{lo(n)}} \bigtriangleup \mathcal{F}_{\mathsf{lo(m)}}] \cup [\{\{x\}\} \sqcup (\mathcal{F}_{\mathsf{hi(n)}} \bigtriangleup (\mathcal{F}_{\mathsf{lo(m)}} \cup \mathcal{F}_{\mathsf{hi(m)}}))]. \tag{4}$$

Thus, it is also necessary to compute the union of families as well as restrict.

Compared to the simple recursion for the computation of basic set operations, the complexity of such "double recursion" procedures are difficult to analyze.

3 Blow-Up Operations

3.1 High-Level Idea

As described in Section 2.2, the ZDD size after performing union or intersection can be bounded by the product of the sizes of operand ZDDs, i.e., $Z(\mathcal{F} \cup \mathcal{G}) = O(Z(\mathcal{F})Z(\mathcal{G}))$ and $Z(\mathcal{F} \cap \mathcal{G}) = O(Z(\mathcal{F})Z(\mathcal{G}))$. Thus, the ZDD of the union or intersection of *two* ZDDs remains

polynomial-sized when the operand ZDDs have polynomial size. However, this does not hold for a non-constant number of ZDDs: even if $Z(\mathcal{F}_k) = O(\text{poly}(m))$ for $k = 1, \ldots, m$, both $Z(\bigcup_{k=1}^m \mathcal{F}_k)$ and $Z(\bigcap_{k=1}^m \mathcal{F}_k)$ may become exponential in m.

We use such families to constitute examples of blow-up. More specifically, for each operation, we constitute an example such that performing this operation incurs the union or intersection of multiple families. Since we prove that the reduced ZDD representing the result of an operation will become exponential in size, we can confirm that *any* algorithm for computing the resultant ZDD incurs worst-case non-polynomial complexity. Combined with concrete instances, we prove that the worst-case complexity of family algebra operations is lower-bounded by an exponential factor.

We use the specific families of sets, hidden weighted bit function and permutation function, as explained below. Note that they are called "function" because they are originally defined as a Boolean function, but we here describe them as equivalent families of sets.

▶ **Definition 3.** *A hidden weighted bit function $\mathcal{H}_m$ is a family of sets defined as $\{S \subseteq \{y_1, \ldots, y_m\} \mid y_{|S|} \in S\}$.*

The hidden weighted bit function $\mathcal{H}_m$ can be represented as a union of elementary families. Define $\mathcal{E}_{m,k} := \{S \subseteq \{y_1, \ldots, y_m\} \mid |S| = k, y_k \in S\}$, i.e., $\mathcal{E}_{m,k}$ consists of the subsets of $\{y_1, \ldots, y_m\}$ where the cardinality is k and y_k is contained. Then, $\mathcal{H}_m = \bigcup_{k=1}^m \mathcal{E}_{m,k}$. It can be easily verified that the size of the ZDD representing $Z(\mathcal{E}_{m,k})$ is $O(m^2)$ for any order of elements (see Section 3.4). However, it is known that the ZDD representing $\mathcal{H}_m$ must become exponential in size.

▶ **Theorem 4** ([3]). *For any order $<$ of elements, $B_<(\mathcal{H}_m) = \Omega(2^{m/5})$. Thus, by Lemma 2, $Z_<(\mathcal{H}_m) = \Omega(2^{m/5}/m)$.*

▶ **Definition 5.** *A permutation function $\mathcal{P}_m$ is a family of subsets of $\{y_1, \ldots, y_{m^2}\}$ such that (i) there is exactly one element from $y_{m(i-1)+1}, y_{m(i-1)+2}, \ldots, y_{m(i-1)+m}$ for $i = 1, \ldots, m$, and (ii) there is exactly one element from $y_j, y_{m+j}, \ldots, y_{m(m-1)+j}$ for $j = 1, \ldots, m$.*

The permutation function $\mathcal{P}_m$ is equivalent to the set of permutations: For $S \subseteq \{y_1, \ldots, y_{m^2}\}$, we associate a binary $m \times m$ matrix where the (i, j)-element is 1 if and only if $y_{m(i-1)+j} \in S$. Then, $S \in \mathcal{P}_m$ if and only if the associated matrix is a permutation matrix.

For $k = 1, \ldots, m$, let $\mathcal{Q}_{m,k}$ be the family of subsets of $\{y_1, \ldots, y_{m^2}\}$ such that there is exactly one element from $y_{m(k-1)+1}, y_{m(k-1)+2}, \ldots, y_{m(k-1)+m}$, and let $\mathcal{Q}_{m,m+k}$ be those such that there is exactly one element from $y_k, y_{m+k}, \ldots, y_{m(m-1)+k}$. Then, $\mathcal{P}_m = \bigcap_{k=1}^{2m} \mathcal{Q}_{m,k}$. Here, $Z(\mathcal{Q}_{m,k}) = O(m^2)$ for any order of elements, as proved in Section 3.4. However, it is again proved that the ZDD representing $\mathcal{P}_m$ must become exponential in size.

▶ **Theorem 6** ([13, Theorem K]). *For any order $<$ of elements, $B_<(\mathcal{P}_m) = \Omega(m2^m)$. Thus, by Lemma 2, $Z_<(\mathcal{P}_m) = \Omega(2^m/m)$.*

We first show the exponential blow-up cases for a specific order of elements in Section 3.2. However, we see that the size of ZDD representing the hidden weighted bit function or the permutation function is exponential regardless of the order of elements. Therefore, in Section 3.3, we prove that for each family generated by the operation in Section 3.2, the ZDD size representing it remains exponential regardless of the order of elements. This means that for each operation, there exists an instance in which the input ZDD size can be polynomial by managing the element order but the output ZDD size must be exponential for any order. Section 3.4 completes the proof by showing that some families can be represented by polynomial-sized ZDDs. Finally, Section 3.5 gives some discussions on the obtained result.
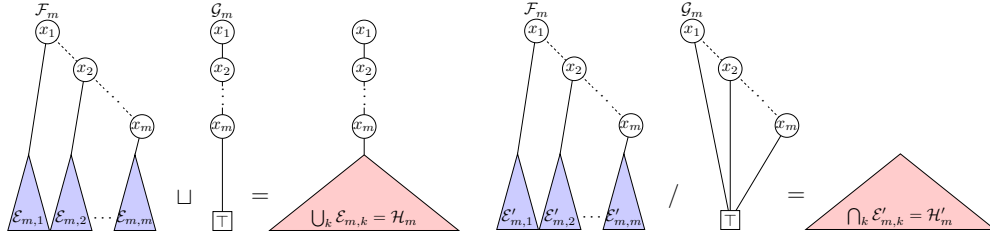
Figure 2 Example of blow-up for join (left) and quotient (right) operations. Blue triangles mean that the ZDD size representing this family is polynomial in m, while red triangle means that its size is exponential in m. Arcs going to $\bot$ terminal are omitted.

3.2 Proofs with Specific Element Order

3.2.1 Join, Disjoint Join, Joint Join, Meet, and Delta

For these operations, we constitute a pair of families that incur the union of $O(m)$ subfamilies. Combined with $\mathcal{E}_{m,k}$, the result after taking an operation contains $\bigcup_k \mathcal{E}_{m,k} = \mathcal{H}_m$, which is the hidden weighted bit function for which the ZDD size is exponential in m.

