

# Knowledge Compilation Meets Communication Complexity

**Simone Bova**  
 TU Wien  
 Austria

**Florent Capelli**  
 Paris 7, IMJ-PRG  
 France

**Stefan Mengel**  
 CNRS, CRIL UMR 8188  
 France

**Friedrich Slivovsky**  
 TU Wien  
 Austria

## Abstract

Choosing a language for knowledge representation and reasoning involves a trade-off between two competing desiderata: succinctness (the encoding should be small) and tractability (the language should support efficient reasoning algorithms). The area of knowledge compilation is devoted to the systematic study of representation languages along these two dimensions—in particular, it aims to determine the relative succinctness of languages. Showing that one language is more succinct than another typically involves proving a nontrivial lower bound on the encoding size of a carefully chosen function, and the corresponding arguments increase in difficulty with the succinctness of the target language. In this paper, we introduce a general technique for obtaining lower bounds on Decomposable Negation Normal Form (DNNFs), one of the most widely studied and succinct representation languages, by relating the size of DNNFs to multi-partition communication complexity. This allows us to directly translate lower bounds from the communication complexity literature into lower bounds on the size of DNNF representations. We use this approach to prove exponential separations of DNNFs from deterministic DNNFs and of CNF formulas from DNNFs.

## 1 Introduction

Finding suitable representation languages to encode information for reasoning is a basic issue of knowledge representation; the task typically involves striking a balance between competing requirements, for instance expressivity and tractability [Brachman and Levesque, 1984; Levesque and Brachman, 1987]. Since the complexity of reasoning algorithms is measured in terms of the size of the representation, a crucial aspect of this enterprise, and a central research topic in the area of *knowledge compilation* [Marquis, 2015], is the *relative succinctness* of representation languages [Gogic *et al.*, 1995]. For instance, satisfiability of a Boolean function can be checked in linear time given its truth table, but we typically prefer encodings in terms of propositional formulas in spite of the increase in the complexity of satisfiability testing because these representations are exponentially more succinct.

In the propositional case, a systematic comparison of fully expressive, tractable representation languages was carried out by Darwiche and Marquis [2002]. One of the main aims of their work was to determine the relative succinctness of languages and decide whether representations in one language can be translated into another language at the cost of increasing the representation size at most polynomially. Showing unconditionally that such a transformation does not exist typically involves two parts: first, giving an upper bound on the representation size of a carefully chosen function  $f$  in the first language; and second, proving a non-trivial lower bound on the representation size of  $f$  in the second language. The latter part tends to become increasingly difficult with the succinctness of the representation language.

Many of the languages considered in knowledge compilation are sub-classes of the class of circuits in decomposable negation normal form, or DNNFs [Darwiche, 2001]. The limitations of DNNFs are generally not well understood, as witnessed by the lack of general techniques for proving strong lower bounds on the size of DNNF representations.

Indeed, lower bounds on the size of DNNF representations can be proved by lifting lower bounds on nondeterministic read-once branching programs using a quasipolynomial simulation of DNNFs by nondeterministic read-once branching programs [Razgon, 2015; Beame and Liew, 2015], or by leveraging lower bounds from monotone circuit complexity [Bova *et al.*, 2014], but these approaches only lead to “weakly exponential” lower bounds of the form  $\exp(n^{\Omega(1)})$  and do not provide a fine-grained understanding of the complexity of DNNFs. “Strongly exponential” lower bounds of the form  $\exp(\Omega(n))$  have been obtained using a more direct approach, but at the cost of fairly involved combinatorial arguments that are particular to the class of functions against which lower bounds are shown [Ponnuswami and Venkateswaran, 2004; Bova *et al.*, 2014].

In this paper, we introduce a general approach to proving lower bounds for DNNFs by establishing a connection between the DNNF size and the *multi-partition communication complexity* of Boolean functions [Duris *et al.*, 2004]. This connection allows us to translate lower bounds on the communication complexity into lower bounds on the DNNF and deterministic DNNF (d-DNNF) size [Darwiche, 2001b]. Using this technique, we gain new insights into the limits of DNNFs and d-DNNFs. In particular, we prove exponential

separations of DNNFs from d-DNNFs and of CNF formulas (even in prime implicates normal form) from DNNFs.

In a nutshell, the communication complexity of a function  $f : X \times Y \rightarrow Z$  is the number of bits two individuals (usually named Alice and Bob) have to exchange to compute  $f(x, y)$  if one only knows  $x$  and the other only knows  $y$ .<sup>1</sup> To obtain lower bounds on this measure, communication complexity abstracts from the specifics of the communication process and takes a purely combinatorial view, the fundamental notion of which is that of a (*combinatorial*) *rectangle*. Formally, a rectangle in  $X \times Y$  is a subset  $R \subseteq X \times Y$  such that  $R = A \times B$  for some  $A \subseteq X$  and  $B \subseteq Y$ . The communication complexity of a function  $f$  yields an upper bound on the size of a partition of the input space into rectangles on which  $f$  assumes the same value. Accordingly, lower bounds on the number of rectangles in such a partition provide lower bounds on the communication complexity, and researchers have developed powerful tools to prove such bounds.

In turn, lower bounds on the communication complexity (or related measures) of a function can be used to derive lower bounds on the space required to compute this function in various models of computation: one argues that the communication complexity of a function  $f : D \rightarrow Z$  is high with respect to *any* (balanced) partition of  $D$  into  $X$  and  $Y$ , and then shows how an encoding of  $f$  in a particular representation language induces such a partition.

This is a standard technique for obtaining lower bounds on the representation size of (Boolean) functions as OBDDs (ordered binary decision diagrams) [Wegener, 2000]. Pipatsrisawat and Darwiche [2010] used a similar approach to prove lower bounds on the size of structured (deterministic) DNNFs, though without making the connection to communication complexity. Recently, Beame and Liew [2015] showed that the best-partition communication complexity of a function can be related to its sentential decision diagram (SDD) size, a subclass of DNNFs introduced by Darwiche [2011].

