

# Existential Closures for Knowledge Compilation\*

Pierre Marquis

CRIL-CNRS, Université d'Artois,  
Lens, France  
marquis@cril.univ-artois.fr

## Abstract

We study the existential closures of several propositional languages  $\mathcal{L}$  considered recently as target languages for knowledge compilation (KC), namely the incomplete fragments  $KROM-C$ ,  $HORN-C$ ,  $K/H-C$ ,  $renH-C$ ,  $AFF$ , and the corresponding disjunction closures  $KROM-C[V]$ ,  $HORN-C[V]$ ,  $K/H-C[V]$ ,  $renH-C[V]$ , and  $AFF[V]$ . We analyze the queries, transformations, expressiveness and succinctness of the resulting languages  $\mathcal{L}[\exists]$  in order to locate them in the KC map. As a by-product, we also address several issues concerning disjunction closures that were left open so far. From our investigation, the language  $HORN - C[V, \exists]$  (where disjunctions and existential quantifications can be applied to Horn CNF formulae) appears as an interesting target language for the KC purpose, challenging the influential DNNF languages.

## 1 Introduction

Knowledge compilation (KC) is concerned with pre-processing for improving the efficiency of computational tasks (see among others [Darwiche, 2001; Cadoli and Donini, 1998; Selman and Kautz, 1996; Schrag, 1996; del Val, 1994]). An important issue within this research area is the choice of a target language into which some pieces of data are to be translated during the off-line, compilation phase [Gogic *et al.*, 1995; Darwiche and Marquis, 2002]. In [Darwiche and Marquis, 2002], the authors argue that such a choice must be based both on the set of queries and transformations which can be achieved in polynomial time once the pre-processed pieces of data are represented in the target language, as well as the succinctness of the language (i.e., its ability to represent data using few space.) The basic queries considered in [Darwiche and Marquis, 2002] include tests for consistency, validity, implicates (clausal entailment), implicants, equivalence, sentential entailment, counting and enumerating theory models (**CO**, **VA**, **CE**, **IM**, **EQ**, **SE**, **CT**, **ME**.) The basic transformations are conditioning (**CD**), closures under the

connectives ( $\wedge C$ ,  $\wedge BC$ ,  $\vee C$ ,  $\vee BC$ ,  $\neg C$ ), and forgetting (**FO**, **SFO**.) The KC map given in [Darwiche and Marquis, 2002] is an evaluation of a dozen of significant propositional languages w.r.t. several criteria: the succinctness of the language and the set of queries and transformations it supports in polynomial time. Such a map can be used as a guide for targeting the “right language” given the requirements imposed by the application under consideration.

The KC map provided in [Darwiche and Marquis, 2002] has been extended to incorporate further propositional languages (also referred to as “fragments”), queries and transformations, see among others [Wachter and Haenni, 2006], [Fargier and Marquis, 2006], [Subbarayan *et al.*, 2007], [Fargier and Marquis, 2008b], [Fargier and Marquis, 2008a], [Fargier and Marquis, 2009]. In [Fargier and Marquis, 2008b; 2008a] an approach to define new target languages for KC has been pointed out. It consists in applying closure principles to previous languages. [Fargier and Marquis, 2008a] consider two disjunctive closure principles: disjunction ( $\vee$ ) and existential closure ( $\exists$ .) Intuitively, the disjunction principle when applied to a propositional language  $\mathcal{L}$  leads to a language  $\mathcal{L}[\vee]$  which qualifies disjunctions of formulae from  $\mathcal{L}$ , while existential closure applied to  $\mathcal{L}$  leads to a language  $\mathcal{L}[\exists]$  which qualifies existentially quantified formulae from  $\mathcal{L}$ . Whatever  $\mathcal{L}$ ,  $\mathcal{L}[\vee]$  satisfies polytime closure under  $\vee$  and  $\mathcal{L}[\exists]$  satisfies polytime forgetting. Applying any/both of those two principles to  $\mathcal{L}$  may lead to new fragments, which can prove strictly more succinct than  $\mathcal{L}$ . Thus, [Fargier and Marquis, 2008a] locate on the KC map all languages obtained by applying those closure principles to some complete languages, i.e., languages  $\mathcal{L}$  for which every propositional formula has an equivalent in  $\mathcal{L}$ . Thus, the disjunctive closures of the languages  $OBDD_{<}$  (ordered binary decision diagrams),  $DNF$  (disjunctive normal forms),  $DNNF$  (decomposable negation normal forms),  $CNF$  (conjunctive normal forms),  $PI$  (prime implicates),  $IP$  (prime implicants),  $MODS$  (models), considered in [Darwiche and Marquis, 2002], have been studied in [Fargier and Marquis, 2008a]. On the other hand, [Fargier and Marquis, 2008b] consider the disjunction closures  $KROM-C[V]$ ,  $HORN-C[V]$ ,  $K/H-C[V]$ ,  $renH-C[V]$ , and  $AFF[V]$ , composed respectively of disjunctions of Krom CNF formulae, disjunctions of Horn CNF formulae, disjunctions of Krom or Horn CNF formulae, disjunctions of renaming Horn CNF formulae, and disjunctions of affine formulae.

\*This research was supported by the project TUPLES-ANR-10-BLAN-0210 of the French National Agency for Research - Agence Nationale de la Recherche

Each of these languages is complete, unlike the underlying languages  $\text{KROM-C}$  (also known as the bijunctive fragment) [Krom, 1970],  $\text{HORN-C}$  [Horn, 1951],  $\text{AFF}$  (also known as the biconditional fragment) [Schaefer, 1978],  $\text{K/H-C}$  (Krom or Horn CNF formulae) and  $\text{renH-C}$ .

In the following, we complete the results reported in [Fargier and Marquis, 2008b; 2008a] by focusing on the existential closures of the ten languages  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$ ,  $\text{AFF}$ ,  $\text{KROM-C}[V]$ ,  $\text{HORN-C}[V]$ ,  $\text{K/H-C}[V]$ ,  $\text{renH-C}[V]$ , and  $\text{AFF}[V]$ . We evaluate each of them along the lines considered in [Darwiche and Marquis, 2002]. The contribution of the paper is mainly as follows:

- For each existential closure of  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$ ,  $\text{AFF}$ ,  $\text{KROM-C}[V]$ ,  $\text{HORN-C}[V]$ ,  $\text{K/H-C}[V]$ ,  $\text{renH-C}[V]$ , and  $\text{AFF}[V]$ , we identify the queries and transformations which are feasible in polynomial time, and those which are not (possibly under some standard assumptions of complexity theory.)
- We demonstrate that the existential closure of each language  $\mathcal{L}$  among the ten languages above is just as expressive as  $\mathcal{L}$ . As to succinctness, we prove that  $\mathcal{L}[\exists]$  is strictly more succinct than  $\mathcal{L}$ , except for  $\mathcal{L} = \text{KROM-C}$  and  $\mathcal{L} = \text{AFF}$  since for those two languages,  $\mathcal{L}[\exists]$  and  $\mathcal{L}$  are polynomially equivalent.
- We complete the results about disjunction closures provided in [Fargier and Marquis, 2008b]; especially, we show that **FO** is not satisfied by any of  $\text{HORN-C}[V]$ ,  $\text{K/H-C}[V]$ ,  $\text{renH-C}[V]$ .
- We show that neither  $\text{d-DNNF}$  nor  $\text{SDNNF}$  is strictly more succinct than any disjunction closure of  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$ ,  $\text{AFF}$ ; it shows such disjunction closures (and the existential closures of them) as interesting alternatives to  $\text{DNNF}$  languages.

The rest of the paper is organized as follows: in Section 2, the queries and transformations considered in the  $\text{KC}$  map, as well as the fundamental notions of expressiveness and succinctness, are recalled. The notions of disjunctive, disjunction and existential closure are also presented. In Section 3, our new results are presented. Section 4 discusses the results and provides some perspectives.

## 2 The $\text{KC}$ Map and Disjunctive Closures

In this paper, we consider subsets of the propositional language  $\text{QDAG}$  of quantified propositional DAGs.  $\text{QDAG}$  is given by:

**Definition 1 (QDAG)** *Let  $PS$  be a denumerable set of propositional variables.  $\text{QDAG}$  is the set of all finite, single-rooted DAGs  $\alpha$  where:*

- each leaf node of  $\alpha$  is labeled by a literal  $l$  over  $PS$ , or by a Boolean constant  $\top$  (always true) or  $\perp$  (always false);
- each internal node of  $\alpha$  is labeled by a connective  $c \in \{\wedge, \vee, \neg, \oplus\}$  and has as many children as required by  $c$ , or is labeled by  $\exists x$  (where  $x \in PS$ ) and has a single child.

All the propositional languages considered so far as target languages to  $\text{KC}$  are subsets of  $\text{QDAG}$ , and typically of  $\text{DAG}$ , the subset of  $\text{QDAG}$  where no node labeled by a quantification is allowed. Especially, the fragments considered in [Wachter and Haenni, 2006] are included in the subset of  $\text{DAG}$  where  $c$  is among  $\{\wedge, \vee, \neg\}$ , and the fragments considered in [Darwiche and Marquis, 2002] (especially  $\text{OBDD}_{<}$ ,  $\text{DNE}$ ,  $\text{DNNF}$ ,  $\text{CNF}$ ,  $\text{PI}$ ,  $\text{IP}$ ,  $\text{MODS}$ ) are subsets of  $\text{DAG-NNF}$  (the subset of the latter when  $\neg$  is disallowed.)

A literal (over  $V \subseteq PS$ ) is an element  $x \in V$  (a positive literal) or a negated one  $\neg x$  (a negative literal), or a Boolean constant.  $\bar{l}$  is the complementary literal of literal  $l$ , so that  $\overline{\top} = \perp$ ,  $\overline{\perp} = \top$ ,  $\bar{x} = \neg x$  and  $\overline{\neg x} = x$ . For a literal  $l$  different from a Boolean constant,  $\text{var}(l)$  denotes the corresponding variable: for  $x \in PS$ , we have  $\text{var}(x) = x$  and  $\text{var}(\neg x) = x$ . A clause (resp. a term) is a finite disjunction (resp. conjunction) of literals. An XOR-clause is a finite XOR-disjunction of literals (the XOR connective is denoted by  $\oplus$ .)

Each element  $\alpha$  of  $\text{QDAG}$  is called a  $\text{QDAG}$  formula.  $\text{Var}(\alpha)$  denotes the set of free variables  $x$  of  $\alpha$ , i.e., those variables  $x$  for which there exists a leaf node  $n_x$  of  $\alpha$  labelled by a literal  $l$  such that  $\text{var}(l) = x$  and there is a path from the root of  $\alpha$  to  $n_x$  such that no node from it is labelled by  $\exists x$ . The size  $|\alpha|$  of a  $\text{QDAG}$  formula  $\alpha$  is the number of nodes plus the number of arcs in the DAG.