▶ **Theorem 7.** *Let $\diamond$ be a binary operator chosen from join ($\sqcup$), disjoint join ($\dot\bowtie$), joint join ($\hat\bowtie$), meet ($\sqcap$), and delta ($\boxplus$). Then, there exists a sequence of families $\mathcal{F}_m$ and $\mathcal{G}_m$ such that (i) $\mathcal{F}_m$ and $\mathcal{G}_m$ are families of subsets of a set of $O(m)$ elements, (ii) $Z(\mathcal{F}_m) + Z(\mathcal{G}_m) = O(m^3)$, and (iii) $Z(\mathcal{F}_m \diamond \mathcal{G}_m) = \Omega(2^{m/5}/m)$.*

Proof. Let us consider the families of subsets of $X \cup Y$, where $X := \{x_1, \ldots, x_m\}$ and $Y := \{y_1, \ldots, y_m\}$. We determine the order of elements as $x_1, \ldots, x_m, y_1, \ldots, y_m$. We define $\mathcal{F}_m$ as

$$\mathcal{F}_m := \bigcup_{k=1}^{m}(\{\{x_k\}\} \sqcup \mathcal{E}_{m,k}).$$

Since $Z(\mathcal{E}_{m,k}) = O(m^2)$ and the ZDD representing $\mathcal{F}_m$ becomes the left one of Figure 2 according to this order, $Z(\mathcal{F}_m) = O(m^3)$.

For the join operation, we let $\mathcal{G}_m := \{X\}$, where $Z(\mathcal{G}_m) = O(m)$. Then,

$$\mathcal{F}_m \sqcup \mathcal{G}_m = (\bigcup_{k=1}^{m}(\{\{x_k\}\} \sqcup \mathcal{E}_{m,k})) \sqcup \{X\} = \bigcup_{k=1}^{m}((\{\{x_k\}\} \sqcup \mathcal{E}_{m,k}) \sqcup \{X\})$$
$$= \bigcup_{k=1}^{m}(\{X\} \sqcup \mathcal{E}_{m,k}) = \{X\} \sqcup (\bigcup_{k=1}^{m} \mathcal{E}_{m,k}) = \{X\} \sqcup \mathcal{H}_m,$$

where the second and fourth equalities hold because join distributes over union and the third equality holds because $\{\{x_k\}\} \sqcup \{X\} = \{X\}$. Thus, the ZDD representing $\mathcal{F}_m \sqcup \mathcal{G}_m$ becomes the right one of Figure 2, meaning that the ZDD size is at least $Z(\mathcal{H}_m) = \Omega(2^{m/5}/m)$. Since every subset in $\mathcal{F}_m$ has at least one element from X, the result of joint join $\mathcal{F}_m \hat\bowtie \mathcal{G}_m$ also becomes $\{X\} \sqcup \mathcal{H}_m$, leading to an exponential-sized ZDD.

For the disjoint join operation, we let $\mathcal{G}_m := \bigcup_{k=1}^{m}\{X \setminus \{x_k\}\}$, where again $Z(\mathcal{G}_m) = O(m)$. Then, every subset in $\{\{x_k\}\} \sqcup \mathcal{E}_{m,k}$ has intersection with all of the subsets in $\mathcal{G}_m$, except for $X \setminus \{x_k\}$. Then,

$$\mathcal{F}_m \dot\bowtie \mathcal{G}_m = \bigcup_{k=1}^{m}((\{x_k\} \cup (X \setminus \{x_k\})) \sqcup \mathcal{E}_{m,k}) = \{X\} \sqcup (\bigcup_{k=1}^{m} \mathcal{E}_{m,k}) = \{X\} \sqcup \mathcal{H}_m,$$

meaning that $Z(\mathcal{F}_m \dot\bowtie \mathcal{G}_m) = \Omega(2^{m/5}/m)$.

For the meet operation, we let $\mathcal{G}_m := \{Y\}$, where $Z(\mathcal{G}_m) = O(m)$. Similar to join, we have $\mathcal{F}_m \sqcap \mathcal{G}_m = \mathcal{H}_m$, meaning that $Z(\mathcal{F}_m \sqcap \mathcal{G}_m) = \Omega(2^{m/5}/m)$.

For the delta operation, we let $\mathcal{G}_m = 2^X$. Since $\{\{x_k\}\} \boxplus 2^X = 2^X$ for any k, we have

$$\mathcal{F}_m \boxplus \mathcal{G}_m = \bigcup_{k=1}^{m}((\{\{x_k\}\} \boxplus 2^X) \sqcup \mathcal{E}_{m,k}) = 2^X \sqcup \left(\bigcup_{k=1}^{m} \mathcal{E}_{m,k}\right) = 2^X \sqcup \mathcal{H}_m.$$

The ZDD size of $\mathcal{F}_m \boxplus \mathcal{G}_m$ is at least $Z(\mathcal{H}_m) = \Omega(2^{m/5}/m)$. ◄

3.2.2 Quotient and Remainder

For the quotient operation, we constitute a pair of families such that performing an operation incurs the intersection of $O(m)$ subfamilies. Here, let $\mathcal{E}'_{m,k} := 2^Y \setminus \mathcal{E}_{m,k}$ be the complement of $\mathcal{E}_{m,k}$ regarding the family of subsets of Y. By De Morgan's laws, we have $\bigcap_k \mathcal{E}'_{m,k} = 2^Y \setminus (\bigcup_k \mathcal{E}_{m,k}) = 2^Y \setminus \mathcal{H}_m =: \mathcal{H}'_m$. The ZDD size representing $\mathcal{H}'_m$ can be lower bounded by the following lemma.

▶ **Lemma 8.** *Suppose that two families $\mathcal{F}, \mathcal{G}$ of subsets of the same set satisfy $Z(\mathcal{F}) = O(f(m))$, $Z(\mathcal{G}) = \Omega(g(m))$, and $\mathcal{F} \supseteq \mathcal{G}$. Then, $Z(\mathcal{F} \setminus \mathcal{G}) = \Omega(g(m)/f(m))$.*

Proof of Lemma 8. $\mathcal{F} \supseteq \mathcal{G}$ implies $\mathcal{F} \setminus (\mathcal{F} \setminus \mathcal{G}) = \mathcal{G}$. Since the ZDD size after taking the difference can be bounded by the product of the sizes of operand ZDDs, we have $Z(\mathcal{G}) = O(Z(\mathcal{F})Z(\mathcal{F} \setminus \mathcal{G}))$. Suppose $Z(\mathcal{F} \setminus \mathcal{G}) = o(g(m)/f(m))$. Then, $Z(\mathcal{G}) = o(f(m) \cdot (g(m)/f(m))) = o(g(m))$, refuting the assumption $Z(\mathcal{G}) = \Omega(g(m))$. Therefore, $Z(\mathcal{F} \setminus \mathcal{G}) = \Omega(g(m)/f(m))$. ◄

Since $Z(2^Y) = O(m)$ and $Z(\mathcal{H}_m) = \Omega(2^{m/5}/m)$, we have $Z(\mathcal{H}'_m) = \Omega(2^{m/5}/m^2)$.