The model of communication complexity used in the above cases only considers a single partition of the input space into  $X$  and  $Y$ . To obtain lower bounds for DNNFs, we turn to a more flexible model that allows for multiple partitions [Duris *et al.*, 2004]. We first prove that one needs at most one rectangle for each gate of a DNNF to cover its satisfying assignments. Here, it is useful to think of a DNNF in terms of its *certificates*. A certificate for a satisfying assignment is simply a minimal satisfied sub-DNNF that contains the output gate. Decomposability ensures that the certificates of a DNNF are trees, and certificates sharing a gate can be split and recombined into trees that are again certificates. It follows that the set of satisfying assignments with certificates that contain a particular gate is a rectangle (Theorem 1), and the union of these rectangles covers the satisfying assignments of the DNNF. In a d-DNNF, each satisfying assignment has exactly one certificate and the resulting rectangle cover is even disjoint.

In order to be able to transfer lower bounds from multi-partition communication complexity, we have to refine the above construction and show that the set of satisfying as-

<sup>1</sup>For standard textbooks on the subject of communication complexity, see [Kushilevitz and Nisan, 1997; Hromkovic, 1997].

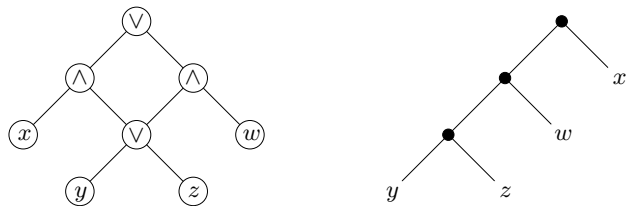


Figure 1: A DNNF (left) and a vtrees (right). The DNNF is structured (it respects the vtrees), but not deterministic.

signments of a DNNF can be covered by a small number of rectangles with respect to *balanced* partitions, that is, partitions where each part contains a constant fraction of the input variables (Theorem 6).

With this connection in place, we can translate lower bounds for the multi-partition communication complexity of a function into lower bounds for its DNNF size. Leveraging a result by Sauerhoff [2003], we get a lower bound of  $\exp(\Omega(n))$  on the d-DNNF size of functions  $S_n$  (Theorem 8) whose DNNF size is polynomial in  $n$  (Proposition 7), thus separating DNNFs from d-DNNFs. Similarly, using a lower bound by Jukna and Schnitger [2002], we obtain a lower bound of  $\exp(\Omega(n^2))$  on the DNNF size of functions  $JS_n$  (Theorem 12) that can be represented by CNF formulas of size  $O(n^2)$  (Proposition 11). This allows us to prove a separation of CNF formulas—in fact, even formulas in prime implicates normal form (PI)—from DNNFs, a result recently established in [Bova *et al.*, 2014].

## 2 Preliminaries

In this section, we introduce the notions of decomposable circuits (from knowledge compilation) and rectangle covers (from communication complexity).

**Decomposable NNFs (DNNFs).** We consider circuits in negation normal form, in short NNFs, which are (Boolean) circuits over fanin 2 conjunction and disjunction gates, labelled with  $\wedge$  and  $\vee$ , whose inputs are labeled by literals.<sup>2</sup> The size of an NNF  $C$ , denoted by  $|C|$ , is the number of its gates.<sup>3</sup>

Let  $X$  be a finite set of variables. An NNF  $C$  over  $X$  is an NNF whose input gates are labelled with literals over variables in  $X$ . The (Boolean) function  $f_C : \{0, 1\}^X \rightarrow \{0, 1\}$  computed by an NNF  $C$  over  $X$  is defined in the usual way. We let  $\text{sat}(C) = \text{sat}(f_C) = f_C^{-1}(1)$  denote the set of satisfying assignments of  $C$  and  $f_C$ . Two NNFs  $C$  and  $C'$  over  $X$  are *equivalent* if  $\text{sat}(C) = \text{sat}(C')$ ; if  $C$  and  $C'$  are equivalent, we write  $C \equiv C'$ . We also write  $C \equiv f$  if  $f_C = f$ .

For a gate  $g$  in an NNF  $C$  over  $X$ , we let  $C_g$  denote the subcircuit of  $C$  rooted at  $g$ . In particular,  $C_g = C$  if  $g$  is the output gate of  $C$ . For an NNF  $C$  over variables  $X$  and a gate  $g \in C$ , we let  $\text{var}(C_g) \subseteq X$  denote the variables appearing at input gates of  $C_g$ .

<sup>2</sup>For technical convenience we admit circuits formed by a single gate labelled with 0 or 1, but assume that circuits with at least two gates do not contain constants.

<sup>3</sup> $|C| > 0$ , as  $C$  contains at least the output gate. Circuits over unbounded fanin conjunctions and disjunctions can be quadratically simulated by fanin 2 circuits.

Let  $g$  be an  $\wedge$ -gate in an NNF  $C$ , and let  $h$  and  $h'$  be two distinct gates wiring  $g$  in  $C$ . Then  $g$  is called *decomposable* if  $\text{var}(C_h) \cap \text{var}(C_{h'}) = \emptyset$ . An NNF whose  $\wedge$ -gates are decomposable is called a *decomposable* NNF (short, *DNNF*).

Let  $g$  be an  $\vee$ -gate in an NNF  $C$ , and let  $h$  and  $h'$  be two distinct gates wiring  $g$  in  $C$ . Then  $g$  is called *deterministic* if  $\text{sat}(C_h) \cap \text{sat}(C_{h'}) = \emptyset$ , viewing each circuit involved in the equation as an NNF over  $\text{var}(C)$ . An NNF whose  $\vee$ -gates are all deterministic is called a *deterministic* NNF.

Let  $Y$  be a finite nonempty set of variables. A *variable tree* (in short, a *vtree*) for the variable set  $Y$  is a rooted, full, ordered, binary tree  $T$  whose leaves correspond bijectively to  $Y$ ; for simplicity, we identify each leaf in  $T$  with the variable in  $Y$  it corresponds to.