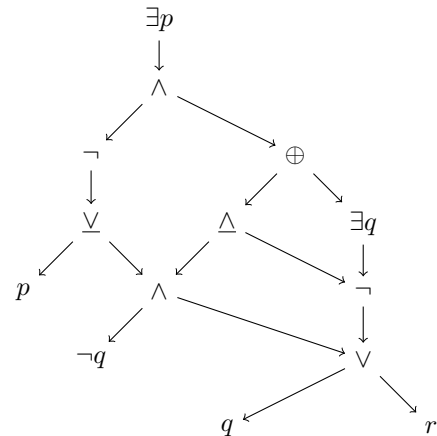


Figure 1: A  $\text{QDAG}$  formula.

Figure 1 presents a  $\text{QDAG}$  formula. Its set of free variables is  $\{q, r\}$ . The DAG rooted at the  $\Delta$  node is a  $\text{DAG}$  formula and the DAG rooted at the  $\underline{\vee}$  node is a  $\text{DAG-NNF}$  formula.

In the following, we consider the next queries and transformations:

**Definition 2 (queries)** *Let  $\mathcal{L} \subseteq \text{QDAG}$ .*

- $\mathcal{L}$  satisfies **CO** (resp. **VA**) iff there exists a polytime algorithm that maps every formula  $\alpha$  from  $\mathcal{L}$  to 1 if  $\alpha$  is consistent (resp. valid), and to 0 otherwise.
- $\mathcal{L}$  satisfies **CE** iff there exists a polytime algorithm that maps every formula  $\alpha$  from  $\mathcal{L}$  and every clause  $\gamma$  to 1 if  $\alpha \models \gamma$  holds, and to 0 otherwise.

- $\mathcal{L}$  satisfies **EQ** (resp. **SE**) iff there exists a polytime algorithm that maps every pair of formulae  $\alpha, \beta$  from  $\mathcal{L}$  to 1 if  $\alpha \equiv \beta$  (resp.  $\alpha \models \beta$ ) holds, and to 0 otherwise.
- $\mathcal{L}$  satisfies **IM** iff there exists a polytime algorithm that maps every formula  $\alpha$  from  $\mathcal{L}$  and every term  $\gamma$  to 1 if  $\gamma \models \alpha$  holds, and to 0 otherwise.
- $\mathcal{L}$  satisfies **CT** iff there exists a polytime algorithm that maps every formula  $\alpha$  from  $\mathcal{L}$  to a nonnegative integer that represents the number of models of  $\alpha$  over  $\text{Var}(\alpha)$  (in binary notation.)
- $\mathcal{L}$  satisfies **ME** iff there exists a polynomial  $p(\cdot, \cdot)$  and an algorithm that outputs all models of an arbitrary formula  $\alpha$  from  $\mathcal{L}$  in time  $p(n, m)$ , where  $n$  is the size of  $\alpha$  and  $m$  is the number of its models (over  $\text{Var}(\alpha)$ .)
- $\mathcal{L}$  satisfies **MC** (model checking) iff there exists a polytime algorithm that maps every formula  $\alpha$  from  $\mathcal{L}$  and every interpretation  $\omega$  over  $\text{Var}(\alpha)$  (represented as a term) to 1 if  $\omega$  is a model of  $\alpha$ , and to 0 otherwise.

**Definition 3 (transformations)** Let  $\mathcal{L} \subseteq \text{QDAG}$ .

- $\mathcal{L}$  satisfies **CD** iff there exists a polytime algorithm that maps every formula  $\alpha$  from  $\mathcal{L}$  and every consistent term  $\gamma$  to a formula from  $\mathcal{L}$  that is logically equivalent to the conditioning  $\alpha \mid \gamma$  of  $\alpha$  on  $\gamma$ , i.e., the formula obtained by replacing each free occurrence of variable  $x$  of  $\alpha$  by  $\top$  (resp.  $\perp$ ) if  $x$  (resp.  $\neg x$ ) is a positive (resp. negative) literal of  $\gamma$ .
- $\mathcal{L}$  satisfies **FO** iff there exists a polytime algorithm that maps every formula  $\alpha$  from  $\mathcal{L}$  and every subset  $\mathbf{X}$  of variables from  $PS$  to a formula from  $\mathcal{L}$  equivalent to  $\exists \mathbf{X}.\alpha$ . If the property holds for each singleton  $\mathbf{X}$ , we say that  $\mathcal{L}$  satisfies **SFO**.
- $\mathcal{L}$  satisfies  **$\wedge\mathbf{C}$**  (resp.  **$\vee\mathbf{C}$** ) iff there exists a polytime algorithm that maps every finite set of formulae  $\alpha_1, \dots, \alpha_n$  from  $\mathcal{L}$  to a formula of  $\mathcal{L}$  that is logically equivalent to  $\alpha_1 \wedge \dots \wedge \alpha_n$  (resp.  $\alpha_1 \vee \dots \vee \alpha_n$ .)
- $\mathcal{L}$  satisfies  **$\wedge\mathbf{BC}$**  (resp.  **$\vee\mathbf{BC}$** ) iff there exists a polytime algorithm that maps every pair of formulae  $\alpha$  and  $\beta$  from  $\mathcal{L}$  to a formula of  $\mathcal{L}$  that is logically equivalent to  $\alpha \wedge \beta$  (resp.  $\alpha \vee \beta$ .)
- $\mathcal{L}$  satisfies  **$\neg\mathbf{C}$**  iff there exists a polytime algorithm that maps every formula  $\alpha$  from  $\mathcal{L}$  to a formula of  $\mathcal{L}$  logically equivalent to  $\neg\alpha$ .

We also consider the following notions of expressiveness and succinctness:

**Definition 4 (expressiveness)** Let  $\mathcal{L}_1$  and  $\mathcal{L}_2$  be two subsets of  $\text{QDAG}$ .  $\mathcal{L}_1$  is at least as expressive as  $\mathcal{L}_2$ , denoted  $\mathcal{L}_1 \leq_e \mathcal{L}_2$ , iff for every formula  $\alpha \in \mathcal{L}_2$ , there exists an equivalent formula  $\beta \in \mathcal{L}_1$ .

**Definition 5 (succinctness)** Let  $\mathcal{L}_1$  and  $\mathcal{L}_2$  be two subsets of  $\text{QDAG}$ .  $\mathcal{L}_1$  is at least as succinct as  $\mathcal{L}_2$ , denoted  $\mathcal{L}_1 \leq_s \mathcal{L}_2$ , iff there exists a polynomial  $p$  such that for every formula  $\alpha \in \mathcal{L}_2$ , there exists an equivalent formula  $\beta \in \mathcal{L}_1$  where  $|\beta| \leq p(|\alpha|)$ .

Finally, we take advantage of the following restriction of the succinctness relation:

**Definition 6 (polynomial translation)** Let  $\mathcal{L}_1$  and  $\mathcal{L}_2$  be two subsets of  $\text{QDAG}$ .  $\mathcal{L}_1$  is said to be polynomially translatable into  $\mathcal{L}_2$ , noted  $\mathcal{L}_1 \geq_p \mathcal{L}_2$ , iff there exists a polytime algorithm  $f$  such that for every  $\alpha \in \mathcal{L}_1$ , we have  $f(\alpha) \in \mathcal{L}_2$  and  $f(\alpha) \equiv \alpha$ .

Whenever  $\mathcal{L}_1$  is polynomially translatable into  $\mathcal{L}_2$ , every query which is supported in polynomial time in  $\mathcal{L}_2$  also is supported in polynomial time in  $\mathcal{L}_1$ ; and conversely, every query which is not supported in polynomial time in  $\mathcal{L}_1$  unless  $\mathbf{P} = \mathbf{NP}$  (resp. unless the polynomial hierarchy  $\text{PH}$  collapses) cannot be supported in polynomial time in  $\mathcal{L}_2$ , unless  $\mathbf{P} = \mathbf{NP}$  (resp. unless  $\text{PH}$  collapses.)

$\sim_e$  is the symmetric part of  $\leq_e$  defined by  $\mathcal{L}_1 \sim_e \mathcal{L}_2$  iff  $\mathcal{L}_1 \leq_e \mathcal{L}_2$  and  $\mathcal{L}_2 \leq_e \mathcal{L}_1$ .  $<_e$  is the asymmetric part of  $\leq_e$  defined by  $\mathcal{L}_1 <_e \mathcal{L}_2$  iff  $\mathcal{L}_1 \leq_e \mathcal{L}_2$  and  $\mathcal{L}_2 \not\leq_e \mathcal{L}_1$ . Similarly,  $\sim_s$  (resp.  $\sim_p$ ) is the symmetric part of  $\leq_s$  (resp.  $\leq_p$ .)  $<_s$  (resp.  $<_p$ ) is the asymmetric part of  $\leq_s$  (resp.  $\leq_p$ .) When  $\mathcal{L}_1 \sim_p \mathcal{L}_2$ ,  $\mathcal{L}_1$  and  $\mathcal{L}_2$  are said to be polynomially equivalent. Obviously enough, polynomially equivalent fragments are equally efficient (and succinct) and possess the same set of tractable queries and transformations.

We are now ready to present disjunctive closures. Intuitively, a closure principle applied to a propositional fragment  $\mathcal{L}$  defines a new propositional language, called a closure of  $\mathcal{L}$ , through the application of “operators” (i.e., connectives or quantifications.) The resulting closure is said to be *disjunctive* when the operators are among  $\vee$  and  $\exists x$  with  $x \in PS$ . Formally:

**Definition 7 (disjunctive closures)** Let  $\mathcal{L} \subseteq \text{QDAG}$  and  $\Delta \subseteq \{\vee, \exists\}$ . The closure  $\mathcal{L}[\Delta]$  of  $\mathcal{L}$  by  $\Delta$  is the subset of  $\text{QDAG}$  inductively defined as follows:<sup>1</sup>

1. if  $\alpha \in \mathcal{L}$ , then  $\alpha \in \mathcal{L}[\Delta]$ ,
2. if  $\vee \in \Delta$  and  $\alpha_i \in \mathcal{L}[\Delta]$  for each  $i \in 1, \dots, n$ , then  $\vee(\alpha_1, \dots, \alpha_n) \in \mathcal{L}[\Delta]$ ,
3. if  $\exists \in \Delta$ ,  $x \in PS$ , and  $\alpha \in \mathcal{L}[\Delta]$ , then  $\exists x.\alpha \in \mathcal{L}[\Delta]$ .