▶ **Theorem 9.** *Let $\diamond$ be a binary operator chosen from quotient ($/$) and remainder ($\%$). Then, there exists a sequence of families $\mathcal{F}_m$ and $\mathcal{G}_m$ such that* (i) *$\mathcal{F}_m$ and $\mathcal{G}_m$ are families of subsets of a set of $O(m)$ elements,* (ii) *$Z(\mathcal{F}_m) + Z(\mathcal{G}_m) = O(m^3)$, and* (iii) *$Z(\mathcal{F}_m \diamond \mathcal{G}_m) = \Omega(2^{m/5}/\text{poly}(m))$.*

Proof. We again consider the families of subsets of $X \cup Y$, where $X := \{x_1, \ldots, x_m\}$ and $Y := \{y_1, \ldots, y_m\}$. We use the same order of elements: $x_1, \ldots, x_m, y_1, \ldots, y_m$. We define $\mathcal{F}_m$ as

$$\mathcal{F}_m := \bigcup_{k=1}^{m}(\{\{x_k\}\} \sqcup \mathcal{E}'_{m,k}).$$

We have $Z(\mathcal{E}'_{m,k}) = O(m^2)$ as proved in Section 3.4, and thus $Z(\mathcal{F}_m) = O(m^3)$. We also define $\mathcal{G}_m := \{\{x_1\}, \ldots, \{x_m\}\}$, where $Z(\mathcal{G}_m) = O(m)$.

Let us consider $\mathcal{F}_m / \mathcal{G}_m$. By definition, $Y' \in \mathcal{F}_m / \mathcal{G}_m$ if and only if $Y' \subseteq Y$ and $\{x_k\} \cup Y' \in \mathcal{F}_m$ for $k = 1, \ldots, m$. From the definition of $\mathcal{F}_m$, it is equivalent to $Y' \in \bigcap_{k=1}^{m} \mathcal{E}'_{m,k}$. Thus, $\mathcal{F}_m / \mathcal{G}_m = \bigcap_{k=1}^{m} \mathcal{E}'_{m,k} = \mathcal{H}'_m$. This means $Z(\mathcal{F}_m / \mathcal{G}_m) = \Omega(2^{m/5}/m^2)$. The ZDDs involved are depicted in Figure 2.

For the remainder operation, we prepared the same families. Since $\mathcal{G}_m \sqcup (\mathcal{F}_m / \mathcal{G}_m) = \{\{x_1\}, \ldots, \{x_m\}\} \sqcup \mathcal{H}'_m$, $Z(\mathcal{G}_m \sqcup (\mathcal{F}_m / \mathcal{G}_m)) = \Omega(2^{m/5}/m^2)$. Also, since $S \in \mathcal{F}_m / \mathcal{G}_m$ if and only if $S \cup G \in \mathcal{F}_m$ for all $G \in \mathcal{G}_m$, all of the subsets in $\mathcal{G}_m \sqcup (\mathcal{F}_m / \mathcal{G}_m)$ are also contained in $\mathcal{F}_m$. In other words, $\mathcal{F}_m \supseteq \mathcal{G}_m \sqcup (\mathcal{F}_m / \mathcal{G}_m)$. Therefore, by using Lemma 8, $Z(\mathcal{F}_m \% \mathcal{G}_m) = \Omega((2^{m/5}/m^2)/m^3) = \Omega(2^{m/5}/m^5)$. ◄
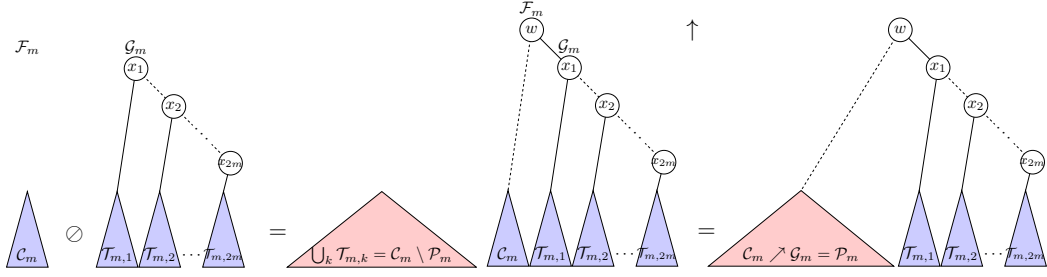
Figure 3 Example of blow-up for permit (left) and maximal (right) operations.

3.2.3 Restrict, Permit, Nonsuperset, and Nonsubset

These operations include inclusion relations of subsets in their definitions, which makes it difficult to generate a hidden weighted bit function as a result of the operation. This is due to the fact that $\mathcal{H}_m$ includes the universal set Y as well as a singleton $\{y_1\}$. For example, if $\mathcal{F}$ is the family of subsets of Y and the universal set Y is included in the result of $\mathcal{F} \oslash \mathcal{G}$, all of the subsets in $\mathcal{F}$ must be included in $\mathcal{F} \oslash \mathcal{G}$ due to the definition of the permit operation.

Instead, we use the permutation function. Because every set in $\mathcal{P}_m$ has cardinality m, the above issue can be alleviated. More specifically, we prepared the complement of the families:

$$\mathcal{C}_m := \{S \subseteq \{y_1, \ldots, y_{m^2}\} \mid |S| = m\}, \quad \mathcal{T}_{m,k} := \mathcal{C}_m \setminus \mathcal{Q}_{m,k} (= \mathcal{C}_m \cap (2^Y \setminus \mathcal{Q}_{m,k})).$$

Here, $\mathcal{C}_m$ is the family of subsets with cardinality m, and thus $\mathcal{T}_{m,k}$ also contains only the subsets with cardinality m. Moreover, by De Morgan's laws,

$$\bigcup_{k=1}^{2m} \mathcal{T}_{m,k} = \mathcal{C}_m \cap \left(\bigcup_{k=1}^{2m} (2^Y \setminus \mathcal{Q}_{m,k}) \right) = \mathcal{C}_m \cap \left(2^Y \setminus \left(\bigcap_{k=1}^{2m} \mathcal{Q}_{m,k} \right) \right) = \mathcal{C}_m \setminus \mathcal{P}_m.$$

We use these families $\mathcal{T}_{m,k}$ to prove the following.

▶ **Theorem 10.** *Let $\diamond$ be a binary operator chosen from restrict ($\triangle$), permit ($\oslash$), nonsuperset ($\searrow$), and nonsubset ($\nearrow$). Then, there exists a sequence of families $\mathcal{F}_m$ and $\mathcal{G}_m$ such that* (i) *$\mathcal{F}_m$ and $\mathcal{G}_m$ are families of subsets of a set of $O(m^2)$ elements,* (ii) *$Z(\mathcal{F}_m) + Z(\mathcal{G}_m) = O(m^4)$, and* (iii) *$Z(\mathcal{F}_m \diamond \mathcal{G}_m) = \Omega(2^m/\mathrm{poly}(m))$.*

Proof. Let us consider the families of subsets of $X \cup Y$, where $X := \{x_1, \ldots, x_{2m}\}$ and $Y := \{y_1, \ldots, y_{m^2}\}$. The order of elements is $x_1, \ldots, x_{2m}$ followed by $y_1, \ldots, y_{m^2}$.

We first consider the permit operation. We define $\mathcal{F}_m := \mathcal{C}_m$ and

$$\mathcal{G}_m := \bigcup_{k=1}^{2m} (\{\{x_k\}\} \sqcup \mathcal{T}_{m,k}).$$

As proved in Section 3.4, $Z(\mathcal{C}_m) = O(m^3)$ and $Z(\mathcal{T}_{m,k}) = O(m^3)$. Thus, $Z(\mathcal{F}_m) = O(m^3)$ and $Z(\mathcal{G}_m) = O(m^4)$. Any set in $\mathcal{F}_m = \mathcal{C}_m$ consists of m elements chosen from $y_1, \ldots, y_{m^2}$, and any set in $\mathcal{G}_m$ consists of m elements from $y_1, \ldots, y_{m^2}$ plus one element from $x_1, \ldots, x_{2m}$. Thus, set $S \in \mathcal{F}_m$ is a subset of some set in $\mathcal{G}_m$ if and only if $\{x_k\} \cup S \in \mathcal{G}_m$ for some k. In other words, $S \in \mathcal{F}_m \oslash \mathcal{G}_m$ if and only if S is included in $\mathcal{T}_{m,k}$ for some k. Since $\mathcal{C}_m \supset \mathcal{T}_{m,k}$ for any k by definition, this means $\mathcal{F}_m \oslash \mathcal{G}_m = \bigcup_{k=1}^{2m} \mathcal{T}_{m,k} = \mathcal{C}_m \setminus \mathcal{P}_m$. Since $Z(\mathcal{C}_m) = O(m^3)$ and $Z(\mathcal{P}_m) = \Omega(2^m/m)$, we have $Z(\mathcal{F}_m \oslash \mathcal{G}_m) = \Omega(2^m/m^4)$ by Lemma 8. The ZDDs involved are depicted in Figure 3.