For every internal node  $v$  of the vtree  $T$ , we let  $v_l$  and  $v_r$  denote respectively the left and right child of  $v$ . Moreover, we denote by  $T_v$  the subtree of  $T$  rooted at  $v$ . We also let  $Y_v \subseteq Y$  denote (the variables corresponding to) the leaves of  $T_v$ .

Let  $C$  be a DNNF over variables  $X$ , and let  $T$  be a vtree for the variable set  $Y$ . Let  $g$  be an  $\wedge$ -gate in  $C$  having wires from gates  $h$  and  $h'$ . We say that  $g$  *respects* the node  $v$  of  $T$  if  $\text{var}(C_h) \subseteq Y_{v_l}$  and  $\text{var}(C_{h'}) \subseteq Y_{v_r}$ . We say that  $C$  *respects* the vtree  $T$  if each  $\wedge$ -gate in  $C$  respects some node in  $T$ . A DNNF  $C$  is called *structured* if it respects some vtree.

**Rectangles and Covers.** Let  $X$  be a finite set of variables. A partition of  $X$  is a sequence of pairwise disjoint subsets (blocks) of  $X$  whose union is  $X$ . A partition  $(X_1, X_2)$  of  $X$  into two blocks is called *balanced* if  $|X|/3 \leq \min(|X_1|, |X_2|)$ ; clearly, this is equivalent to  $\max(|X_1|, |X_2|) \leq 2|X|/3$ .

Let  $(X_1, X_2)$  be a partition of  $X$ . For  $b_1: X_1 \rightarrow \{0, 1\}$  and  $b_2: X_2 \rightarrow \{0, 1\}$ , we let  $b_1 \cup b_2: X_1 \cup X_2 \rightarrow \{0, 1\}$  denote the assignment whose restriction to  $X_i$  equals  $b_i$  for  $i = 1, 2$ . Also, for  $B_1 \subseteq \{0, 1\}^{X_1}$  and  $B_2 \subseteq \{0, 1\}^{X_2}$ , we let  $B_1 \times B_2 = \{b_1 \cup b_2 \mid b_1 \in B_1, b_2 \in B_2\}$ .

A (*combinatorial*) *rectangle* over  $X$  is a function  $r: \{0, 1\}^X \rightarrow \{0, 1\}$  such that there exist an *underlying* partition  $(X_1, X_2)$  of  $X$  and functions  $r^i: \{0, 1\}^{X_i} \rightarrow \{0, 1\}$  for  $i = 1, 2$  such that  $\text{sat}(r) = \text{sat}(r^1) \times \text{sat}(r^2)$ . A rectangle is called *balanced* if its underlying partition is balanced.

We also call a subset  $R$  of  $\{0, 1\}^X$  a *rectangle* over  $X$ , with underlying partition  $(X_1, X_2)$ , if there exists a rectangle  $r: \{0, 1\}^X \rightarrow \{0, 1\}$ , with underlying partition  $(X_1, X_2)$ , such that  $R = \text{sat}(r)$ .

Let  $f: \{0, 1\}^X \rightarrow \{0, 1\}$  be a function. A finite set  $\{r_i\}$  of rectangles over  $X$  is called a *rectangle cover* of  $f$  if

$$\text{sat}(f) = \bigcup_i \text{sat}(r_i); \quad (1)$$

the rectangle cover is called *disjoint* if the union in (1) is disjoint. A rectangle cover is called *balanced* if each rectangle in the cover is balanced.

Note that, if  $f$  has a rectangle cover as in (1), then  $f \equiv \bigvee_i (C_i^1 \wedge C_i^2)$ , where  $C_i^1$  (respectively,  $C_i^2$ ) is an NNF over the first (respectively, second) block of the partition underlying the  $i$ th rectangle in the cover; the outermost  $\vee$ -gate is deterministic if the cover is disjoint, and the  $i$ th  $\wedge$ -gate displayed is decomposable (by the partition of the  $i$ th rectangle).

### 3 Knowledge Compilation Meets Communication Complexity

In this section we show how to construct, given a (deterministic) DNNF  $C$ , a (disjoint) rectangle cover of size at most  $|C|$  for  $f_C$  (Theorem 6), thus establishing a fundamental connection between knowledge compilation and communication complexity.

The construction is based on basic but crucial combinatorial properties of (deterministic) DNNFs, notably on the notion of certificate for a DNNF, and an operation eliminating gates in DNNFs, which we now define and study.

Let  $C$  be a DNNF on variables  $X$ . A *certificate* of  $C$  is a DNNF  $T$  on variables  $X$  such that:  $T$  contains the output gate of  $C$ ; if  $T$  contains an  $\wedge$ -gate  $v$  of  $C$ , then  $T$  also contains every gate of  $C$  having an output wire to  $v$ ; if  $T$  contains an  $\vee$ -gate  $v$  of  $C$ , then  $T$  also contains exactly one gate of  $C$  having an output wire to  $v$ . The output gate of  $T$  coincides with the output gate of  $C$ , and the gates of  $T$  inherit their labels and wires from  $C$ .

We let  $\text{cert}(C)$  denote the set of certificates of  $C$ . It is readily verified that

$$\text{sat}(C) = \bigcup_{T \in \text{cert}(C)} \text{sat}(T). \quad (2)$$

The following fact is an easy consequence of decomposability (and constant freeness).

**Fact 1.** *Let  $C$  be a DNNF and let  $T \in \text{cert}(C)$ . The graph underlying  $T$  is a binary tree. Moreover, no two leaves of  $T$  are labeled by the same variable.*

For a gate  $g$  of a DNNF  $C$ , we let  $\text{cert}(C, g)$  denote the set of certificates of  $C$  containing the gate  $g$  and let

$$\text{sat}(C, g) = \bigcup_{T \in \text{cert}(C, g)} \text{sat}(T); \quad (3)$$

in words,  $\text{sat}(C, g)$  contains those satisfying assignments of  $C$  that satisfy the subcircuit rooted at gate  $g$ .