Thus, an element of  $\mathcal{L}[\Delta]$  can be viewed as a “tree” which “internal nodes” are labelled by quantifications of the form  $\exists x$  or by  $\vee$  and its “leaf nodes” are labelled by elements of  $\mathcal{L}$ . Accordingly, the formulae  $\alpha_i$  considered in item 2. of Definition 7 do not share any common subgraphs.

The set  $\mathcal{D}(\mathcal{L})$  (resp.  $\vee(\mathcal{L})$ ,  $\exists(\mathcal{L})$ ) of all disjunctive (resp. disjunction, existential) closures of a subset  $\mathcal{L}$  of  $\text{QDAG}$  is defined inductively by:

- $\mathcal{L}$  belongs to  $\mathcal{D}(\mathcal{L})$  (resp.  $\vee(\mathcal{L})$ ,  $\exists(\mathcal{L})$ );
- If  $\mathcal{L}$  belongs to  $\mathcal{D}(\mathcal{L})$  (resp.  $\vee(\mathcal{L})$ ,  $\exists(\mathcal{L})$ ) and  $\Delta \subseteq \{\vee, \exists\}$  (resp.  $\Delta \subseteq \{\vee\}$ ,  $\Delta \subseteq \{\exists\}$ ), then  $\mathcal{L}[\Delta]$  belongs to  $\mathcal{D}(\mathcal{L})$  (resp.  $\vee(\mathcal{L})$ ,  $\exists(\mathcal{L})$ .)

[Fargier and Marquis, 2008a] provide several general-scope characterization results for disjunctive closures. Especially, they show that for a given  $\mathcal{L} \subseteq \text{DAG}$ , only three disjunctive closures are worth to be considered since  $\mathcal{L}[\exists][\exists] =$

<sup>1</sup>In order to alleviate the notations, when  $\Delta = \{\delta_1, \dots, \delta_n\}$ , we write  $\mathcal{L}[\delta_1, \dots, \delta_n]$  instead of  $\mathcal{L}[\{\delta_1, \dots, \delta_n\}]$ .

$\mathcal{L}[\exists]$ ,  $\mathcal{L}[V][V] = \mathcal{L}[V]$  and  $\mathcal{L}[\exists][V] \sim_p \mathcal{L}[V][\exists] \sim_p \mathcal{L}[V, \exists]$  (see item 4. of Proposition 1 and item 1. of Proposition 2 in [Fargier and Marquis, 2008a].) They also study the disjunctive closures of the languages  $\text{OBDD}_{\leq}$ ,  $\text{DNF}$ ,  $\text{DNNF}$ ,  $\text{CNF}$ ,  $\text{PI}$ ,  $\text{IP}$ ,  $\text{MODS}$  considered in [Darwiche and Marquis, 2002].

### 3 Analyzing New Existential Closures

[Fargier and Marquis, 2008b] focus on the disjunction closures of the following incomplete fragments:

#### Definition 8 (some incomplete fragments)

- $\text{KROM-C}$  is the subset of  $\text{CNF}$  consisting of formulae in which each clause is binary, i.e., it contains at most two literals.
- $\text{HORN-C}$  is the subset of  $\text{CNF}$  consisting of formulae in which each clause is Horn, i.e., it contains at most one positive literal.
- $\text{K/H-C}$  is the union of  $\text{KROM-C}$  and  $\text{HORN-C}$ .
- $\text{renH-C}$  is the subset of  $\text{CNF}$  consisting of formulae  $\alpha$  for which there exists a subset  $V$  of  $\text{Var}(\alpha)$  (called a Horn renaming for  $\alpha$ ) such that the formula  $V(\alpha)$  obtained by substituting in  $\alpha$  every literal  $l$  over  $V$  by its complementary literal  $\bar{l}$  is a  $\text{HORN-C}$  formula.<sup>2</sup>
- $\text{AFF}$  is the subset of  $\text{DAG}$  consisting of conjunctions of XOR clauses.

In the following, we complete the results provided in [Fargier and Marquis, 2008b] by studying the queries, transformations, expressiveness and succinctness of the existential closures of the propositional fragments considered in [Fargier and Marquis, 2008b], i.e., we consider the languages  $\text{KROM-C}[\exists]$ ,  $\text{HORN-C}[\exists]$ ,  $\text{K/H-C}[\exists]$ ,  $\text{renH-C}[\exists]$ ,  $\text{AFF}[\exists]$ ,  $\text{KROM-C}[V, \exists]$ ,  $\text{HORN-C}[V, \exists]$ ,  $\text{K/H-C}[V, \exists]$ ,  $\text{renH-C}[V, \exists]$ ,  $\text{AFF}[V, \exists]$ .

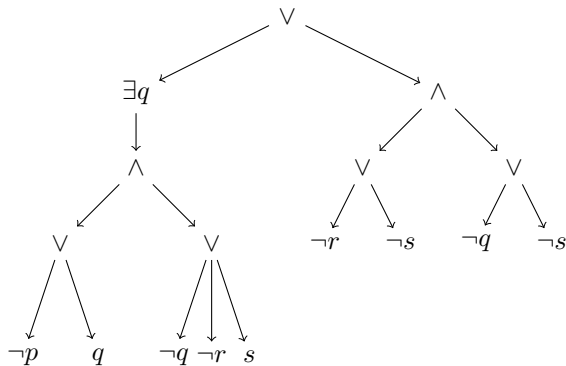


Figure 2: A  $\text{HORN-C}[V, \exists]$  formula.

Figure 2 presents a  $\text{HORN-C}[V, \exists]$  formula equivalent to the  $\text{CNF}$  formula  $(\neg p \vee \neg r) \wedge (\neg r \vee \neg s) \wedge (\neg q \vee \neg s)$ .

<sup>2</sup>Note that there exists linear time algorithms for recognizing  $\text{renH-C}$  formulae (see e.g. [del Val, 2000]); furthermore, such recognition algorithms typically give a Horn renaming when it exists.

$\mathcal{L}$	CO	VA	CE	IM	EQ	SE	CT	ME	MC
$\text{renH-C}[V, \exists]$	✓	○	✓	○	○	○	○	✓	✓
$\text{K/H-C}[V, \exists]$	✓	○	✓	○	○	○	○	✓	✓
$\text{HORN-C}[V, \exists]$	✓	○	✓	○	○	○	○	✓	✓
$\text{renH-C}[\exists]$	✓	✓	✓	✓	○	○	○	✓	✓
$\text{K/H-C}[\exists]$	✓	✓	✓	✓	○	○	○	✓	✓
$\text{HORN-C}[\exists]$	✓	✓	✓	✓	○	○	○	✓	✓

Table 1: Queries. ✓ means “satisfies” and ○ means “does not satisfy unless  $\text{P} = \text{NP}$ .”

$\mathcal{L}$	CD	FO	SFO	$\wedge\text{C}$	$\wedge\text{BC}$	$\vee\text{C}$	$\vee\text{BC}$	$\neg\text{C}$
$\text{renH-C}[V, \exists]$	✓	✓	✓	○	○	✓	✓	○
$\text{K/H-C}[V, \exists]$	✓	✓	✓	○	○	✓	✓	○
$\text{HORN-C}[V, \exists]$	✓	✓	✓	○	✓	✓	✓	○
$\text{renH-C}[\exists]$	✓	✓	✓	!	!	!	!	!
$\text{K/H-C}[\exists]$	✓	✓	✓	○	○	!	!	!
$\text{HORN-C}[\exists]$	✓	✓	✓	✓	✓	!	!	!

Table 2: Transformations. ✓ means “satisfies,” while ○ means “does not satisfy unless  $\text{P} = \text{NP}$ .” ! means that the transformation is not always feasible within the fragment.

We know that each of  $\text{KROM-C}$  and  $\text{AFF}$  satisfies **FO** [Fargier and Marquis, 2008b]. Furthermore, for any  $\mathcal{L} \subseteq \text{DAG}$ , if  $\mathcal{L}$  satisfies **FO**, then  $\mathcal{L}[V]$  satisfies **FO** as well. As a consequence, we immediately get that

- $\text{KROM-C}[\exists] \sim_p \text{KROM-C}$ , and  $\text{AFF}[\exists] \sim_p \text{AFF}$ ,
- $\text{KROM-C}[V, \exists] \sim_p \text{KROM-C}[V]$ , and  $\text{AFF}[V, \exists] \sim_p \text{AFF}[V]$ .

Since  $\text{KROM-C}$ ,  $\text{AFF}$ ,  $\text{KROM-C}[V]$ , and  $\text{AFF}[V]$  have been studied in [Fargier and Marquis, 2008b], we mainly focus on  $\text{HORN-C}[\exists]$ ,  $\text{K/H-C}[\exists]$ ,  $\text{renH-C}[\exists]$ ,  $\text{HORN-C}[V, \exists]$ ,  $\text{K/H-C}[V, \exists]$ ,  $\text{renH-C}[V, \exists]$  in the following.

As to queries and transformations, we have obtained the following results:

**Proposition 1** The results given in Table 1 and in Table 2 hold.

As to expressiveness, it turns out that for any language  $\mathcal{L}$  among  $\text{K/H-C}$ ,  $\text{HORN-C}$ ,  $\text{renH-C}$ ,  $\text{K/H-C}[V]$ ,  $\text{HORN-C}[V]$ ,  $\text{renH-C}[V]$ , going from  $\mathcal{L}$  to  $\mathcal{L}[\exists]$  does not lead to a shift:

**Proposition 2** For every  $\mathcal{L}$  among  $\text{K/H-C}$ ,  $\text{HORN-C}$ ,  $\text{renH-C}$ ,  $\text{K/H-C}[V]$ ,  $\text{HORN-C}[V]$ ,  $\text{renH-C}[V]$ , we have  $\mathcal{L} \sim_e \mathcal{L}[\exists]$ .

As a consequence (see [Fargier and Marquis, 2008b]), we have  $\text{HORN-C}[V, \exists] \sim_e \text{K/H-C}[V, \exists] \sim_e \text{renH-C}[V, \exists]$ : these three languages are just equally expressive since they are complete for propositional logic (just observe that the complete language  $\text{DNF}$  is a subset of each of them.) Contrastingly, each of  $\text{HORN-C}[\exists]$ ,  $\text{K/H-C}[\exists]$ ,  $\text{renH-C}[\exists]$  is strictly less expressive than any complete language.

Furthermore, we have:

$$\begin{aligned} \text{renH-C}[\exists] &<_e \text{K/H-C}[\exists] <_e \text{HORN-C}[\exists] \\ \text{K/H-C}[\exists] &<_e \text{KROM-C}[\exists] \end{aligned}$$

while  $\text{HORN-C}[\exists]$  and  $\text{KROM-C}[\exists]$  are incomparable w.r.t.  $\leq_e$  (and  $\text{AFF}[\exists]$ , which is polynomially equivalent to  $\text{AFF}$ , is incomparable w.r.t.  $\leq_e$  with any of the three incomplete fragments above.)