The nonsubset operation can be treated with the same families. Since $\mathcal{F}_m \nearrow \mathcal{G}_m = \mathcal{F}_m \setminus (\mathcal{F}_m \oslash \mathcal{G}_m)$ by definition, we have $\mathcal{F}_m \nearrow \mathcal{G}_m = \mathcal{C}_m \setminus (\mathcal{C}_m \setminus \mathcal{P}_m) = \mathcal{P}_m$, where the last equality holds due to $\mathcal{C}_m \supset \mathcal{P}_m$. Thus, $Z(\mathcal{F}_m \nearrow \mathcal{G}_m) = \Omega(2^m/m)$.

The restrict and nonsuperset operations can be handled by nearly the same families. We define the same $\mathcal{G}_m$ and let $\mathcal{F}_m := \{X\} \sqcup \mathcal{C}_m$. Similar to the proof of the permit operation, set $X \cup S \in \mathcal{F}_m$ ($S \subseteq Y$) is a superset of some sets in $\mathcal{G}_m$ if and only if $\{x_k\} \cup S \in \mathcal{G}_m$ for some k. This means $\mathcal{F}_m \triangle \mathcal{G}_m = \{X\} \sqcup (\bigcup_{k=1}^{2m} \mathcal{T}_{m,k}) = \{X\} \sqcup (\mathcal{C}_m \setminus \mathcal{P}_m)$, whose ZDD size is $\Omega(2^m/m^4)$. For the nonsuperset operation, we have $\mathcal{F}_m \searrow \mathcal{G}_m = \mathcal{F}_m \setminus (\mathcal{F}_m \triangle \mathcal{G}_m) = \{X\} \sqcup \mathcal{P}_m$, yielding $Z(\mathcal{F}_m \searrow \mathcal{G}_m) = \Omega(2^m/m)$. ◀

3.2.4 Maximal and Minimal

For these operations, we use the close relationship with the nonsuperset and nonsubset operations. We prepare a family having $\mathcal{F}_m$ and $\mathcal{G}_m$ appearing in the proof of Theorem 10 as a subfamily.

▶ **Theorem 11.** *Let $^\diamond$ be a unary operator chosen from maximal ($^\uparrow$) and minimal ($^\downarrow$). Then, there exists a sequence of families $\mathcal{F}_m$ such that* (i) *$\mathcal{F}_m$ is a family of subsets of a set of $O(m^2)$ elements,* (ii) *$Z(\mathcal{F}_m) = O(m^4)$, and* (iii) *$Z(\mathcal{F}_m^\diamond) = \Omega(2^m/\mathrm{poly}(m))$.*

Proof. Let us consider the family of subsets of $\{w\} \cup X \cup Y$, where $X := \{x_1, \ldots, x_{2m}\}$ and $Y := \{y_1, \ldots, y_{m^2}\}$. The order of elements is $w, x_1, \ldots, x_{2m}$ followed by $y_1, \ldots, y_{m^2}$.

We first consider the maximal operation. We define $\mathcal{F}_m$ as

$$\mathcal{F}_m := \mathcal{C}_m \cup [\{\{w\}\} \sqcup \mathcal{G}_m], \quad \text{where } \mathcal{G}_m := \bigcup_{k=1}^{2m} (\{\{x_k\}\} \sqcup \mathcal{T}_{m,k}).$$

Here, we observe that this $\mathcal{G}_m$ is the same as that appearing in the proof of Theorem 10. The ZDD size is bounded as $Z(\mathcal{F}_m) = O(Z(\mathcal{C}_m) + Z(\mathcal{G}_m)) = O(m^4)$. Every set in $\mathcal{C}_m$ has m elements and every set in $\{\{w\}\} \sqcup \mathcal{G}_m$ has $m+2$ elements. Thus, every set in the latter family is maximal, while a set in the former family is maximal if and only if it is not a subset of any set included in the latter family. Therefore, we have

$$\mathcal{F}_m^\uparrow = [\mathcal{C}_m \nearrow (\{\{w\}\} \sqcup \mathcal{G}_m)] \cup [\{\{w\}\} \sqcup \mathcal{G}_m] = [\mathcal{C}_m \nearrow \mathcal{G}_m] \cup [\{\{w\}\} \sqcup \mathcal{G}_m] = \mathcal{P}_m \cup [\{\{w\}\} \sqcup \mathcal{G}_m],$$

where the second equality holds because all of the sets in $\mathcal{C}_m$ do not include w and the last equality follows from the proof of Theorem 10. The resultant ZDD is like the right one in Figure 3, which implies $Z(\mathcal{F}_m^\uparrow) \geq Z(\mathcal{P}_m) = \Omega(2^m/m)$.

The minimal can be treated in a similar way. We define

$$\mathcal{F}_m := \mathcal{G}_m \cup [\{\{w\}\} \sqcup \{\{x_1, \ldots, x_{2m}\}\} \sqcup \mathcal{C}_m],$$

where $\mathcal{G}_m$ is the same family as that above. We again have $Z(\mathcal{F}_m) = O(m^4)$. Every set in $\mathcal{G}_m$ has $m+1$ elements and every set in $\{\{w\}\} \sqcup \{X\} \sqcup \mathcal{C}_m$ has $3m+1$ elements. Thus, every set in the former family is minimal, while a set in the latter family is minimal if and only if it is not a superset of any set included in the former family. Now we have

$$\begin{aligned} \mathcal{F}_m^\downarrow &= \mathcal{G}_m \cup [(\{\{w\}\} \sqcup \{X\} \sqcup \mathcal{C}_m) \searrow \mathcal{G}_m] \\ &= \mathcal{G}_m \cup [\{\{w\}\} \sqcup ((\{X\} \sqcup \mathcal{C}_m) \searrow \mathcal{G}_m)] = \mathcal{G}_m \cup [\{\{w\}\} \sqcup \{X\} \sqcup \mathcal{P}_m], \end{aligned}$$

where the second equality holds because none of the sets in $\mathcal{G}_m$ includes w and the last equality follows from the proof of Theorem 10. This again implies $Z(\mathcal{F}_m^\downarrow) \geq Z(\mathcal{P}_m) = \Omega(2^m/m)$. ◀

3.2.5 Minimal Hitting Set and Closure

For these operations, we can constitute much simpler examples.