The crucial combinatorial property of DNNFs is that  $\text{sat}(C, g)$  is a rectangle separating the variables in the subcircuit of  $C$  rooted at  $g$ . Formally,

**Theorem 1.** *Let  $C$  be a DNNF on variables  $X$  and let  $g$  be a gate of  $C$ . Then  $\text{sat}(C, g)$  is a rectangle over  $X$  with underlying partition  $(\text{var}(C_g), X \setminus \text{var}(C_g))$ .*

In view of proving Theorem 1, we prepare the following.

**Lemma 2.** *Let  $g$  be a gate of a DNNF  $C$  and let  $T \in \text{cert}(C, g)$ . Then*

$$\text{var}(T) \setminus \text{var}(T_g) \subseteq \text{var}(C) \setminus \text{var}(C_g).$$

*Proof of Lemma 2.* Otherwise, let  $x$  be a variable contained in both  $\text{var}(T) \setminus \text{var}(T_g)$  and  $\text{var}(C_g)$ . Then there exists a certificate  $T' \in \text{cert}(C, g)$  such that  $x \in \text{var}(T'_g)$ . By Fact 1,  $T$  and  $T'$  are trees. By replacing  $T_g$  in  $T$  by  $T'_g$ , we obtain a certificate  $T$  of  $C$  where  $x$  occurs twice, contradicting Fact 1.  $\square$

*Proof of Theorem 1.* Let  $Y = \text{var}(C_g)$  and  $Y' = X \setminus Y$ . Let  $b$  and  $b'$  be in  $\text{sat}(C, g)$ . It is sufficient to show that  $b|_Y \cup b'|_{Y'}$  is in  $\text{sat}(C, g)$ , where  $b|_Y$  denotes the restriction of  $b$  to  $Y$  and  $b'|_{Y'}$  denotes the restriction of  $b'$  to  $Y'$ .

By (3), there exist certificates  $T$  and  $T'$  in  $\text{cert}(C, g)$  such that  $b \in \text{sat}(T)$  and  $b' \in \text{sat}(T')$ . Then  $b$  satisfies all literals in  $T_g$ . Since  $\text{var}(T_g) \subseteq Y$ , it follows that  $b|_Y$  satisfies all literals in  $T_g$ . Similarly,  $b'$  satisfies all literals in  $T' \setminus T'_g$ ; and, by Lemma 2, it holds that  $\text{var}(T') \setminus \text{var}(T'_g) \subseteq Y'$ . Hence  $b'|_{Y'}$  satisfies all literals in  $T' \setminus T'_g$ .

By Fact 1,  $T$  and  $T'$  are trees. By replacing  $T'_g$  in  $T'$  by  $T_g$ , we obtain a certificate  $S$  of  $C$  containing  $g$ , that is,  $S \in \text{cert}(C, g)$ . By the above observations,  $b|_Y \cup b'|_{Y'}$  satisfies all literals in  $S$ , that is,  $b|_Y \cup b'|_{Y'}$  is in  $\text{sat}(S)$ . It follows by (3) that  $b|_Y \cup b'|_{Y'}$  is in  $\text{sat}(C, g)$ .  $\square$

As  $\text{cert}(C) = \bigcup_{g \in C} \text{cert}(C, g)$ , it trivially follows by (2) and (3) that

$$\text{sat}(C) = \bigcup_{g \in C} \text{sat}(C, g). \quad (4)$$

In words,  $C$  is covered by the rectangles induced by its gates (recall Theorem 1). However, in view of reusing known lower bounds on the size of rectangle covers (see Section 4), we need to find a subset of gates of  $C$  generating a *balanced* rectangle cover for  $C$ .

To this aim, we first introduce and study an operation on DNNFs that boils down to relabelling a noninput gate by 0 and propagating the information in the circuit.

Let  $C$  be a DNNF on variables  $X$ . We define an operation (*0-propagation*) that, given a DNNF  $C$  with some gates labelled with 0, returns either a single gate labelled with a 0 or a DNNF where no gates are labelled with 0. The operation iterates the following step until all 0s disappear (or the DNNF reduces to a single gate labelled 0). Let  $g$  be a gate in  $C$  labelled with 0. Then: delete all input wires of  $g$ ; delete all output wires of  $g$  to  $\vee$ -gates; relabel all  $\wedge$ -gates wired by  $g$  and all fanin 0  $\vee$ -gates by 0; delete all gates with no directed paths to the output gate.

Now we define the DNNF on variables  $X$  obtained by *eliminating the noninput gate  $g$  in  $C$* , denoted by  $C - g$ , as the result of relabelling  $g$  by 0 and performing 0-propagation.

The impact of passing from  $C$  to  $C - g$  is dropping all certificates containing  $g$  in (2), as formalized by the following proposition (whose proof is omitted for space limitations).

**Proposition 3.** *Let  $C$  be a DNNF and let  $g$  be a noninput gate of  $C$ . Then*

$$\text{sat}(C - g) = \bigcup_{T \in \text{cert}(C) \setminus \text{cert}(C, g)} \text{sat}(T).$$

The following lemma states a property of gate elimination crucial to our construction: in passing from  $C$  to  $C - g$  we *only* forget satisfying assignments in the rectangle  $\text{sat}(C, g)$ .