As to succinctness, the picture is rather different. Typically going from  $\mathcal{L}$  to  $\mathcal{L}[\exists]$  leads to a succinctness increase:

**Proposition 3**

1. None of CNF or DNF is at least as succinct as any of  $\text{HORN-C}[\exists]$ ,  $\text{K/H-C}[\exists]$  or  $\text{renH-C}[\exists]$ .
2.  $\text{AC}^3$ , the language containing all disjunctions of CNF formulae and all conjunctions of DNF formulae, is not at least as succinct as any of  $\text{HORN-C}[\forall, \exists]$ ,  $\text{K/H-C}[\forall, \exists]$  and  $\text{renH-C}[\forall, \exists]$ .

Since  $\text{HORN-C}$  (resp.  $\text{K/H-C}$ ,  $\text{renH-C}$ ) is a subset of both CNF and  $\text{HORN-C}[\exists]$  (resp.  $\text{K/H-C}[\exists]$ ,  $\text{renH-C}[\exists]$ ), item 1. of Proposition 3 shows that going from  $\mathcal{L}$  to  $\mathcal{L}[\exists]$  for any of these three languages leads to a strictly more succinct language. Furthermore, since  $\text{HORN-C}[\forall]$  (resp.  $\text{K/H-C}[\forall]$ ,  $\text{renH-C}[\forall]$ ) is a subset of both  $\text{AC}^3$  and  $\text{HORN-C}[\forall, \exists]$  (resp.  $\text{K/H-C}[\forall, \exists]$ ,  $\text{renH-C}[\forall, \exists]$ ), item 2. of Proposition 3 shows that a similar conclusion can be drawn for those three languages, namely  $\text{HORN-C}[\forall]$  (resp.  $\text{K/H-C}[\forall]$ ,  $\text{renH-C}[\forall]$ ) is strictly less succinct than  $\text{HORN-C}[\forall, \exists]$  (resp.  $\text{K/H-C}[\forall, \exists]$ ,  $\text{renH-C}[\forall, \exists]$ .) As a by-product, this shows that for sure, none of  $\text{HORN-C}[\forall]$ ,  $\text{K/H-C}[\forall]$ ,  $\text{renH-C}[\forall]$  satisfies **FO**, an issue left open in [Fargier and Marquis, 2008b].

Finally, we have also compared the disjunction closures of  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$ ,  $\text{AFF}$ ,  $\text{PI}$  with the two main subsets of  $\text{DNNF}$ , namely the set  $\text{d-DNNF}$  of deterministic  $\text{DNNFs}$  and the set  $\text{SDNNF}$  of structured  $\text{DNNFs}$  (the union for all vtrees  $T$  of all  $\text{DNNF}$  languages  $\text{DNNF}_T$  respecting  $T$ ) (see [Pipatsrisawat and Darwiche, 2008] for details.) What makes those subsets so significant is that every available compilation algorithm which outputs a  $\text{DNNF}$  formula actually computes a  $\text{d-DNNF}$  formula [Darwiche, 2001; 2004] or a  $\text{SDNNF}$  formula [Pipatsrisawat and Darwiche, 2008; 2010b]. First of all, as an easy consequence of the fact that  $\text{DNNF}$  is not at least as succinct as  $\text{DNF}$ , unless  $\text{PH}$  collapses, we have that:

**Proposition 4**  $\text{d-DNNF}$  is not at least as succinct as any language among the existential closures of  $\text{KROM-C}[\forall]$ ,  $\text{HORN-C}[\forall]$ ,  $\text{K/H-C}[\forall]$ ,  $\text{renH-C}[\forall]$ ,  $\text{AFF}[\forall]$ ,  $\text{PI}[\forall]$ , and  $\text{OBDD}_{<}[\forall]$ , unless  $\text{PH}$  collapses.

This shows those existential closures as interesting alternatives to  $\text{d-DNNF}$  for applications where **FO**,  $\forall\text{C}$  (and for some of them  $\wedge\text{BC}$ ) are required since such closures offer them while  $\text{d-DNNF}$  does not (the price to be paid is that **VA**, **IM** and **CT** are lost while they are offered by  $\text{d-DNNF}$ .)

As to  $\text{SDNNF}$ , we have got that:

**Proposition 5**  $\text{SDNNF}$  is not at least as succinct as any language among the disjunctive closures of  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$ ,  $\text{AFF}$ ,  $\text{PI}$ , and  $\text{CNF}$ .

Thus, every language that is at most as succinct as  $\text{SDNNF}$  is not at least as succinct as any disjunctive closure of  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$ ,  $\text{AFF}$ ,  $\text{PI}$ ,

$\text{CNF}$ . This includes in particular  $\text{OBDD}_{<}$ ,  $\text{OBDD}_{<}[\forall]$ , the language  $\text{O-DDG}$  of Ordered Decomposable Decision Diagrams [Fargier and Marquis, 2006], and the language  $\text{AOMDD}$  of AND/OR Multi-Valued Decision Diagrams (restricted to propositional variables) [Mateescu *et al.*, 2008] (or equivalently the language  $\text{SO-DDG}$  of Strongly Ordered Decomposable Decision Diagrams [Fargier and Marquis, 2006].)

## 4 Conclusion and Perspectives

In this paper, we have studied the existential closures of both incomplete languages ( $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$ , and  $\text{AFF}$ ) and complete languages (the corresponding disjunction closures  $\text{KROM-C}[\forall]$ ,  $\text{HORN-C}[\forall]$ ,  $\text{K/H-C}[\forall]$ ,  $\text{renH-C}[\forall]$ , and  $\text{AFF}[\forall]$ .) The results given above show that for each complete language  $\mathcal{L}$  under consideration, the corresponding existential closure  $\mathcal{L}[\exists]$  is at least as good as  $\mathcal{L}$  as a target language for **KC**. Indeed,  $\mathcal{L}$  and  $\mathcal{L}[\exists]$  satisfy the same queries and when  $\mathcal{L}$  does not satisfy **FO**,  $\mathcal{L}[\exists]$  offers it for free and is equally expressive but strictly more succinct than  $\mathcal{L}$ . For the incomplete languages considered here, namely  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$ ,  $\text{AFF}$ , the same conclusions can be drawn as to transformations, expressiveness and succinctness; the price to be paid for getting **FO** for free and obtaining a strictly more succinct language is paid by the loss of the **EQ** and **SE** queries when the existential closures of  $\text{HORN-C}$ ,  $\text{K/H-C}$ ,  $\text{renH-C}$  are considered.

Our study also shows  $\text{HORN-C}[\forall, \exists]$  as a valuable complete target language for the **KC** purpose. Indeed,  $\text{HORN-C}[\forall, \exists]$  satisfies the same queries and transformations as  $\text{DNNF}_T$  or  $\text{AFF}[\forall]$  (which is polynomially equivalent to  $\text{AFF}[\forall, \exists]$ .) Especially,  $\text{HORN-C}[\forall, \exists]$  satisfies  $\wedge\text{BC}$  and this paves the way for bottom-up compilation algorithms targeting  $\text{HORN-C}[\forall, \exists]$ . As noted in [Pipatsrisawat and Darwiche, 2008], this is important for applications from formal verification based on unbounded model checking which require bottom-up, incremental compilation of formulae, where pieces of the knowledge base are compiled independently and then conjoined together.<sup>3</sup> Furthermore,  $\text{HORN-C}[\forall, \exists]$  guarantees some polynomial-sized representations for families of propositional formulae, like the  $\text{CNF}$  representations of circular bit shift functions (resp. the  $\text{CNF}$  formulae  $\alpha_n = \bigwedge_{i=1}^n (\neg x_i \vee \neg y_i)$ ), which have only exponential-sized  $\text{SDNNF}$  representations (resp. exponential-sized  $\text{AFF}[\forall]$  representations, see [Fargier and Marquis, 2008b].) Compared to  $\text{d-DNNF}$ , while it does not satisfy the queries **VA**, **IM** and **CT**,  $\text{HORN-C}[\forall, \exists]$  offers the transformations **FO**,  $\forall\text{C}$  and  $\wedge\text{BC}$ . Finally,  $\text{d-DNNF}$  is not at least as succinct as  $\text{HORN-C}[\forall, \exists]$  unless  $\text{PH}$  collapses.

From the practical side, it is important to note that some empirical evidence in favour of  $\text{HORN-C}[\forall, \exists]$  compilations and  $\text{renH-C}[\forall, \exists]$  already exists. On the one hand, [Nishimura *et al.*, 2004] (resp. [Samer and Szeider, 2008]) have shown that the problem of determining whether a given

<sup>3</sup>Since  $\text{HORN-C}[\forall, \exists]$  satisfies **CO** and **CL** (i.e., every propositional clause has a polynomial-sized representation in the language), getting  $\wedge\text{BC}$  is optimal in the sense that no propositional language can satisfy both  $\wedge\text{C}$ , **CO**, and **CL**, unless  $\text{P} = \text{NP}$ .

CNF formula  $\alpha$  has a strong HORN-C-backdoor set (resp. a HORN-C-backdoor tree) containing at most  $k$  variables (resp. leaves) is fixed-parameter tractable with parameter  $k$ . Interestingly, the algorithm given in [Samer and Szeider, 2008] can be used to determine "efficiently" (i.e., for sufficiently "small"  $k$ ) whether a HORN - C[V] compilation of reasonable size (i.e., linear in  $k$  and the size of  $\alpha$ ) exists. As mentioned in [Samer and Szeider, 2008]: "There is some empirical evidence that real-world instances actually have small backdoor sets". On the other hand, [Boufkhad *et al.*, 1997] present some compilation algorithms targeting respectively HORN - C[V] and  $\text{renH} - \text{C}[V]$ , and evaluate them on a number of benchmarks. While such results show the feasibility of HORN - C[V,  $\exists$ ] compilation and  $\text{renH} - \text{C}[V, \exists]$  compilation, we can hardly use those results to compare the two target fragments HORN - C[V,  $\exists$ ] and  $\text{renH} - \text{C}[V, \exists]$  with  $\text{OBDD}_{\leq}$  and DNNF for which some experimental results are also available. Indeed, the compilation algorithms given in [Boufkhad *et al.*, 1997] are based on an old-style DPLL SAT solver, and the performances of such solvers are dramatically overtaken by those of modern SAT solvers, based on a CDCL architecture. Accordingly, the main perspective of this work is to develop and evaluate compilation algorithms based on a CDCL SAT solver, and which target HORN - C[V,  $\exists$ ] and  $\text{renH} - \text{C}[V, \exists]$ .

## Acknowledgments

The author would like to thank the anonymous reviewers for their helpful comments.