▶ **Theorem 12.** *Let $\diamond$ be a unary operator chosen from minimal hitting set ($\sharp$) and closure ($\cap$). Then, there exists a sequence of families $\mathcal{F}_m$ such that* (i) *$\mathcal{F}_m$ is a family of subsets of a set of $O(m^2)$ elements,* (ii) *$Z(\mathcal{F}_m) = O(m^4)$, and* (iii) *$Z(\mathcal{F}_m^\diamond) = \Omega(2^m/\mathrm{poly}(m))$.*

Proof. Let $X := \{x_1, \ldots, x_{2m}\}$ and $Y := \{y_1, \ldots, y_{m^2}\}$. For $k = 1, \ldots, m$, we set $S_k := \{y_{m(k-1)+1}, y_{m(k-1)+2}, \ldots, y_{m(k-1)+m}\}$, and $S_{m+k} := \{y_k, y_{m+k}, \ldots, y_{m(m-1)+k}\}$.

For minimal hitting set operation, we consider a family of subsets of Y. We define $\mathcal{F}_m := \{S_1, \ldots, S_{2m}\}$. Since the ZDD size can be bounded by the sum of cardinality of a set in the family [16], $Z(\mathcal{F}_m) \leq \sum_i |S_i| = O(m^2)$. For $S \subseteq \{y_1, \ldots, y_{m^2}\}$, we associate a binary $m \times m$ matrix, where the (i,j)-element is 1 if and only if $y_{m(i-1)+j} \in S$. Then, $S \cap S_k \neq \emptyset$ means that the k-th row of the matrix has at least one 1 and $S \cap S_{m+k} \neq \emptyset$ means that the k-th column of the matrix has at least one 1. Thus, $S \in \mathcal{F}_m^\sharp$ if and only if the corresponding matrix has at least one 1 for any column or row and no proper subset of S satisfies this property. The minimal matrix having this property is the permutation matrix, and thus $\mathcal{F}_m^\sharp = \mathcal{P}_m$, that is, the permutation function. This implies $Z(\mathcal{F}_m^\sharp) = \Omega(2^m/m)$.

For closure operation, we consider a family of subsets of $X \cup Y$. For $k = 1, \ldots, m$ and $\ell = 1, \ldots, m$, we define $R_{k,\ell} := (X \setminus \{x_k, x_{m+\ell}\}) \cup ((Y \setminus S_k \setminus S_{m+\ell}) \cup \{y_{m(k-1)+\ell}\})$. We define $\mathcal{F}_m := \{R_{k,\ell} \mid k, \ell = 1, \ldots, m\}$. Again, since the ZDD size can be bounded by the sum of cardinality of a set in the family [16], $Z(\mathcal{F}_m) \leq \sum_{k,\ell} |R_{k,\ell}| = O(m^4)$. Then, we show that $\mathcal{F}_m^\cap \cap \mathcal{C}_m = \mathcal{P}_m$, where $\mathcal{P}_m$ is the permutation function. If it is shown, $Z(\mathcal{F}_m^\cap \cap \mathcal{C}_m) = Z(\mathcal{P}_m) = \Omega(2^m/m)$. On the other hand, $Z(\mathcal{F}_m^\cap \cap \mathcal{C}_m) = O(Z(\mathcal{F}_m^\cap)Z(\mathcal{C}_m))$. Since $Z(\mathcal{C}_m) = O(m^3)$, we can deduce that $Z(\mathcal{F}_m^\cap) = \Omega(2^m/\mathrm{poly}(m))$.

We now prove $\mathcal{F}_m^\cap \cap \mathcal{C}_m = \mathcal{P}_m$. First, we show that $\mathcal{F}_m^\cap \cap \mathcal{C}_m \subseteq \mathcal{P}_m$. $R_{k,\ell}$ does not contain any element in S_k and $S_{m+\ell}$ except for $y_{m(k-1)+\ell}$. By fixing k, if $\mathcal{F}' \subseteq \mathcal{F}$ contains at least one $R_{k,\ell}$ for some ℓ, $S = \bigcap_{S' \in \mathcal{F}'} S'$ contains at most one element from S_k. Moreover, S does not contain x_k if and only if $\mathcal{F}'$ contains at least one $R_{k,\ell}$ for some ℓ. Similarly, by fixing ℓ, if $\mathcal{F}'$ contains at least one $R_{k,\ell}$ for some k, which is equivalent to that S does not contain $x_{m+\ell}$, S contains at most one element from $S_{m+\ell}$. Now we can say that when S contains no element in X, S contains at most one element in S_k for any $k = 1, \ldots, 2m$. This means that if S contains no element in X and m elements in Y, $S \in \mathcal{P}_m$. Thus, $\mathcal{F}_m^\cap \cap \mathcal{C}_m \subseteq \mathcal{P}_m$. Next, we show that $\mathcal{F}_m^\cap \cap \mathcal{C}_m \supseteq \mathcal{P}_m$. Let σ be an arbitrary permutation of $1, \ldots, m$. Then, $\{y_{\sigma(1)}, y_{m+\sigma(2)}, \ldots, y_{(m-1)m+\sigma(m)}\} = R_{1,\sigma(1)} \cap R_{2,\sigma(2)} \cap \cdots \cap R_{m,\sigma(m)}$. This means that any set in $\mathcal{P}_m$ is in $\mathcal{F}_m^\cap$. Thus, $\mathcal{F}_m^\cap \cap \mathcal{C}_m \supseteq \mathcal{P}_m$. This concludes $\mathcal{F}_m^\cap \cap \mathcal{C}_m = \mathcal{P}_m$. ◀

3.3 Consideration for Element Order

The above proofs fix the order of elements for each operation. Thus, there is still a possibility that the resultant ZDD size becomes smaller by managing the order of elements. However, it seems that the size of resultant ZDD remains exponential regardless of the order of elements, since every resultant family contains a hidden weighted bit function, a permutation function, or similar families as a subfamily. In the following, we prove that every resultant family has an exponential ZDD size regardless of the order of elements.

▶ **Definition 13.** *Let $\mathcal{F}$ be a family of subsets of set X, and let Y, Y' be the subsets of X satisfying $Y \cap Y' = \emptyset$. We define $\mathcal{F}|_{Y,Y'}$ as the family of subsets of $X \setminus (Y \cup Y')$ such that $S \in \mathcal{F}|_{Y,Y'}$ if and only if $S \cup Y \in \mathcal{F}$.*

In other words, $\mathcal{F}|_{Y,Y'}$ is the family of sets generated from $\mathcal{F}$ by first extracting the sets containing every element of Y, but no element of Y', and then eliminating all of the elements of Y from every set. This operation is called *conditioning* and it is a famous result that this can be performed in polynomial time with BDDs [6]. For the sake of completeness, we show this can also be performed in polynomial time with ZDDs, and then we prove the following.

▶ **Lemma 14.** *Let $\mathcal{F}$ be a family of subsets of a set X of $O(f(m))$ elements. If there exist $Y, Y' \subseteq X$ such that $Z_<(\mathcal{F}|_{Y,Y'}) = \Omega(g(m))$ for any order $<$ of elements, we have $Z_<(\mathcal{F}) = \Omega(g(m)/f(m))$ for any order $<$ of elements.*

If this lemma holds, we can show that the resultant families in Section 3.2 all have an exponential ZDD size regardless of the order of elements. This is because the resultant families in Section 3.2 all have a hidden weighted bit function, a permutation function, or its complements as a subfamily and all of them have an exponential ZDD size regardless of the order of elements; a detailed discussion is given later.

Proof of Lemma 14. If we can show $Z_<(\mathcal{F}|_{Y,Y'}) = O(Z_<(\mathcal{F})f(m))$ for any $Y, Y' \subseteq X$ and any order $<$ of elements, Lemma 14 can be proved as follows: Suppose that there is an order $<$ of elements satisfying $Z_<(\mathcal{F}) = o(g(m)/f(m))$. Then, by the above equation, we have $Z_<(\mathcal{F}|_{Y,Y'}) = o((g(m)/f(m)) \cdot f(m)) = o(g(m))$. This contradicts the assumption that $Z_<(\mathcal{F}|_{Y,Y'}) = \Omega(g(m))$ for any order $<$ of elements.