**Lemma 4.** *Let  $C$  be a DNNF and let  $g$  be a noninput gate of  $C$ . Then  $C - g$  is a DNNF and*

$$\text{sat}(C) \setminus \text{sat}(C, g) \subseteq \text{sat}(C - g) \subseteq \text{sat}(C).$$

*Proof.* Note that gate elimination preserves decomposability. The inclusions follow directly from Proposition 3, recalling (2) and (3).  $\square$

In general, an assignment can satisfy more than one certificate. In this case, the left inclusion in Lemma 4 is strict. For instance, let  $D$  be a DNNF and let  $C = D \vee D$ . Let  $g$  be the output gate of one copy of  $D$  in  $C$ . Then  $\text{sat}(C) = \text{sat}(D)$  and  $\text{sat}(C, g) = \text{sat}(D)$ , so that  $\text{sat}(C) \setminus \text{sat}(C, g) = \emptyset$ , but  $\text{sat}(C - g) = \text{sat}(D)$ .

By contrast, the left inclusion in Lemma 4 becomes an equality in the deterministic case; in other words, eliminating a gate  $g$  in a deterministic DNNF  $C$  removes *exactly* the assignments in the rectangle  $\text{sat}(C, g)$ . Formally,

**Lemma 5.** *Let  $C$  be a deterministic DNNF and let  $g$  be a noninput gate of  $C$ . Then  $C - g$  is a deterministic DNNF and*

$$\text{sat}(C) \setminus \text{sat}(C, g) = \text{sat}(C - g).$$

*Proof.* We show that gate elimination preserves determinism. Assume that  $C - g$  contains a nondeterministic  $\vee$ -gate  $h$ , wired by gates  $k$  and  $k'$  such that  $b \in \text{sat}((C - g)_k)$  and  $b \in \text{sat}((C - g)_{k'})$ . It follows by Proposition 3 that there exist certificates  $T$  and  $T'$  in  $\text{cert}(C) \setminus \text{cert}(C, g)$  such that  $b \in \text{sat}(T_k)$  and  $b \in \text{sat}(T'_{k'})$ . Then  $b \in \text{sat}(C_k)$  and  $b \in \text{sat}(C_{k'})$ , that is,  $h$  is nondeterministic in  $C$ , a contradiction.

For the equality, by Lemma 4 it suffices to prove that  $\text{sat}(C - g)$  is contained in  $\text{sat}(C) \setminus \text{sat}(C, g)$ . Assume  $b \in \text{sat}(C - g)$  so that, by Proposition 3, it holds that  $b \in \text{sat}(T')$  for some  $T' \in \text{cert}(C) \setminus \text{cert}(C, g)$ . In particular,  $b \in \text{sat}(C)$ . It suffices to show that  $b \notin \text{sat}(C, g)$ .

Otherwise, by (3),  $b \in \text{sat}(T)$  for some  $T \in \text{cert}(C, g)$ . Since  $T' \notin \text{cert}(C, g)$ , we have  $T \neq T'$ . It follows that there exist two distinct gates  $k$  and  $k'$  in  $C$ , wiring an  $\vee$ -gate  $h$  in  $C$ , such that  $T$  contains  $k$  and  $T'$  contains  $k'$ . Then  $b \in \text{sat}(T_k)$  and  $b \in \text{sat}(T'_{k'})$ , so that  $b \in \text{sat}(C_k)$  and  $b \in \text{sat}(C_{k'})$ . Again,  $h$  is nondeterministic in  $C$ , a contradiction.  $\square$

It follows from Lemma 4 that the process of iteratively eliminating gates in a DNNF (until it becomes unsatisfiable) yields a rectangle cover; moreover, by Lemma 5, the rectangle cover is disjoint if the DNNF is deterministic.

We strengthen the above remark by proving that a suitable elimination sequence in a (deterministic) DNNF  $C$  yields not just a (disjoint) rectangle cover of  $C$ , but indeed a *balanced* one, a crucial feature for the intended application (Section 4).

**Theorem 6.** *Let  $C$  be a (deterministic) DNNF computing a function  $f$ . Then  $f$  has a balanced (disjoint) rectangle cover of size at most  $|C|$ .*

*Proof.* Let  $C = C^0$  be a (deterministic) DNNF over variables  $X = \text{var}(C)$  computing  $f$ . For  $i = 0, 1, \dots$ , we find a suitable gate  $g_i \in C^i$  and construct the (deterministic) DNNF  $C^{i+1} = C^i - g_i$  by eliminating  $g_i$  in  $C^i$ , until we hit  $l \leq |C|$  such that  $C^l \equiv 0$ . Along the way, we construct a set

$$\{R_i \mid i = 0, \dots, l - 1\} \quad (5)$$

that, we claim, is the desired rectangle cover of  $f$ .

For  $i = 0, 1, \dots$ , we choose the gate  $g_i$  as follows. We distinguish two cases. If  $2|X|/3 < |\text{var}(C^i)|$  then, by a descent from the output gate of  $C^i$ , we find a gate  $g_i \in C^i$

	NNF	DNNF	DNF	CNF	PI	IP
DNNF	$\not\leq^{**}$	$\leq$	$\leq$	$\not\leq^{**}$	$\not\leq^{**}$	$\leq$
d-DNNF	$\not\leq^{**}$	$\not\leq^*$	?	$\not\leq^{**}$	$\not\leq^{**}$	?

Table 1: Relative succinctness of knowledge compilation languages, taking into account the results of Section 4.1 (\*) and Section 4.2 (\*\*).

such that  $|X|/3 \leq |\text{var}(C_{g_i}^i)| \leq 2|X|/3$ . By Theorem 1 we have that  $\text{sat}(C^i, g_i)$  is a rectangle over  $X$  with underlying partition  $(\text{var}(C_{g_i}^i), X \setminus \text{var}(C_{g_i}^i))$ . Then  $R_i = \text{sat}(C^i, g_i)$  is a balanced rectangle over  $X$ .

If  $|\text{var}(C^i)| < 2|X|/3$  then we let  $g_i$  be the root of  $C^i$ . We obtain the desired rectangle as follows. Let  $\text{var}(C^i) \subseteq X' \subseteq X$  be obtained by adding to  $\text{var}(C^i)$  enough variables from  $X \setminus \text{var}(C^i)$  so that  $(X', X \setminus X')$  is a balanced partition of  $X$ . We put

$$R_i = (\text{sat}(C^i) \times \{0, 1\}^{X' \setminus \text{var}(C^i)}) \times \{0, 1\}^{X \setminus X'}$$

where we view  $C^i$  as a DNNF over  $\text{var}(C^i)$ . Then  $R_i$  is trivially a balanced rectangle over  $X$ .