## References

- [Boufkhad *et al.*, 1997] Y. Boufkhad, E. Grégoire, P. Marquis, B. Mazure, and L. Saïs. Tractable cover compilations. In *Proc. of IJCAI'97*, pages 122–127, 1997.
- [Cadoli and Donini, 1998] M. Cadoli and F.M. Donini. A survey on knowledge compilation. *AI Communications*, 10(3–4):137–150, 1998.
- [Darwiche and Marquis, 2002] A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
- [Darwiche, 2001] A. Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):608–647, 2001.
- [Darwiche, 2004] A. Darwiche. New advances in compiling cnf into decomposable negation normal form. In *Proc. of ECAI'04*, pages 328–332, 2004.
- [Dechter and Rish, 1994] R. Dechter and I. Rish. Directional resolution: the Davis-Putnam procedure, revisited. In *Proc. of KR'94*, pages 134–145, 1994.
- [del Val, 1994] A. del Val. Tractable databases: How to make propositional unit resolution complete through compilation. In *Proc. of KR'94*, pages 551–561, 1994.
- [del Val, 2000] A. del Val. On 2-SAT and Renamable Horn. In *Proc. of AAAI'00*, pages 279–284, 2000.
- [Fargier and Marquis, 2006] H. Fargier and P. Marquis. On the use of partially ordered decision graphs in knowledge compilation and quantified Boolean formulae. In *Proc. of AAAI'06*, pages 42–47, 2006.
- [Fargier and Marquis, 2008a] H. Fargier and P. Marquis. Extending the knowledge compilation map: Closure principles. In *Proc. of ECAI'08*, pages 50–54, 2008.
- [Fargier and Marquis, 2008b] H. Fargier and P. Marquis. Extending the knowledge compilation map: Krom, Horn, affine and beyond. In *Proc. of AAAI'08*, pages 442–447, 2008.
- [Fargier and Marquis, 2009] H. Fargier and P. Marquis. Knowledge compilation properties of trees-of-bdds, revisited. In *Proc. of IJCAI'09*, pages 772–777, 2009.
- [Gogic *et al.*, 1995] G. Gogic, H.A. Kautz, Ch.H. Papadimitriou, and B. Selman. The comparative linguistics of knowledge representation. In *Proc. of IJCAI'95*, pages 862–869, 1995.
- [Horn, 1951] A. Horn. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic*, 16:14–21, 1951.
- [Krom, 1970] M.R. Krom. The decision problem for formulas in prenex conjunctive normal form with binary disjunctions. *Journal of Symbolic Logic*, 35:210–216, 1970.
- [Lang *et al.*, 2003] J. Lang, P. Liberatore, and P. Marquis. Propositional independence - formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.
- [Mateescu *et al.*, 2008] R. Mateescu, R. Dechter, and R. Marinescu. And/or multi-valued decision diagrams (aomdds) for graphical models. *Journal of Artificial Intelligence Research*, 33:465–519, 2008.
- [Nishimura *et al.*, 2004] N. Nishimura, P. Ragde, and S. Szeider. Detecting backdoor sets with respect to horn and binary clauses. In *Proc. of SAT'04*, 2004.
- [Pipatsrisawat and Darwiche, 2008] K. Pipatsrisawat and A. Darwiche. New compilation languages based on structured decomposability. In *Proc. of AAAI'08*, pages 517–522, 2008.
- [Pipatsrisawat and Darwiche, 2010a] K. Pipatsrisawat and A. Darwiche. A lower bound on the size of decomposable negation normal form. In *Proc. of AAAI'10*, 2010.
- [Pipatsrisawat and Darwiche, 2010b] K. Pipatsrisawat and A. Darwiche. Top-down algorithms for constructing structured dnnf: Theoretical and practical implications. In *Proc. of ECAI'10*, pages 3–8, 2010.
- [Roth, 1996] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1–2):273–302, 1996.
- [Samer and Szeider, 2008] M. Samer and S. Szeider. Backdoor trees. In *Proc. of AAAI'08*, pages 363–368, 2008.
- [Schaefer, 1978] Th. J. Schaefer. The complexity of satisfiability problems. In *Proc. of STOC'78*, pages 216–226, 1978.
- [Schrag, 1996] R. Schrag. Compilation for critically constrained knowledge bases. In *Proc. of AAAI'96*, pages 510–515, 1996.

- [Selman and Kautz, 1996] B. Selman and H.A. Kautz. Knowledge compilation and theory approximation. *Journal of the ACM*, 43:193–224, 1996.
- [Sipser, 1983] M. Sipser. Borel sets and circuit complexity. In *Proc. of STOC'83*, pages 61–69, 1983.
- [Subbarayan *et al.*, 2007] S. Subbarayan, L. Bordeaux, and Y. Hamadi. Knowledge compilation properties of tree-of-BDDs. In *Proc. of AAAI'07*, pages 502–507, 2007.
- [Wachter and Haenni, 2006] M. Wachter and R. Haenni. Propositional DAGs: A new graph-based language for representing Boolean functions. In *Proc. of KR'06*, pages 277–285, 2006.

## Appendix

**Proof:**[Proposition 1]

We start with the queries:

- **CO:** Direct from item 1. of Proposition 4 from [Fargier and Marquis, 2008a] showing that when a subset  $\mathcal{L}$  of DAG satisfies **CO**, then  $\mathcal{L}[\exists]$  and  $\mathcal{L}[V, \exists]$  satisfies **CO** as well, plus the fact that each of  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{renH-C}$  satisfies **CO**.
- **VA:**
  - $\text{renH-C}[V, \exists]$ ,  $\text{K/H-C}[V, \exists]$  and  $\text{HORN-C}[V, \exists]$ : Direct from the fact that DNF is a subset of each of these fragments and DNF does not satisfy **VA** unless  $\text{P} = \text{NP}$ .
  - $\text{renH-C}[\exists]$ ,  $\text{K/H-C}[\exists]$  and  $\text{HORN-C}[\exists]$ :  $\alpha$  is valid iff it is implied by the term *new* where  $\text{new} \in \text{PS} \setminus \text{Var}(\alpha)$ , which belongs to each of the three fragments under consideration here. The fact that each of  $\text{renH-C}[\exists]$ ,  $\text{K/H-C}[\exists]$  and  $\text{HORN-C}[\exists]$  satisfies **IM** concludes the proof.
  - $\text{HORN-C}[\exists]$ : A formula  $\alpha$  from  $\text{HORN-C}[\exists]$  is valid iff its universal closure is valid. The fact that the validity problem for closed, prenex QBFs with a  $\text{HORN-C}$  matrix is in **P** concludes the proof.
  - $\text{K/H-C}[\exists]$ : Let  $\alpha$  be a formula from  $\text{K/H-C}[\exists]$ ; if  $\alpha$  is a  $\text{HORN-C}[\exists]$  formula, then its validity can be decided in polynomial time (see the item above); in the remaining case, it is a  $\text{KROM-C}[\exists]$  formula; now, since  $\text{KROM-C}[\exists]$  is polynomially translatable into  $\text{KROM-C}$ , and since  $\text{KROM-C}$  (and more generally its superset CNF) satisfies **VA**, its validity can also be decided in polynomial time in this case.
  - $\text{renH-C}[\exists]$ : Let us prove that a  $\text{renH-C}[\exists]$  formula  $\alpha = \exists X.\beta$  is valid iff for any Horn renaming  $V$  of  $\beta$ , we have  $\exists X.V(\beta)$  valid (then the fact that  $\text{HORN-C}[\exists]$  satisfies **VA** completes the proof.) Towards a contradiction.
- **CE, ME:** Direct from item 2. of Proposition 4 from [Fargier and Marquis, 2008a] given the fact that each fragment under consideration satisfies **CO** and **CD**, and contains only proper formulae. Indeed, a formula  $\alpha$  from QDAG implies a clause  $\gamma$  iff the conditioning of  $\alpha$  by the term corresponding to  $\neg\gamma$  is inconsistent.

• **IM:**

- $\text{renH-C}[V, \exists]$ ,  $\text{K/H-C}[V, \exists]$  and  $\text{HORN-C}[V, \exists]$ : By reduction from the validity problem for those fragments. Let  $\alpha$  be any formula from one of those fragments.  $\alpha$  is valid iff it is implied by the term *new* where  $\text{new} \in \text{PS} \setminus \text{Var}(\alpha)$ , which belongs to each of the three fragments under consideration here.
- $\text{renH-C}[\exists]$ ,  $\text{K/H-C}[\exists]$  and  $\text{HORN-C}[\exists]$ : Since each of  $\text{K/H-C}[\exists]$  and  $\text{HORN-C}[\exists]$  is polynomially translatable into  $\text{renH-C}[\exists]$ , it is enough to prove the result for  $\text{renH-C}[\exists]$ . Since  $\text{KROM-C}[\exists] \sim_p \text{KROM-C}$ , and since  $\text{KROM-C}$  satisfies **IM** [Fargier and Marquis, 2008b], we just have to care for the  $\text{renH-C}[\exists]$  formulae (including  $\text{HORN-C}[\exists]$  formulae as a proper subset.) Let us first show that considering  $\text{HORN-C}[\exists]$  formulae is enough. Let  $\exists X.\alpha$  be a  $\text{renH-C}[\exists]$  formula such that  $\alpha$  is a  $\text{renH-C}$  formula, and let  $\gamma$  be a term. Let  $V$  be any Horn renaming for  $\alpha$ . We have  $\gamma \models \exists X.\alpha$  iff  $\gamma \Rightarrow (\exists X.\alpha)$  is valid. Now, the substitution metatheorem shows that if  $\gamma \Rightarrow (\exists X.\alpha)$  is valid then  $V(\gamma \Rightarrow (\exists X.\alpha))$  is valid. Since for every formula  $\beta$ ,  $V(V(\beta)) \equiv \beta$ , we also have that if  $V(\gamma \Rightarrow (\exists X.\alpha))$  is valid, then  $\gamma \Rightarrow \exists X.\alpha$  is valid. Altogether, we get that  $\gamma \models \exists X.\alpha$  iff  $V(\gamma \Rightarrow (\exists X.\alpha))$  is valid. Now,  $V(\gamma \Rightarrow (\exists X.\alpha))$  is valid iff  $V(\gamma) \Rightarrow V(\exists X.\alpha)$  is valid iff  $V(\gamma) \models V(\exists X.\alpha)$ . Since for every  $x \in \text{Var}(\alpha)$ ,  $V(x)$  is a literal, we have  $V(\exists X.\alpha) \equiv \exists X.V(\alpha)$ . Accordingly,  $\gamma$  is an implicant of the  $\text{renH-C}[\exists]$  formula  $\exists X.\alpha$  iff the term  $V(\gamma)$  is an implicant of the  $\text{HORN-C}[\exists]$  formula  $\exists X.V(\alpha)$ .