Next, we fix an arbitrary order $<$ of elements and show $Z_<(\mathcal{F}|_{Y,Y'}) = O(Z_<(\mathcal{F})f(m))$. Here, we consider the operations for constructing a ZDD representing $\mathcal{F}|_{Y,Y'}$ from the ZDD of $\mathcal{F}$. We first extract the sets that contain every element of Y but do not contain any element of Y'. Then, we eliminate all elements of Y.

The former step can be achieved by the intersection operation. Let $\mathcal{G}$ be the family of subsets of X such that $S \in \mathcal{G}$ if and only if S contains all of the elements in Y but does not contain any element in Y'. In other words, $\mathcal{G} := \{S \subseteq X \mid S \cap Y = Y \wedge S \cap Y' = \emptyset\}$. Then, $\mathcal{F} \cap \mathcal{G}$ is the desired family. The ZDD representing $\mathcal{G}$ has the following form: (i) For any $x \in Y$, there is only one ZDD node labeled x whose lo-child is $\bot$ while its hi-child is the next-level node. (ii) For any $x \in Y'$, there is no node labeled x by the reduction rule of ZDD. (iii) for any $x \in X \setminus (Y \cup Y')$, there is only one ZDD node labeled x whose lo-child and hi-child are both the next-level node. Thus, we have $Z_<(\mathcal{G}) = O(f(m))$ because the base set X of $\mathcal{F}$ has $O(f(m))$ elements and, for any element $x \in X$, there is at most one node labeled x. Finally, $Z_<(\mathcal{F} \cap \mathcal{G}) = O(Z_<(\mathcal{F})f(m))$.

The latter can be achieved by eliminating the nodes labeled $x \in Y$ and replacing the branches heading it. For a node labeled $x \in Y$, its lo-child must be $\bot$, since the ZDD is reduced and every set in $\mathcal{F} \cap \mathcal{G}$ must contain x. For this node, we first make all of the arcs heading to it point to its hi-child. Then, we eliminate this node. By performing this operation for every node labeled $x \in Y$, we finally obtain the ZDD of $\mathcal{F}|_{Y,Y'}$. Since this operation does not increase the size of ZDD, we have $Z_<(\mathcal{F}|_{Y,Y'}) = O(Z_<(\mathcal{F})f(m))$.   ◀

Now we can show that the resultant families in the proof of Section 3.2 have exponential ZDD size regardless of the order of elements. For example, for the join operation, $(\{X\} \sqcup \mathcal{H}_m)|_{X,\emptyset} = \mathcal{H}_m$ and $Z(\mathcal{H}_m) = \Omega(2^{m/5}/m)$ for any order of elements of Y (and thus that of $X \cup Y$). Therefore, by Lemma 14, $Z(\mathcal{F}_m \sqcup \mathcal{G}_m) = \Omega(2^{m/5}/m^2)$ for any order of elements of $X \cup Y$. Similar arguments hold for the other operations. We here show that all the resultant families in the proof of Section 3.2 have exponential ZDD size regardless of the order of elements.

Disjoint join $\bowtie$ and joint join $\bowtie$: The resultant family of these operations in the proof of Theorem 7 is $(\{X\} \sqcup \mathcal{H}_m)$. Here, $(\{X\} \sqcup \mathcal{H}_m)|_{X,\emptyset} = \mathcal{H}_m$.

Meet $\sqcap$: In the proof of Theorem 7, we already have $\mathcal{F}_m \sqcap \mathcal{G}_m = \mathcal{H}_m$. Thus, $Z(\mathcal{F}_m \sqcap \mathcal{G}_m) = \Omega(2^{m/5}/m)$ for any order of elements.

Delta $\boxplus$: In the proof of Theorem 7, we have $\mathcal{F}_m \boxplus \mathcal{G}_m = 2^X \sqcup \mathcal{H}_m$. Since $(2^X \sqcup \mathcal{H}_m)|_{X,\emptyset} = \mathcal{H}_m$, $Z(\mathcal{F}_m \boxplus \mathcal{G}_m) = \Omega(2^{m/5}/\mathrm{poly}(m))$ for any order of elements.

Quotient $/$: $Z(2^Y) = O(m)$ and $Z(\mathcal{H}_m) = \Omega(2^{m/5}/m)$ for any order of elements, and $Z(\mathcal{H}'_m) = \Omega(2^{m/5}/m^2)$ for any order of elements by Lemma 8. This also holds for $\mathcal{F}_m / \mathcal{G}_m$ in the proof of Theorem 9 since it equals $\mathcal{H}'_m$.

Remainder $\%$: Since $\mathcal{F}_m = \bigcup_k(\{\{x_k\}\} \sqcup \mathcal{E}'_{m,k})$ and $\mathcal{G}_m \sqcup (\mathcal{F}_m / \mathcal{G}_m) = \{\{x_1\}, \ldots, \{x_m\}\} \sqcup \mathcal{H}'_m$, $\mathcal{F}_m \% \mathcal{G}_m = \bigcup_k(\{\{x_k\}\} \sqcup (\mathcal{E}'_{m,k} \setminus \mathcal{H}'_m))$. Thus, $(\mathcal{F}_m \% \mathcal{G}_m)|_{\{x_1\}, X \setminus \{x_1\}} = \mathcal{E}'_{m,1} \setminus \mathcal{H}'_m$. Here, $Z(\mathcal{E}'_{m,1}) = O(m^2)$ and $Z(\mathcal{H}'_m) = \Omega(2^{m/5}/m^2)$ for any order of elements, and $Z(\mathcal{E}'_{m,1} \setminus \mathcal{H}'_m) = \Omega(2^{m/5}/m^4)$ for any order of elements by Lemma 8. Thus, by Lemma 14, $Z(\mathcal{F}_m \% \mathcal{G}_m) = \Omega(2^{m/5}/m^6)$ for any order of elements because it is a family of subsets of a set with $O(m^2)$ elements.

Permit $\oslash$ and nonsubset $\nearrow$: We already have $\mathcal{F}_m \oslash \mathcal{G}_m = \mathcal{C}_m \setminus \mathcal{P}_m$ and $\mathcal{F}_m \nearrow \mathcal{G}_m = \mathcal{P}_m$ in the proof of Theorem 10. Since $Z(\mathcal{C}_m) = O(m^3)$ and $\mathcal{P}_m = \Omega(2^m/m)$ for any order of elements, $Z(\mathcal{C}_m \setminus \mathcal{P}_m) = \Omega(2^m/m^4)$ for any order of elements by Lemma 8.

Restrict $\triangle$ and nonsuperset $\searrow$: We have $(\{X\} \sqcup (\mathcal{C}_m \setminus \mathcal{P}_m))|_{X,\emptyset} = \mathcal{C}_m \setminus \mathcal{P}_m$ and $(\{X\} \sqcup \mathcal{P}_m)|_{X,\emptyset} = \mathcal{P}_m$; see the proof of Theorem 10.

Maximal $\uparrow$: In the proof of Theorem 11, we have $\mathcal{F}_m^\uparrow|_{\emptyset, \{w\} \cup X} = \mathcal{P}_m$.

Minimal $\downarrow$: In the proof of Theorem 11, we have $\mathcal{F}_m^\downarrow|_{\{w\} \cup X, \emptyset} = \mathcal{P}_m$.