It follows from the above construction and Lemma 4 that the set in (5) is a balanced rectangle cover of  $C$ . Moreover, if  $C$  is deterministic, then such rectangle cover is disjoint, because  $\text{sat}(C^i, g_i) \cap \text{sat}(C^{i+1}) = \emptyset$  by Lemma 5.  $\square$

## 4 Separating Knowledge Representation Languages via Communication Complexity

In this section, we combine the connection between (deterministic) DNNFs and (disjoint) rectangle covers established in Section 3 with deep combinatorial lower bounds on the size of (disjoint) rectangle covers from the communication complexity literature to obtain exponential separations of DNNFs from deterministic DNNFs (Section 4.1) and of prime implicates (PI) from DNNFs (Section 4.2).

As illustrated in Table 1, these results allow us to answer several questions regarding the relative succinctness of languages left open in the “knowledge compilation map” (cf. Table 3 of [Darwiche and Marquis, 2002]).

### 4.1 DNNFs Versus Deterministic DNNFs

We first prove an exponential separation of DNNFs from deterministic DNNFs. The two classes are separated by a function introduced and studied by Sauerhoff [2003].

Let  $g_n: \{0, 1\}^n \rightarrow \{0, 1\}$  be the function evaluating to 1 if and only if the sum of its inputs is divisible by 3. The *Sauerhoff function*  $S_n: \{0, 1\}^{n^2} \rightarrow \{0, 1\}$  is defined on the  $n \times n$  matrix  $X = (x_{ij})_{1 \leq i, j \leq n}$  of variables by

$$S_n(X) = R_n(X) \vee C_n(X) \quad (6)$$

where  $R_n, C_n: \{0, 1\}^{n^2} \rightarrow \{0, 1\}$  are defined by

$$R_n(X) = \bigoplus_{i=1}^n g_n(x_{i1}, x_{i2}, \dots, x_{in})$$

and  $C_n(X) = R_n(X^\top)$ , where  $X^\top$  denotes the transpose of  $X$ , and  $\oplus$  denotes addition modulo 2.

The Sauerhoff function has polynomial DNNF size.

**Proposition 7.**  $S_n$  in (6) has DNNF size  $O(n^2)$ .

*Proof (Sketch).* The functions  $R_n$  and  $C_n$  have OBDDs of size  $O(n^2)$ , ordering the variables by rows and columns, respectively; their disjunction has size  $O(n^2)$ .  $\square$

We use a highly nontrivial exponential lower bound on the size of balanced disjoint rectangle covers for  $S_n$  [Sauerhoff, 2003, Theorem 4.10].

**Theorem 8** (Sauerhoff). *Any balanced disjoint rectangle cover of the Sauerhoff function  $S_n$  in (6) has size  $2^{\Omega(n)}$ .*

We remark that Sauerhoff actually proves the above lower bound only for rectangles whose underlying partitions have blocks of the same size  $\pm 1$ , but a careful inspection of the proof reveals that the same argument can be lifted to our more relaxed notion of balancedness.

By putting together Theorem 6 and Theorem 8, we get the following lower bound, which, in combination with Proposition 7, yields an explicit, unconditional, exponential separation of DNNFs from deterministic DNNFs:

**Theorem 9.**  $S_n$  in (6) has deterministic DNNF size  $2^{\Omega(n)}$ .

### 4.2 Prime Implicates Versus DNNFs

Next, we show a (strongly) exponential separation of prime implicates (PIs) from DNNFs. In recent (unpublished) work [Bova *et al.*, 2014], we established this separation by means of an involved combinatorial proof; here, we obtain the same result by leveraging a lower bound on the multi-partition communication complexity of a function studied by Jukna and Schnitger [2002], which is defined as follows.

For  $n \geq 2$ , let  $K_n$  be the set of all 2-element subsets (edges) of  $\{1, \dots, n\}$ . We view every subset of  $K_n$  as the edge set of a graph  $G$  whose vertex set is  $\{1, \dots, n\}$ . We identify edges in  $K_n$  with Boolean variables, so that the graph  $G \subseteq K_n$  is encoded by the  $\{0, 1\}$ -assignment of  $K_n$  mapping a variable (edge) to 1 if and only if it is in the edge set of  $G$ .

A triangle  $T$  on  $n$  vertices is a graph with vertices  $\{1, \dots, n\}$  and edges  $\{\{i, j\}, \{i, k\}, \{j, k\}\}$ , where  $|\{i, j, k\}| = 3$ ; it corresponds to the assignment of  $K_n$  mapping  $\{i, j\}, \{i, k\}, \{j, k\}$  to 1 and the other edges to 0.

We let  $\mathcal{T}_n$  be the set of all triangles on  $n$  vertices. For a set  $A \subseteq \mathcal{T}_n$ , we let

$$JS_n^A: \{0, 1\}^{K_n} \rightarrow \{0, 1\} \quad (7)$$

denote the function accepting exactly those graphs over  $\{1, \dots, n\}$  that avoid all triangles in  $A$  (the edge set of no triangle in  $A$  is contained in the edge set of the input graph).

Jukna and Schnitger [2002, Theorem 3.1] show an exponential lower bound on the size of balanced rectangle covers for functions as in (7).

**Theorem 10** (Jukna and Schnitger). *For every  $n$  there exists  $A_n \subseteq \mathcal{T}_n$  of size  $O(n^2)$  such that any balanced rectangle cover of  $JS_n^{A_n}$  in (7) has size  $2^{\Omega(n^2)}$ .*

The *Jukna-Schnitger function*  $JS_n: \{0, 1\}^{K_n} \rightarrow \{0, 1\}$  is defined by

$$JS_n = JS_n^{A_n} \quad (8)$$

where  $A_n$  is chosen by Theorem 10 ( $n \geq 2$ ).