Let  $\exists X.\alpha$  be a  $\text{renH-C}[\exists]$  formula such that  $\alpha = \bigwedge_{i=1}^k \delta_i$  is a  $\text{renH-C}$  formula, and let  $\gamma = l_1 \wedge \dots \wedge l_p$  be a term. Let  $V$  be any Horn renaming for  $\alpha$ . We have  $\gamma \models \exists X.\alpha$  iff  $\gamma \Rightarrow (\exists X.\alpha)$  is valid iff  $\exists X'.(\gamma \Rightarrow \alpha')$  is valid, where  $\alpha'$  is obtained by renaming every variable  $x \in X$  occurring in  $\alpha$  by a fresh variable  $x'$ , and  $X'$  is the set of all fresh variables  $x'$  introduced when  $\alpha'$  is generated (just note that  $\gamma$  does not contain any variable from  $X'$ .) Let  $V' = (V \setminus X) \cup \{x' \mid x \in V \cap X\}$ ; obviously enough,  $V'$  is a Horn renaming for  $\alpha'$ . Let now  $S$  be the set of clauses  $\delta'_i$  of  $\alpha'$  such that no literal  $l_j$  ( $j \in 1, \dots, p$ ) of  $\gamma$  occurs in  $\delta'_i$ . Then,  $\exists X'.(\gamma \Rightarrow \alpha')$  is equivalent to  $\beta = \exists X'.(\bigwedge_{\delta'_i \in S} \bar{l}_1 \vee \dots \vee \bar{l}_p \vee \delta'_i)$ . Clearly, the size of  $\beta$  is polynomial in the size of  $\alpha$  plus the size of  $\gamma$ . Observe also that, by construction, every variable of  $\text{Var}(\gamma)$  is pure in this CNF $[\exists]$  formula  $\beta$ . As a consequence,  $W = V' \cup \{\text{var}(l_j) \mid l_j \text{ is a negative literal of } \gamma\}$  is a Horn renaming of  $\bigwedge_{\delta'_i \in S} \bar{l}_1 \vee \dots \vee \bar{l}_p \vee \delta'_i$ . This shows that  $\beta$  is a  $\text{renH-C}[\exists]$  formula which is valid iff  $\gamma$  is an implicant of  $\alpha$ .

We now show that the problem of deciding whether a  $\text{renH-C}[\exists]$  formula is valid can be reduced

in polynomial time to the problem of deciding whether a  $\text{HORN-C}[\exists]$  formula is valid. Let  $\exists X.\alpha$  be a  $\text{renH-C}[\exists]$  formula such that  $\alpha$  is a  $\text{renH-C}$  formula. Let  $V$  be any Horn renaming for  $\alpha$ . The substitution metatheorem for propositional logic shows that if  $\exists X.\alpha$  is valid then  $V(\exists X.\alpha)$  is valid. Since for every formula  $\beta$ ,  $V(V(\beta)) \equiv \beta$ , we also have that if  $V(\exists X.\alpha)$  is valid, then  $\exists X.\alpha$  is valid. Altogether, we get that  $\exists X.\alpha$  is valid iff  $V(\exists X.\alpha)$  is valid. Since for every  $x \in \text{Var}(\alpha)$ ,  $V(x)$  is a literal, we have  $V(\exists X.V(\alpha)) \equiv \exists X.V(\alpha)$ . Accordingly,  $\exists X.\alpha$  is valid iff the  $\text{HORN-C}[\exists]$  formula  $\exists X.V(\alpha)$  is valid iff  $\forall \text{Var}(\alpha) \setminus X \exists X.V(\alpha)$  is valid.

The fact that the validity problem for closed quantified Boolean formulae in prenex normal form with a Horn matrix can be decided in polynomial time concludes the proof.

- **EQ:**

- $\text{renH-C}[\forall, \exists]$ ,  $\text{K/H-C}[\forall, \exists]$  and  $\text{HORN-C}[\forall, \exists]$ : By reduction from the validity problem for those fragments. Let  $\alpha$  be any formula from one of those fragments.  $\alpha$  is valid iff it is equivalent to a valid clause, like  $a \vee \neg a$  which belongs to each of the three fragments under consideration here.
- $\text{renH-C}[\exists]$ ,  $\text{K/H-C}[\exists]$  and  $\text{HORN-C}[\exists]$ : For every propositional formulae  $\alpha'$  and  $\gamma'$  we have that  $\alpha' \models \gamma'$  iff  $\alpha' \wedge \gamma' \equiv \alpha'$ . Consider now the formulae  $\alpha'$  and  $\gamma'$  used for proving that none of  $\text{renH-C}[\exists]$ ,  $\text{K/H-C}[\exists]$  and  $\text{HORN-C}[\exists]$  satisfies **SE** unless  $\text{P} = \text{NP}$  (see the item **SE** in this proof.) Since none of the  $\text{new}_i$  variables occurs in  $\alpha'$ , the formula  $\alpha' \wedge \gamma'$  can be turned in linear time into the equivalent formula  $\exists \{ \text{new}_1, \dots, \text{new}_n \} \alpha' \wedge ((\neg \text{new}_1 \vee \dots \vee \neg \text{new}_n) \wedge \bigwedge_{i=1}^n ((\text{new}_i \vee \neg x_i) \wedge (\text{new}_i \vee \neg \text{not} - x_i)))$ , which is a  $\text{HORN-C}[\exists]$  formula. This concludes the proof.

- **SE:**

- $\text{renH-C}[\forall, \exists]$ ,  $\text{K/H-C}[\forall, \exists]$  and  $\text{HORN-C}[\forall, \exists]$ : By reduction from the validity problem for those fragments. Let  $\alpha$  be any formula from one of those fragments.  $\alpha$  is valid iff it is entailed by a valid clause, like  $a \vee \neg a$  which belongs to each of the three fragments under consideration here.
- $\text{renH-C}[\exists]$ ,  $\text{K/H-C}[\exists]$  and  $\text{HORN-C}[\exists]$ : It is enough to prove the result for  $\text{HORN-C}[\exists]$  since this fragment is included in the two remaining ones. Let  $\alpha$  be a CNF formula over  $n$  variables  $x_1, \dots, x_n$ . Let  $\alpha'$  be the  $\text{HORN-C}$  formula obtained by replacing every positive literal  $x_i$  in  $\alpha$  by the negative literal  $\neg \text{not} - x_i$  (where each  $\text{not} - x_i$  is a fresh variable), conjoined with  $n$  additional clauses  $\neg x_i \vee \neg \text{not} - x_i$  ( $i \in 1, \dots, n$ ). Let  $\beta'$  be the  $\text{KROM-C}$  formula  $\bigwedge_{i=1}^n x_i \vee \text{not} - x_i$ . By construction,  $\alpha$  is consistent iff  $\alpha' \wedge \beta'$  is consistent iff  $\alpha' \not\models \neg \beta'$ .  $\neg \beta'$  is equivalent to  $\bigvee_{i=1}^n \neg x_i \wedge \neg \text{not} - x_i$ , which in turn is equivalent to the formula  $\gamma' =$

$$\exists \{ \text{new}_1, \dots, \text{new}_n \} ((\neg \text{new}_1 \vee \dots \vee \neg \text{new}_n) \wedge \bigwedge_{i=1}^n ((\text{new}_i \vee \neg x_i) \wedge (\text{new}_i \vee \neg \text{not} - x_i)))$$

(where each  $\text{new}_i$  is a fresh variable). The fact that  $\alpha'$  and  $\gamma'$  are  $\text{HORN-C}[\exists]$  formulae which can be computed in time polynomial in the size of  $\alpha$  shows the **coNP**-hardness of **SE** for  $\text{HORN-C}[\exists]$  formulae and concludes the proof.

- **CT:** Direct from the fact that  $\text{KROM-C}$  does not satisfy **CT** [Roth, 1996], given that  $\text{KROM-C}$  is polynomially translatable into each of the fragments under consideration.
- **MC:** Direct from item 4. of Proposition 4 from [Fargier and Marquis, 2008a] showing that when a subset  $\mathcal{L}$  of DAG satisfies **CO** and **CD**, then  $\mathcal{L}[\exists]$  and  $\mathcal{L}[\forall, \exists]$  satisfies **MC**, plus the fact that each of  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{renH-C}$  satisfies **CO** and **CD**.

As to the transformations:

- **CD:** Direct from item 1. of Proposition 4 from [Fargier and Marquis, 2008a] showing that when a subset  $\mathcal{L}$  of DAG satisfies **CD**, then  $\mathcal{L}[\exists]$  and  $\mathcal{L}[\forall, \exists]$  satisfies **CD** as well, plus the fact that each of  $\text{KROM-C}$ ,  $\text{HORN-C}$ ,  $\text{renH-C}$  satisfies **CD**.
- **FO:** Obvious since all the languages considered here are existential closures.
- **SFO:** Obvious since all the languages considered here satisfy **FO**.
- **ABC:**
  - $\text{renH-C}[\forall, \exists]$ ,  $\text{K/H-C}[\forall, \exists]$ ,  $\text{K/H-C}[\exists]$ : Direct from the fact that none of these three fragments satisfies **ABC**.
  - $\text{HORN-C}[\forall, \exists]$ : Direct from the fact that each clause is a  $\text{HORN-C}[\forall, \exists]$  formula, plus the fact that  $\text{HORN-C}[\forall, \exists]$  satisfies **CO**.
  - $\text{renH-C}[\exists]$ : Direct from the fact that the restricted transformation **ABC** is not always feasible within the fragment.
  - $\text{HORN-C}[\exists]$ : Let  $\exists X_1.\alpha_1, \dots, \exists X_n.\alpha_n$  be  $n$   $\text{HORN-C}[\exists]$  formulae where each  $\alpha_i$  ( $i \in 1, \dots, n$ ) is a  $\text{HORN-C}$  formula. For each  $i \in 1, \dots, n$ , let  $\alpha_i^i$  be the  $\text{HORN-C}$  formula obtained by replacing in  $\alpha_i$  every occurrence of  $x \in X_i$  by a fresh variable  $x^i$ , and let  $X_i^i$  be the set of all the variables  $x^i$  generated in the construction of  $\alpha_i$ . By construction, every variable from  $X_i^i$  does not occur in any  $\alpha_j^j$  when  $j \neq i$ . Hence,  $\bigwedge_{i=1}^n \exists X_i.\alpha_i$  is equivalent to  $\exists \bigcup_{i=1}^n X_i^i \bigwedge_{i=1}^n \alpha_i^i$ . Clearly enough,  $\exists \bigcup_{i=1}^n X_i^i \bigwedge_{i=1}^n \alpha_i^i$  is a  $\text{HORN-C}[\exists]$  formula, and it can be generated in polynomial time from  $\exists X_1.\alpha_1, \dots, \exists X_n.\alpha_n$ .
- **ABC:**
  - $\text{renH-C}[\forall, \exists]$ ,  $\text{K/H-C}[\forall, \exists]$ ,  $\text{K/H-C}[\exists]$ : Let  $\alpha$  be a CNF formula over  $n$  variables  $x_1, \dots, x_n$ . Let  $\alpha'$  be the  $\text{HORN-C}$  formula obtained by replacing every positive literal  $x_i$  in  $\alpha$  by the negative literal

$\neg not - x_i$  (where each  $not - x_i$  is a fresh variable), conjoined with  $n$  additional clauses  $\neg x_i \vee \neg not - x_i$  ( $i \in 1, \dots, n$ ). Let  $\beta'$  be the KROM-C formula  $\bigwedge_{i=1}^n x_i \vee not - x_i$ . Observe that  $\beta'$  is consistent, hence each of  $\alpha'$  and  $\beta'$  is a  $renH-C[V, \exists]$  formula and a  $K/H-C[\exists]$  formula. By construction,  $\alpha$  is consistent iff  $\alpha' \wedge \beta'$  is consistent. If any of  $renH-C[V, \exists]$ ,  $K/H-C[V, \exists]$ ,  $K/H-C[\exists]$  would satisfy  $\wedge BC$ , since each of these fragments satisfy **CO**, we would have  $P = NP$ .