Minimal hitting set $\sharp$: In the proof of Theorem 12, we already have $\mathcal{F}_m^\sharp = \mathcal{P}_m$.

Closure $\cap$: In the proof of Theorem 12, we have $\mathcal{F}_m^\cap \cap \mathcal{C}_m = \mathcal{P}_m$. Since $Z(\mathcal{C}_m) = O(m^3)$ for any order of elements, $Z(\mathcal{F}_m^\cap) = \Omega(2^m/\mathrm{poly}(m))$ for any order of elements.

3.4   Polynomially Bounded ZDDs

We complete the proof of this section by showing that the ZDD sizes of some families appearing in the previous proofs are bounded by a polynomial of m. To prove the size bound, we consider the following *linear network model* to distinguish whether a set is contained in the family $\mathcal{F}$. Note that the idea of a linear network model comes from Knuth's book [13, Theorem M], where it was used to prove the bound of BDD size. Suppose that the order of elements is $x_1 < x_2 < \cdots < x_n$. There are n computational modules $M_1, \ldots, M_n$. Module M_i receives an input of one bit indicating whether x_i is included in the set. Module M_i sends a_{i+1} bits of information to module M_{i+1}. Overall, every module M_i receives an input x_i and a_i bits of information from M_{i-1} and sends a_{i+1} bits of information to M_{i+1}. Since module M_1 has no preceding module, we set $a_1 = 0$. The final module, M_n, outputs one bit indicating whether the set is included in the family $\mathcal{F}$. An overview of the linear network model is drawn in Figure 4. The following lemma suggests that if we can construct a small linear network for the family $\mathcal{F}$, the ZDD size of $\mathcal{F}$ can be bounded.

▶ **Lemma 15.** *For family $\mathcal{F}$ of subsets of $\{x_1, \ldots, x_n\}$, assume that we can construct the linear network model described above to distinguish whether a set is contained in $\mathcal{F}$. Then, the size of ZDD representing $\mathcal{F}$ is bounded by $Z(\mathcal{F}) \leq 2 + \sum_{i=1}^n 2^{a_i}$.*

Proof. For $k = 1, \ldots, n$, we consider the number of distinct subfamilies $\mathcal{F}|_{X,Y}$, where $X \cup Y = \{x_1, \ldots, x_{k-1}\}$. This is because by the node sharing rule, the number of nodes labeled x_k is upper-bounded by the number of possible distinct subfamilies.
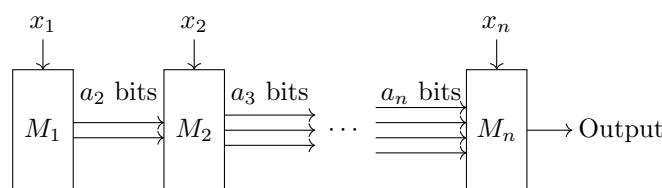
Figure 4 Schematic overview of linear network model.

We observe that the input to module M_i is a_i bits. This means that, regardless of the inclusion of $x_1, \ldots, x_{k-1}$, the subfamily $\mathcal{F}|_{X,Y}$ is completely determined by the information of a_i bits. Therefore, there are at most 2^{a_i} distinct subfamilies, yielding the result that the number of nodes labeled x_k is upper-bounded by 2^{a_i}. Since there are two terminal nodes $\top$ and $\bot$, the overall ZDD size is bounded by $Z(\mathcal{F}) \leq 2 + \sum_{i=1}^{n} 2^{a_i}$. ◀

By Lemma 15, we only have to consider a small linear network for every family.

$\mathcal{E}_{m,k}$ **in Section 3.1:** The family $\mathcal{E}_{m,k}$ is defined as $\{S \subseteq \{y_1, \ldots, y_m\} \mid |S| = k, y_k \in S\}$. In judging whether $S \in \mathcal{E}_{m,k}$ with a linear network, the module M_t is only concerned with the number of elements from $y_1, \ldots, y_t$ in S and whether y_k is in S. The former information can be represented with $\lceil \log(m+1) \rceil$ bits and the latter can be represented with 1 bit. Thus, we can construct a linear network with $a_t = \lceil \log(m+1) \rceil + 1$ bits. By Lemma 15, we have $Z(\mathcal{E}_{m,k}) \leq 2 + m2^{\lceil \log(m+1) \rceil + 1} = O(m^2)$.

$\mathcal{Q}_{m,k}$ **in Section 3.1:** Each of the families $\mathcal{Q}_{m,k}$ $(k = 1, \ldots, 2m)$ is the family of subsets of $\{y_1, \ldots, y_{m^2}\}$ such that there is exactly one element from a set of m selected elements. In constructing a linear network, the module M_t is only concerned with the number of selected elements in S: zero, one, or more than one. This information can be represented with 2 bits. Thus, we have $Z(\mathcal{Q}_{m,k}) \leq 2 + m^2 2^2 = O(m^2)$.

$\mathcal{E}'_{m,k}$ **in Section 3.2.2:** The linear network for $\mathcal{E}'_{m,k} = 2^Y \setminus \mathcal{E}_{m,k}$ can be the same as that for $\mathcal{E}_{m,k}$, except that the output is inverted. Thus, $Z(\mathcal{E}'_{m,k}) = O(m^2)$.

$\mathcal{C}_m$ **in Section 3.2.3:** The family $\mathcal{C}_m$ is defined as $\{S \subseteq \{y_1, \ldots, y_{m^2}\} \mid |S| = m\}$. Similar to the case of $\mathcal{Q}_{m,k}$, every module only retains the number of elements from $y_1, \ldots, y_t$ in S. Moreover, we should only count this number until m; if the count exceeds m, we can immediately determine that S is not in $\mathcal{C}_m$. This count value can be represented with $\lceil \log(m+2) \rceil$ bits. Thus, we have $Z(\mathcal{C}_m) \leq 2 + m^2 2^{\lceil \log(m+2) \rceil} = O(m^3)$.

$\mathcal{T}_{m,k}$ **in Section 3.2.3:** For $\mathcal{T}_{m,k} = \mathcal{C}_m \setminus \mathcal{Q}_{m,k}$, we can construct a linear network by combining the networks for $\mathcal{C}_m$ and $\mathcal{Q}_{m,k}$. We have $\lceil \log(m+2) \rceil$ bits for $\mathcal{C}_m$ and 2 bits for $\mathcal{Q}_{m,k}$. Thus, we have $Z(\mathcal{T}_{m,k}) \leq 2 + m^2 2^{\lceil \log(m+2) \rceil + 2} = O(m^3)$.

We finally note that the ZDD sizes of the above families remain polynomial in m even if the order of elements is different from $y_1 < y_2 < \cdots < y_m < \cdots < y_{m^2}$. Since the cardinality constraint is symmetric, we can reuse the same linear network for different orders of elements. The existence of specific elements can also be treated by changing the input that is watched.

3.5 Discussion

Finally, we give some discussions for the presented results. First, we argue theoretical results for BDDs. As stated in Lemma 2, the sizes of BDD and ZDD differ only by a linear factor of the size of the base item set. All the results in Section 3.2 have the same form that the number of elements is $O(\mathrm{poly}(m))$, the input ZDD sizes are $O(\mathrm{poly}(m))$, and the output ZDD size is exponential in m. Therefore, even if these families are represented by BDDs,

the input BDD sizes are all $O(\text{poly}(m))$, and the output BDD sizes are all exponential in m. Moreover, the output BDD sizes remain exponential in m for any order $<$ of elements since Lemma 2 holds for any order $<$ of elements. This constitutes the theoretical result that the family algebra operations in Table 1, except for the first four operations, cannot be performed in polynomial time in the input BDD sizes.