It is readily verified that the Jukna-Schnitger function has polynomial PI size. Recall that a CNF  $F$  is in *prime implicate (PI)* form if every clause entailed by  $F$  is already entailed by a clause of  $F$ , and no clause of  $F$  entails another clause of  $F$ .

**Proposition 11.**  $JS_n$  in (8) has PI size  $O(n^2)$ .

*Proof (Sketch).* Let  $JS_n = JS_n^{A_n}$ . Take the CNF  $F_n$  stating that every triangle in  $A_n$  has an edge that is not in the input graph; it computes  $JS_n$  and it is in PI. Also,  $F_n$  has size  $O(n^2)$ , since  $|A_n| = O(n^2)$  by Theorem 10.  $\square$

By combining Theorem 6 and Theorem 10, we obtain the following lower bound.

**Theorem 12.**  $JS_n$  in (8) has DNNF size  $2^{\Omega(n^2)}$ .

Jointly, Proposition 11 and Theorem 12 yield an unconditional, strongly exponential separation of PIs from DNNFs. As already observed in [Bova *et al.*, 2014], since  $\text{PI} \subseteq \text{CNF} \subseteq \text{NNF}$  and  $\text{d-DNNF} \subseteq \text{DNNF}$ , the remaining separations in Table 1 marked with  $\dagger$  follow from this result.

## 5 Structured Knowledge Representation Languages and Communication Complexity

The lower bound techniques for structured DNNFs introduced by Pipatsrisawat and Darwiche [2010] have a natural interpretation in terms of communication complexity. Their main result can be paraphrased thus:

**Theorem 13** (Pipatsrisawat and Darwiche). *Let  $D$  be a (deterministic) structured DNNF on variables  $X$  computing a function  $f$  and respecting a vtree  $T$ . For every node  $v \in T$ ,  $f$  has a (disjoint) rectangle cover of size at most  $|D|$  where each rectangle has underlying partition  $(X_v, X \setminus X_v)$ .*

*Proof (Sketch).* Let  $v$  be a node in  $T$ . We can find a gate  $g$  of  $D$  such that, for every certificate  $C$  of  $D$  containing  $g$ , it holds that  $\text{var}(C_g) \subseteq X_v$  and  $\text{var}(C \setminus C_g) \subseteq X \setminus X_v$ . We can show as in Theorem 1 that  $\text{sat}(D, g)$  is thus a rectangle with underlying partition  $(X_v, X \setminus X_v)$ . We then apply a similar elimination process as in the proof of Theorem 6.  $\square$

In contrast to Theorem 6, the above statement speaks about rectangle covers whose rectangles share the same underlying partition. Such covers are closely related to a measure known as the best-partition communication complexity [Lipton and Sedgewick, 1981], and Theorem 13 allows us to transfer lower bounds on the best-partition communication complexity and prove a conjecture by Pipatsrisawat and Darwiche [2010].

Let  $X_n = \{x_1, \dots, x_n\}$  and  $Y_n = \{y_1, \dots, y_n\}$ . Let  $T$  be a vtree for  $X_n \cup Y_n$  where the subtree rooted at the left (respectively, right) child of the root is a right-linear vtree for  $X_n$  (respectively,  $Y_n$ ). Pipatsrisawat and Darwiche [2010] conjecture that any deterministic DNNF computing

$$f_n = (x_1 \wedge y_1) \vee \dots \vee (x_n \wedge y_n) \quad (9)$$

and respecting  $T$  has size exponential in  $n$ . We appeal to a nice piece of communication complexity theory to prove the following statement (thus confirming the conjecture).

**Proposition 14.** *Let  $T$  be any vtree for  $X_n \cup Y_n$  containing a vtree for  $X_n$  as a subtree. Then any deterministic DNNF computing  $f_n$  in (9) and respecting  $T$  has size at least  $2^n - 1$ .*

Let  $f: \{0, 1\}^Z \rightarrow \{0, 1\}$  be a function, and let  $(Z_1, Z_2)$  be a partition of  $X$  where  $|Z_1| = |Z_2| = n$ . The *communication matrix* of  $f$  relative to  $(Z_1, Z_2)$ , denoted by  $M(f, Z_1, Z_2)$  is a (Boolean) matrix whose rows and columns are indexed by assignments of  $Z_1$  and  $Z_2$ , respectively, and whose  $(b_1, b_2)$ th entry equals  $f(b_1 \cup b_2)$ .<sup>4</sup>

A basic fact in communication complexity is that the rank of the communication matrix is a lower bound on the size of disjoint rectangle covers of a function [Jukna, 2012, Section 4.1].

**Theorem 15.** *Let  $(Z_1, Z_2)$  be a partition of the variables of a function  $f$ , where  $|Z_1| = |Z_2| = n$ . Then every disjoint rectangle cover of  $f$  into rectangles with underlying partition  $(Z_1, Z_2)$  has size at least  $\text{rank}(M(f, Z_1, Z_2))$ .*

The complement of the function  $f_n$  in (9), called the *disjointness function*,

$$d_n = \neg f_n = (\neg x_1 \vee \neg y_1) \wedge \dots \wedge (\neg x_n \vee \neg y_n), \quad (10)$$

is a well studied object in communication complexity; we denote by  $D_n = M(d_n, X_n, Y_n)$  its communication matrix. The following fact is folklore [Jukna, 2012, Exercise 7.1].