- $HORN-C[V, \exists]$ : Let  $\alpha$  and  $\beta$  be two  $HORN-C[V, \exists]$  formulae. From item 1. of Proposition 2 from [Fargier and Marquis, 2008a], one can compute in polynomial time a  $HORN-C[V][\exists]$  formula  $\exists X.\alpha'$  (resp.  $\exists Y.\beta'$ ) equivalent to  $\alpha$  (resp.  $\beta$ ) where  $\alpha'$  (resp.  $\beta'$ ) is a  $HORN-C[V]$  formula. Let  $\alpha''$  (resp.  $\beta''$ ) be the  $HORN-C[V]$  formula obtained by replacing in  $\alpha'$  (resp.  $\beta'$ ) every occurrence of  $x \in X$  (resp.  $x \in Y$ ) by a fresh variable  $x'$  and let  $X'$  (resp.  $Y'$ ) be the set of all variables  $x'$  generated in the construction of  $\alpha''$  (resp.  $\beta''$ ). By construction  $\alpha \wedge \beta$  is equivalent to  $(\exists X.\alpha') \wedge (\exists Y.\beta')$ , which is in turn equivalent to  $\exists X' \cup Y'.(\alpha'' \wedge \beta'')$ . Now,  $HORN-C[V]$  satisfies  $\wedge BC$  (cf. Proposition 2 from [Fargier and Marquis, 2008b]). Hence, a  $HORN-C[V]$  formula  $\gamma$  equivalent to  $\alpha'' \wedge \beta''$  can be generated in time polynomial in the size of  $\alpha''$  plus the size of  $\beta''$ . Accordingly,  $\exists X' \cup Y'.\gamma$  is a  $HORN-C[V, \exists]$  formula equivalent to  $\alpha \wedge \beta$ , and it can be computed in time polynomial in the size of  $\alpha$  plus the size of  $\beta$ .
- $renH-C[\exists]$ :  $a \vee b \vee c$  and  $\neg a \vee \neg b \vee \neg c$  are two  $renH-C[\exists]$  formulae, but there does not exist a  $renH-C[\exists]$  formula equivalent to their conjunction, since  $(a \vee b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$  is not equivalent to a Horn renamable formula and  $renH-C[\exists] \sim_e renH-C$ .
- $HORN-C[\exists]$ : Direct from the fact that  $HORN-C[\exists]$  satisfies  $\wedge C$ .

•  $\vee C$ :

- $renH-C[V, \exists]$ ,  $K/H-C[V, \exists]$  and  $HORN-C[V, \exists]$ : Obvious since all the languages considered here are disjunction closures.
- $renH-C[\exists]$ ,  $K/H-C[\exists]$  and  $HORN-C[\exists]$ : Direct from the fact that  $\vee BC$  is not always feasible within the fragment.

•  $\vee BC$ :

- $renH-C[V, \exists]$ ,  $K/H-C[V, \exists]$  and  $HORN-C[V, \exists]$ : Obvious since all languages considered here satisfy  $\vee C$ .
- $renH-C[\exists]$ : Consider the two  $renH-C[\exists]$  formulae  $\alpha = (a \vee b) \wedge (\neg b \vee \neg c)$  and  $\beta = \neg a \wedge c$ .  $\alpha \vee \beta$  is equivalent to  $(a \vee b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$ , which is not equivalent to a  $renH-C$  formula, hence not to a  $renH-C[\exists]$  formula since  $renH-C[\exists] \sim_e renH-C$ .

- $K/H-C[\exists]$  and  $HORN-C[\exists]$ :  $\neg a \vee b$  and  $c$  are both  $KROM-C[\exists]$  and  $HORN-C[\exists]$  formulae but their disjunction  $\neg a \vee b \vee c$  neither is equivalent to a  $KROM-C[\exists]$  formula nor to a  $HORN-C[\exists]$  formula since it is neither a  $KROM-C$  formula nor a  $HORN-C$  none and we have  $KROM-C[\exists] \sim_e KROM-C$  and  $HORN-C[\exists] \sim_e HORN-C$ .

•  $\neg C$ :

- $renH-C[V, \exists]$ ,  $K/H-C[V, \exists]$  and  $HORN-C[V, \exists]$ : DNF is a subset of each of these fragments. Now, a DNF formula  $\alpha$  is valid iff  $\neg \alpha$  is inconsistent. The fact that each of  $renH-C[V, \exists]$ ,  $K/H-C[V, \exists]$  and  $HORN-C[V, \exists]$  satisfies **CO** completes the proof.
- $renH-C[\exists]$ ,  $K/H-C[\exists]$  and  $HORN-C[\exists]$ : Consider the consistent  $KROM-C$  formula  $\alpha = (\neg a \vee b) \wedge (\neg a \vee c) \wedge (a \vee \neg b) \wedge (\neg b \vee c) \wedge (a \vee \neg c) \wedge (b \vee \neg b)$ . It belongs to each of the fragments above.  $\neg \alpha$  is equivalent to  $(a \vee b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$ , which is not equivalent to a  $renH-C$  formula, hence not a  $renH-C[\exists]$  formula since  $renH-C[\exists] \sim_e renH-C$ . As such, it is not equivalent to a  $K/H-C[\exists]$  formula or a  $HORN-C[\exists]$  formula. ■

**Proof:**[Proposition 2] Since  $\mathcal{L}$  is a subset of  $\mathcal{L}[\exists]$ , we immediately get that  $\mathcal{L}[\exists] \leq_e \mathcal{L}$ . Since each  $\mathcal{L}$  among  $HORN-C[V]$ ,  $K/H-C[V]$ ,  $renH-C[V]$  is complete for propositional logic (as a superset of DNF), we have that  $\mathcal{L}[\exists] \sim_e \mathcal{L}$ . It remains to show that every  $HORN-C[\exists]$  (resp.  $K/H-C[\exists]$ ,  $renH-C[\exists]$ ) formula is equivalent to a  $HORN-C$  (resp.  $K/H-C$ ,  $renH-C$ ) formula. It is well-known that the prime implicates of a  $KROM-C$  (resp.  $HORN-C$ ) formula are Krom (resp. Horn) clauses. Since a formula equivalent to  $\exists X.\alpha$  consists of the conjunction of all prime implicates of  $\alpha$  not containing any symbol from  $X$  [Lang *et al.*, 2003], we get that  $KROM-C[\exists] \sim_e KROM-C$ , and  $HORN-C[\exists] \sim_e HORN-C$ . Consequently, we also have  $K/H-C[\exists] \sim_e K/H-C$ . Finally, we have to show that when  $\alpha$  is a  $renH-C$  formula,  $\exists X.\alpha$  is equivalent to a  $renH-C$  formula. Let  $V$  be a Horn renaming for  $\alpha$  and let  $\omega_1, \dots, \omega_n$  be the models of  $\alpha$  viewed as terms (so that we can write  $\alpha \equiv \bigvee_{i=1}^n \omega_i$ ). Since  $V$  can be viewed as a substitution, from the substitution theorem for propositional logic, we have that  $V(\alpha) \equiv V(\bigvee_{i=1}^n \omega_i) \equiv \bigvee_{i=1}^n V(\omega_i)$ . Since existential quantifications "distributes over" disjunctions,  $\exists X.V(\alpha)$  is equivalent to  $\bigvee_{i=1}^n \exists X.V(\omega_i)$ . Now, each  $\exists X.V(\omega_i)$  (for  $i \in 1, \dots, n$ ) is equivalent to the term obtained by removing in  $V(\omega_i)$  every literal over  $X$ . Since  $V$  is a substitution of the form  $v \mapsto \neg v$  for  $v \in V$ , each  $\exists X.V(\omega_i)$  (for  $i \in 1, \dots, n$ ) is also equivalent to  $V(\omega_i^{\bar{L}X})$  where  $\omega_i^{\bar{L}X}$  is the term obtained by removing in  $\omega_i$  every literal over  $X$ . Thus,  $\exists X.V(\alpha)$  is equivalent to  $\bigvee_{i=1}^n V(\omega_i^{\bar{L}X})$ , which is in turn equivalent to  $V(\bigvee_{i=1}^n \omega_i^{\bar{L}X})$ , which is equivalent to  $V(\exists X.\alpha)$ . Since  $V(\alpha)$  is a  $HORN-C$  formula,  $\exists X.V(\alpha)$  is equivalent to a  $HORN-C$  formula (as explained above). Hence,  $V(\exists X.\alpha)$  is equivalent to a  $HORN-C$  formula. Since  $V$  is a Horn renaming,  $\exists X.\alpha$  is equivalent to a  $renH-C$  formula. ■

**Proof:**[Proposition 3] Our proof is based on a theorem due to Sipser [Sipser, 1983]. This theorem can be expressed as follows: consider any Boolean function  $\alpha_k^n$  over  $n^{2k-2}$  variables, represented by a NNF formula having the shape of a tree of depth  $k > 1$  and such that all the leaves are labeled by variables occurring once in the formula, the  $i^{th}$  level ( $i \in 1, \dots, k-1$ ) from the bottom consists of nodes labeled by  $\wedge$  (resp.  $\vee$ ) when  $i$  is even (resp. odd), the outdegree of the root node and the deepest internal nodes (those at depth  $k-1$ ) is equal to  $n > 1$  and the outdegree of every other internal node is equal to  $n^2$ . Sipser showed that such an  $\alpha_k^n$  and its dual  $\alpha_k^{*n}$  (whose NNF representation can be obtained by replacing in the NNF representation of  $\alpha_k^n$  every  $\wedge$  by  $\vee$  and vice-versa) cannot be represented by a polynomial-sized circuit over  $\{\neg, \vee, \wedge\}$  of depth at most  $k-1$ . In this construction, the variables of  $\alpha_k^n$  (or its dual) can be negated or not. Obviously, by considering  $\neg\alpha_k^n$  and simplifying, a similar conclusion can be drawn for formulae  $\beta_k^n$  over  $n^{2k-2}$  variables, having the shape of a regular tree of depth  $k > 1$  and such that all the leaves are labeled by negated variables, the  $i^{th}$  level ( $i \in 1, \dots, k-1$ ) from the bottom consists of nodes labeled by  $\vee$  (resp.  $\wedge$ ) when  $i$  is even (resp. odd), the outdegree of the root node and the deepest internal nodes (those at depth  $k-1$ ) is equal to  $n > 1$  and the outdegree of every other internal node is equal to  $n^2$ .