Second, we discuss how often such exponential blow-up occurs. Although we rely on specific families, the hidden weighted bit function $\mathcal{H}_m$ and the permutation function $\mathcal{P}_m$, the heart of the above proofs is that even a single operation may cause us to compute the union or intersection of multiple subfamilies. Apart from these families, it is usual that taking the union or intersection of multiple families leads to exponential blow-up. To imagine this, we consider encoding a family described by polynomial-sized conjunctive normal form (CNF) into BDD/ZDD. Every clause can be encoded into a polynomial-sized BDD/ZDD. Moreover, if the entire CNF is encoded into BDD/ZDD, we can solve SAT, or even more difficult #SAT, in linear time with respect to the size of BDD/ZDD [13]. However, it is a famous fact that SAT and #SAT are in NP-complete and #P-complete, respectively, meaning that they are believed not to be solved in polynomial time. This means that for many CNFs, the BDD/ZDD after taking intersection of clauses does not remain polynomial-sized. Therefore, apart from the specific examples used in the proof, there are many cases yielding the blow-up of BDD/ZDD size after single family algebra operation.

Finally, we argue the limitation of some of the above results that the permutation function is not such a "devilish" example. The permutation function is a family of subsets of a set with $O(m^2)$ elements and its ZDD size can only be lower bounded by $\Omega(2^m/\text{poly}(m))$. Since the ZDD size of the family of subsets of a set with $O(m^2)$ can be at most $\Omega(2^{m^2}/\text{poly}(m))$, it is far from being the worst-case. We should investigate whether there is a family of sets generated by restrict or similar operations whose ZDD size is lower bounded by $\Omega(\alpha^n/\text{poly}(n))$, where $\alpha > 1$ and n is the number of elements in the base set.

4 Conclusion

We proved that the worst-case complexity of carrying out certain kinds of a family algebra operation on BDDs/ZDDs once is lower bounded by an exponential factor. These include all of the operations raised by Knuth [13, §7.1.4 Ex. 203,204,236,243] except for the basic set operations. In particular, we resolved the controversy over the complexity of the join operation, which had arisen prominently in past literature. We also resolved the open problem regarding the worst-case complexity of the quotient operation.

Future directions include the followings. First, we only prove the lower-bound of the complexity of carrying out a single operation. It should be investigated whether we can obtain a non-trivial upper-bound of the complexity. Second, it is unknown whether a "double recursion" procedure like those in Section 2.2 always leads to an exponential worst-case complexity. It is important to investigate whether there are non-trivial operations that should require a double recursion procedure even though the worst-case complexity is polynomial.

References

1   B. Bollig. A simpler counterexample to a long-standing conjecture on the complexity of Bryant's apply algorithm. *Inf. Process. Lett.*, 114(3):124–129, 2014. `doi:10.1016/j.ipl.2013.11.003`.

2   R. E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Trans. Comput.*, C-35(8):677–691, 1986. `doi:10.1109/TC.1986.1676819`.

3   R. E. Bryant. On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication. *IEEE Trans. Comput.*, 40(2):205–213, 1991. `doi:10.1109/12.73590`.

4   O. Coudert. Solving graph optimization problems with ZBDDs. In *Proc. of the European Design & Test Conference (ED&TC 97)*, pages 224–228, 1997. `doi:10.1109/EDTC.1997.582363`.

5   O. Coudert, J. C. Madre, and H. Fraisse. A new viewpoint on two-level logic minimization. In *Proc. of the 30th ACM/IEEE Design Automation Conference (DAC 1993)*, pages 625–630, 1993. `doi:10.1145/157485.165071`.

6   A. Darwiche and P. Marquis. A knowledge compilation map. *J. Artif. Intell. Res.*, 17(1):229–264, 2002. `doi:10.1613/jair.989`.

7   U. Gupta, P. Kalla, and V. Rao. Boolean Gröbner basis reductions on finite field datapath circuits using the unate cube set algebra. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 38(3):576–588, 2019. `doi:10.1109/TCAD.2018.2818726`.

8   T. Inoue, H. Iwashita, J. Kawahara, and S. Minato. Graphillion: software library for very large sets of labeled graphs. *Int. J. Softw. Tools Technol. Trans.*, 18(1):57–66, 2016. `doi:10.1007/s10009-014-0352-z`.

9   T. Inoue, N. Yasuda, S. Kawano, Y. Takenobu, S. Minato, and Y. Hayashi. Distribution network verification for secure restoration by enumerating all critical failures. *IEEE Trans. Smart Grid*, 6(2):843–852, 2015. `doi:10.1109/TSG.2014.2359114`.

10   A. Ito, R. Ueno, and N. Homma. Efficient formal verification of Galois-Field arithmetic circuits using ZDD representation of Boolean polynomials. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 41(3):794–798, 2022. `doi:10.1109/TCAD.2021.3059924`.

11   J. Kawahara, T. Inoue, H. Iwashita, and S. Minato. Frontier-based search for enumerating all constrained subgraphs with compressed representation. *IEICE Trans. Fundamentals*, E100-A(9):1773–1784, 2017. `doi:10.1587/transfun.E100.A.1773`.

12   J. Kawahara, T. Saitoh, H. Suzuki, and R. Yoshinaka. Solving the longest oneway-ticket problem and enumerating letter graphs by augmenting the two representative approaches with ZDDs. In *Proc. of the Computational Intelligence in Information System (CIIS 2016)*, pages 294–305, 2016. `doi:10.1007/978-3-319-48517-1_26`.

13   D. E. Knuth. *The Art of Computer Programming: Vol. 4A. Combinatorial Algorithms, Part 1*, volume 4A: Combinatorial Algorithms, Part I. Addison-Wesley Professional, 2011.

14   S. Minato. Zero-suppressed BDDs for set manipulation in combinatorial problems. In *Proc. of the 30th ACM/IEEE Design Automation Conference (DAC 1993)*, pages 272–277, 1993. `doi:10.1145/157485.164890`.

15   S. Minato. Calculation of unate cube set algebra using zero-suppressed BDDs. In *Proc. of the 31st ACM/IEEE Design Automation Conference (DAC 1994)*, pages 420–424, 1994. `doi:10.1145/196244.196446`.

16   S. Minato, T. Uno, and H. Arimura. LCM over ZBDDs: Fast generation of very large-scale frequent itemsets using a compact graph-based representation. In *Proc. of Advances in Knowledge Discovery and Data Mining (PAKDD 2008)*, pages 234–246, 2008. `doi:10.1007/978-3-540-68125-0_22`.

17   H. G. Okuno, S. Minato, and H. Isozaki. On the properties of combination set operations. *Inf. Process. Lett.*, 66(4):195–199, 1998. `doi:10.1016/S0020-0190(98)00067-2`.

18   F. Somenzi. CUDD: CU decision diagram package, 1997. URL: `https://github.com/ivmai/cudd`.

19   Y. Takenobu, N. Yasuda, S. Kawano, S. Minato, and Y. Hayashi. Evaluation of annual energy loss reduction based on reconfiguration scheduling. *IEEE Trans. Smart Grid*, 9(3):1986–1996, 2018. `doi:10.1109/TSG.2016.2604922`.

20   R. Yoshinaka, J. Kawahara, S. Denzumi, H. Arimura, and S. Minato. Counterexamples to the long-standing conjecture on the complexity of BDD binary operations. *Inf. Process. Lett.*, 112(16):636–640, 2012. `doi:10.1016/j.ipl.2012.05.007`.