**Proposition 16.**  $\text{rank}(D_n) = 2^n$ .

*Proof of Proposition 14.* Let  $C$  be any deterministic DNNF computing  $f_n$  in (9) and respecting a vtree  $T$  as in the hypothesis. By Theorem 13,  $f_n$  has a disjoint rectangle cover of size at most  $|C|$  where each rectangle has underlying partition  $(X_n, Y_n)$ . Let  $E_n = M(f_n, X_n, Y_n)$ . By Theorem 15,  $|C| \geq \text{rank}(E_n)$ . Since  $D_n = 1 - E_n$  by (10), we have  $2^n = \text{rank}(D_n) = \text{rank}(1 - E_n) \leq \text{rank}(1) + \text{rank}(E_n) = 1 + \text{rank}(E_n)$  by Proposition 16 and basic linear algebra, hence  $\text{rank}(E_n) \geq 2^n - 1$ . We conclude that  $|C| \geq 2^n - 1$ .  $\square$

We conclude by noting that the same general strategy used for obtaining the lower bounds in Section 4 works for structured DNNFs. For instance, the exponential lower bound on the structured DNNF size of the circular bit shift (CBS) function [Pipatsrisawat, 2010], follows directly by Theorem 13 and known exponential lower bounds on the size of rectangle covers for CBS in the best-partition model [Kushilevitz and Nisan, 1997, Chapter 7.2].

## 6 Conclusion

We established a connection between the DNNF size and the multi-partition communication complexity of Boolean functions. This connection allowed us to translate lower bounds from communication complexity into lower bounds on the (deterministic) DNNF size and prove exponential separations of DNNFs from d-DNNFs and of PIs from DNNFs.

We are confident that the applicability of our approach goes beyond the specific lower bound results proved here. In particular, we hope that it can help resolve a few questions from the “knowledge compilation map” that remain open [Darwiche and Marquis, 2002].

<sup>4</sup>We regard communication matrices as matrices over the reals.

## Acknowledgments

The first author was supported by the FWF Austrian Science Fund (P26200). The second author was supported by the French Agence Nationale de la Recherche, AGGREG project reference ANR-14-CE25-0017-01. The third author was supported by the ANR Blanc International ALCOCLAN. The fourth author was supported by the FWF Austrian Science Fund (P27721).

## References

- [Beame and Liew, 2015] P. Beame and V. Liew. New Limits for Knowledge Compilation and Applications to Exact Model Counting. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 131–140, 2015.
- [Bova et al., 2014] S. Bova, F. Capelli, S. Mengel, and F. Slivovsky. A Strongly Exponential Separation of DNNFs from CNF Formulas. *CoRR*, abs/1411.1995, 2014.
- [Brachman and Levesque, 1984] R.J. Brachman and H.J. Levesque. The Tractability of Subsumption in Frame-Based Description Languages. In *Proceedings of the 4th AAAI Conference on Artificial Intelligence (AAAI 1984)*, pages 34–37, 2015.
- [Darwiche, 2001] A. Darwiche. Decomposable Negation Normal Form. *J. ACM*, 48(4):608–647, 2001.
- [Darwiche, 2001b] A. Darwiche. On the Tractable Counting of Theory Models and its Application to Truth Maintenance and Belief Revision. *J. Appl. Non-Classical Logics*, 11(1-2):11–34, 2001.
- [Darwiche, 2011] A. Darwiche. SDD: A New Canonical Representation of Propositional Knowledge Bases. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 819–826, 2011.
- [Darwiche and Marquis, 2002] A. Darwiche and P. Marquis. A Knowledge Compilation Map. *J. Artif. Intell. Res.*, 17:229–264, 2002.
- [Duris et al., 2004] P. Duris, J. Hromkovic, S. Jukna, M. Sauerhoff, and G. Schnitger. On Multi-Partition Communication Complexity. *Inf. Comput.*, 194(1):49–75, 2004.
- [Gogic et al., 1995] G. Gogic, H.A. Kautz, C.H. Papadimitriou, and B. Selman. The Comparative Linguistics of Knowledge Representation. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, pages 862–869, 1995.
- [Hromkovic, 1997] J. Hromkovic. *Communication Complexity and Parallel Computing*. Springer, 1997.
- [Jukna and Schnitger, 2002] S. Jukna and G. Schnitger. Triangle-Freeness is Hard to Detect. *Combinatorics, Probability & Computing*, 11(6):549–569, 2002.
- [Jukna, 2012] S. Jukna. *Boolean Function Complexity*. Springer, 2012.
- [Kushilevitz and Nisan, 1997] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.
- [Levesque and Brachman, 1987] H.J. Levesque and R.J. Brachman. Expressiveness and Tractability in Knowledge Representation and Reasoning. *Comput. Intell.*, 3:78–93, 1987.
- [Lipton and Sedgewick, 1981] R.J. Lipton and R. Sedgewick. Lower Bounds for VLSI. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing (STOC 1981)*, pages 300–307, 1981.
- [Marquis, 2015] P. Marquis. Compile! In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 4112–4118, 2015.
- [Pipatsrisawat and Darwiche, 2010] T. Pipatsrisawat and A. Darwiche. A Lower Bound on the Size of Decomposable Negation Normal Form. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*, 2010.
- [Pipatsrisawat, 2010] T. Pipatsrisawat. *Reasoning with Propositional Knowledge: Frameworks for Boolean Satisfiability and Knowledge Compilation*. PhD thesis, University of California Los Angeles, 2010.
- [Ponnuswami and Venkateswaran, 2004] A.K. Ponnuswami and H. Venkateswaran. Monotone Multilinear Boolean Circuits for Bipartite Perfect Matching Require Exponential Size. In *Proceedings of the 24th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2004)*, pages 460–468, 2004.
- [Razgon, 2015] I. Razgon. Quasipolynomial Simulation of DNNF by a Non-Deterministic Read-Once Branching Program. In *Proceedings of the 21st International Conference on Principles and Practice of Constraint Programming (CP 2015)*, pages 367–375, 2015.
- [Sauerhoff, 2003] M. Sauerhoff. Approximation of Boolean Functions by Combinatorial Rectangles. *Theor. Comput. Sci.*, 1-3(301):45–78, 2003.
- [Wegener, 2000] I. Wegener. *Branching Programs and Binary Decision Diagrams*. SIAM, 2000.