- Consider the Boolean function  $\alpha_3^{*n}$  over  $n^4$  negated variables. By construction, it can be represented by a conjunction of  $n$  DNF formulae  $\beta_1, \dots, \beta_n$ , where each  $\beta_i$  ( $i \in 1, \dots, n$ ) consists of  $n^2$  terms  $\gamma_{i,j}$  ( $j \in 1, \dots, n^2$ ), and each term  $\gamma_{i,j}$  ( $j \in 1, \dots, n^2$ ) consists of the conjunction of  $n$  negated variables  $\neg x_{i,j,k}$  ( $k \in 1, \dots, n$ ) occurring only once in  $\alpha_3^{*n}$ . For each  $i \in 1, \dots, n$ , consider now the HORN-C formula  $h_i$  such that  $h_i =$

$$\left( \bigvee_{j=1}^{n^2} \neg y_{i,j} \right) \wedge \bigwedge_{j=1}^{n^2} \bigwedge_{k=1}^n (y_{i,j} \vee \neg x_{i,j,k}).$$

$h_i$  contains  $n^3 + 1$  clauses of size at most  $n^2$ , hence  $\bigwedge_{i=1}^n h_i$  contains  $n^4 + n$  clauses of size at most  $n^2$ . Let  $Y = \bigcup_{i=1}^n (\bigcup_{j=1}^{n^2} \{y_{i,j}\})$ . By construction, the HORN-C[ $\exists$ ] formula  $\exists Y. \bigwedge_{i=1}^n h_i$  can be generated in time polynomial in  $n$ . By definition, this formula  $\exists Y. \bigwedge_{i=1}^n h_i$  also is a K/H-C[ $\exists$ ] formula and a renH-C[ $\exists$ ] formula. From Sipser theorem,  $\alpha_3^{*n}$  has no polynomial-sized CNF representation and no polynomial-sized DNF representation. It remains to show that  $\alpha_3^{*n}$  is equivalent to  $\exists Y. \bigwedge_{i=1}^n h_i$ . First of all, since each  $y_{i,j}$  ( $i \in 1, \dots, n, j \in 1, \dots, n^2$ ) does not occur in  $h_{i'}$  ( $i' \in 1, \dots, n$ ) unless  $i' = i$ , we have that  $\exists Y. \bigwedge_{i=1}^n h_i$  is equivalent to  $\bigwedge_{i=1}^n (\exists \bigcup_{j=1}^{n^2} \{y_{i,j}\}. h_i)$ . Finally, by construction, for each  $i \in 1, \dots, n$ ,  $\beta_i = \bigvee_{j=1}^{n^2} \gamma_{i,j}$  is the IP representation of  $\exists \bigcup_{j=1}^{n^2} \{y_{i,j}\}. h_i$ , hence it is equivalent to it. The substitution theorem for propositional logic concludes the proof.

- The proof is similar to the one given in item 1., but lifted to a greater depth. Consider the Boolean function

$\alpha_4^n$  over  $n^6$  variables. By construction, it can be represented by a disjunction of  $n$  conjunctions  $\beta_1, \dots, \beta_n$  of DNF formulae, where each  $\beta_i$  ( $i \in 1, \dots, n$ ) is the conjunction of  $n^2$  DNF  $\gamma_{i,j}$  ( $j \in 1, \dots, n^2$ ), each DNF  $\gamma_{i,j}$  ( $j \in 1, \dots, n^2$ ) consists of the disjunction of  $n^2$  terms  $\delta_{i,j,k}$  ( $k \in 1, \dots, n^2$ ), and finally each term  $\delta_{i,j,k}$  ( $k \in 1, \dots, n^2$ ) consists of the conjunction of  $n$  negated variables  $\neg x_{i,j,k,l}$  ( $l \in 1, \dots, n$ ) occurring only once in  $\alpha_4^n$ . For each  $i \in 1, \dots, n$  and  $j \in 1, \dots, n^2$ , consider now the HORN-C formula  $h_{i,j}$  such that  $h_{i,j} =$

$$\left( \bigvee_{k=1}^{n^2} \neg y_{i,j,k} \right) \wedge \bigwedge_{k=1}^{n^2} \bigwedge_{l=1}^n (y_{i,j,k} \vee \neg x_{i,j,k,l}).$$

$h_{i,j}$  contains  $n^3 + 1$  clauses of size at most  $n^2$ , hence the HORN-C formula  $\bigwedge_{j=1}^{n^2} h_{i,j}$  contains  $n^5 + n^2$  clauses of size at most  $n^2$ . Let  $Y = \bigcup_{i=1}^n (\bigcup_{j=1}^{n^2} (\bigcup_{k=1}^{n^2} \{y_{i,j,k}\}))$ . By construction, the HORN-C[ $\forall, \exists$ ] formula  $\exists Y. (\bigvee_{i=1}^n (\bigwedge_{j=1}^{n^2} h_{i,j}))$  can be generated in time polynomial in  $n$ . By definition, this formula  $\exists Y. (\bigvee_{i=1}^n (\bigwedge_{j=1}^{n^2} h_{i,j}))$  also is a K/H-C[ $\forall, \exists$ ] formula and a renH-C[ $\forall, \exists$ ] formula. From Sipser theorem,  $\alpha_4^n$  has no polynomial-sized AC<sup>3</sup> representation. It remains to show that  $\alpha_4^n$  is equivalent to  $\exists Y. (\bigvee_{i=1}^n (\bigwedge_{j=1}^{n^2} h_{i,j}))$ . First of all, since existential quantifications "distribute" over disjunctions and since each  $y_{i,j,k}$  ( $i \in 1, \dots, n, j \in 1, \dots, n^2, k \in 1, \dots, n^2$ ) does not occur in  $h_{i',j'}$  ( $i' \in 1, \dots, n, j' \in 1, \dots, n^2$ ) unless  $i' = i$  and  $j' = j$ , we have that  $\exists Y. (\bigvee_{i=1}^n (\bigwedge_{j=1}^{n^2} h_{i,j}))$  is equivalent to  $\bigvee_{i=1}^n (\bigwedge_{j=1}^{n^2} \exists \bigcup_{k=1}^{n^2} \{y_{i,j,k}\}. h_{i,j})$ . Finally, by construction, for each  $i \in 1, \dots, n$  and  $j \in 1, \dots, n^2$ ,  $\gamma_{i,j} = \bigvee_{k=1}^{n^2} \delta_{i,j,k}$  is the IP representation of  $\exists \bigcup_{k=1}^{n^2} \{y_{i,j,k}\}. h_{i,j}$ , hence it is equivalent to it. The substitution theorem for propositional logic concludes the proof. ■

**Proof:**[Proposition 4] The result comes easily from the fact that d-DNNF is not at least as succinct as DNF, unless the polynomial hierarchy collapses [Darwiche and Marquis, 2002], plus the fact that DNF is polynomial translatable into each of KROM-C[ $\forall$ ], HORN-C[ $\forall$ ], K/H-C[ $\forall$ ], renH-C[ $\forall$ ], AFF[ $\forall$ ], PI[ $\forall$ ] and OBDD<sub><</sub>[ $\forall$ ] (observe that a term is a KROM-C formula, a HORN-C formula, a K/H-C formula, a renH-C formula, an AFF formula, and a PI formula, and that it can be turned in linear time into an OBDD<sub><</sub> formula whatever the underlying ordering <), and finally that any existential closure of a propositional language  $\mathcal{L} \subseteq \text{QDAG}$  is at least as succinct as  $\mathcal{L}$ . ■

**Proof:**[Proposition 5] Since PI  $\subseteq$  CNF and since any disjunctive closure of a language  $\mathcal{L}$  is at least as succinct as  $\mathcal{L}$ , it is enough to show that the family of cir-

cular bit shift functions have polynomial-sized representations in KROM-C, HORN-C, K/H-C, PI, and AFF. Indeed, it has been proven that such functions do not have polynomial-sized SDNNF representations [Pipatsrisawat and Darwiche, 2010a]. For any positive integer  $m$ , consider the following Boolean function over  $2^{m+1} + m$  variables  $cbS_m(x_1, \dots, x_{2^m}, y_1, \dots, y_{2^m}, i_1, \dots, i_m)$  which is the semantics of the CNF formula  $\alpha_m =$

$$\bigwedge_{i=1}^{2^m} (\neg x_i \vee y_{i+num(i_1, \dots, i_m) \bmod 2^m}) \wedge (x_i \vee \neg y_{i+num(i_1, \dots, i_m) \bmod 2^m}).$$

In this formula,  $num$  is the function from  $\{0, 1\}^m$  to the set of natural numbers which gives the integer represented by the binary string  $i_1 \dots i_m$ . Thus, the variables  $i_1, \dots, i_m$  make precise how many bits of the binary string  $x_1 \dots x_{2^m}$  must be (circularly) shifted, and  $cbS_m(x_1, \dots, x_{2^m}, y_1, \dots, y_{2^m}, i_1, \dots, i_m) = 1$  exactly when the shifted variables  $x_1, \dots, x_{2^m}$  and the variables  $y_1, \dots, y_{2^m}$  are pairwise equal.

Clearly enough,  $\alpha_m$  is a KROM-C formula and a HORN-C formula. Since  $\text{KROM-C} \subseteq \text{K/H-C}$ ,  $\text{HORN-C} \subseteq \text{renH-C}$ , and KROM-C is polynomially translatable into PI,  $\alpha_m$  is polynomially translatable into a K/H-C formula, a renH-C formula, and a PI formula. Finally, as to AFF, it is enough to observe that  $\alpha_m$  can be turned in linear time into the equivalent AFF formula  $\bigwedge_{i=1}^{2^m} x_i \oplus y_{i+num(i_1, \dots, i_m) \bmod 2^m} \oplus \top$ . ■