Extending the Knowledge Compilation Map: Krom, Horn, Affine and Beyond

Hélène Fargier
IRIT-CNRS, Université Paul Sabatier,
Toulouse, France
fargier@irit.fr

Pierre Marquis
Université Lille-Nord de France, Artois
CRIL UMR CNRS 8188, Lens, France
marquis@cril.univ-ortois.fr

Abstract
We extend the knowledge compilation map introduced by Darwiche and Marquis with three influential propositional fragments, the Krom CNF one (also known as the bijunctive fragment), the Horn CNF fragment and the affine fragment (also known as the biconditional fragment) as well as seven additional languages based on them, and composed respectively of Krom or Horn CNF formulas, renamable Horn CNF formulas, disjunctions of Horn CNF formulas, disjunctions of Krom or Horn CNF formulas, disjunctions of renamable Horn CNF formulas, and disjunction of affine formulas. Each fragment is evaluated w.r.t. several criteria, including the complexity of basic queries and transformations, and its spatial efficiency is also analyzed.

Introduction
Knowledge compilation (KC) is considered in many AI applications where short on-line response times are expected. It consists in turning (during an off-line phase) the initial data into a form that ensures the tractability of the requests and transformations of interest (see among others (Darwiche 2001; Cadoli & Donini 1998; Boufkhad et al. 1997; Selman & Kautz 1996; Schrag 1996; Gogic et al. 1995; Marquis 1995; del Val 1994; Dechter & Rish 1994)).

Many languages can be considered as target languages for knowledge compilation. In the propositional case, (Darwiche & Marquis 2002) investigate a dozen such languages, called propositional fragments. In this paper, the authors argue that the choice of a target language for a compilation purpose must be based both on the spatial efficiency of the language (i.e., its ability to represent data using little space) as well as its temporal efficiency, i.e., its ability to enable a set of queries and transformations to be achieved in polynomial time. The basic queries considered in (Darwiche & Marquis 2002) include tests for consistency, validity, implicates (clausal entailment), implicants, equivalence, sentential entailment, counting andEnumerating theory models (CO, VA, CE, EQ, SE, IM, CT, ME). The basic transformations are conditioning (CD), closures under the connectives (\(\land, \lor, \neg\)) (\(\land C, \land BC, \lor C, \lor BC, \neg C\)), and forgetting which can be viewed as a closure operation under existential quantification (FO, SFO).

The KC map and its extensions (Wachter & Haenni 2006; Fargier & Marquis 2006; Subbarayan, Bordeaux, & Hamadi 2007) have put forward the interest of fragments satisfying the property of decomposability, especially DNNF (Darwiche 2001). Indeed this fragment enables CO, CE, CD and FO in polynomial time, while being more succinct than DNF and OBDD. On the other hand, the property of decomposability is seldom compatible with a polynomial handling of the conjunctive transformations \(\land C\) or even \(\land BC\); thus DNNF does not enable \(\land BC\) in polytime unless \(P = NP\).

In this paper, we focus on ten propositional fragments which have not been considered in the above-mentioned papers. We first consider the Krom CNF fragment KROM-\(C\) (also known as the bijunctive fragment), the Horn CNF fragment HORN-\(C\), and the affine fragment AFF (also known as the biconditional fragment) (Schafer 1978), as well as K/H-C (Krom or Horn CNF formulas) and renH-C, the class of renamable Horn CNF formulas. None of these first five fragments is fully expressive w.r.t. propositional logic (there exist propositional formulas which cannot be represented in any of them). But full expressiveness can be recovered by considering disjunctions of such formulas; we thus include in our investigation the following fragments, which are complete for propositional logic: KROM-\([C]\), HORN-\([C]\), K/H-\([C]\), renH-\([C]\), and AFF\([C]\) are composed respectively of disjunctions of Krom CNF formulas, disjunctions of Horn CNF formulas, disjunctions of Krom or Horn CNF formulas, disjunctions of renamable Horn CNF formulas, and disjunctions of affine formulas. Interestingly, each of these classes enables CE in polynomial time, just like DNNF does; furthermore, it has been shown that, from the practical side, for some propositional formulas the size of renH-\([C]\) compilations can be much smaller than the size of DNNF compilations (Boufkhad et al. 1997).

The contribution of this paper consists of an evaluation of KROM-\(C\), HORN-\(C\), K/H-\(C\), renH-\(C\), AFF, KROM-\([C]\), HORN-\([C]\), K/H-\([C]\), renH-\([C]\), and AFF\([C]\) following the lines of (Darwiche & Marquis 2002). While (Boufkhad et al. 1997) considers only the clausal entailment issue, all the queries and transformations considered in (Darwiche & Marquis 2002) are investigated here for the
ten fragments and the spatial efficiency of those fragments is also analyzed. Among other things, our results show that, when $\wedge BC$ is expected, $\text{HORN-NF}[\lor], \text{KROM-NF}[\lor]$ and $\text{AFF}[\lor]$ are very interesting alternatives to DNNF, which does not satisfy it. The $\wedge BC$ transformation is of the utmost value in a number of applications; for instance, it offers the opportunity of incrementally compiling devices for the diagnosis issue: when a device is composed of a small number of components, each component can be compiled separately, and connecting them amounts mainly to conjoin the corresponding compiled forms, which can be done efficiently when a compilation fragment satisfying $\wedge BC$ is targeted.

The paper is organized as follows. After some formal preliminaries, we recall the languages, queries, transformations and the notion of succinctness considered in the KC map. We then present our results concerning the evaluation of the languages $\text{KROM-C}, \text{HORN-C}, \text{K/H-C}, \text{enH-C}, \text{AFF}, \text{KROM-NF}[\lor], \text{K/H-C}[\lor], \text{enH-NF}[\lor]$ and $\text{AFF}[\lor]$. Those results are discussed just before the concluding section. For space reasons, proofs are omitted.

Formal Preliminaries

We assume the reader familiar with the basics of propositional logic, including the notion of satisfaction, entailment (|=) and equivalence (=). All the propositional fragments we consider in this paper are subsets of the following propositional language $\text{DAG}_{PS}$:

**Definition 1** Let $PS$ be a set of propositional variables (or atoms). $\text{DAG}_{PS}$ is the set of all finite, single-rooted DAGs where each leaf node is labeled by a literal over $PS$ or one of the two Boolean constants $\top$ or $\bot$, and each internal node is labeled by $\land, \lor$ or $\oplus$ and has arbitrarily many children. The elements of $\text{DAG}_{PS}$ are called formulas.

The fragment $\text{DAG-NNF}_{PS}$, considered in (Darwiche & Marquis 2002) is the set of all $\text{DAG}_{PS}$ formulas in which the XOR connective $\oplus$ does not occur. For any formula $\alpha$, $\text{Var}(\alpha)$ denotes the set of atoms of $PS$ occurring in $\alpha$. The size $|\alpha|$ of $\alpha$ is the total number of vertices and arcs in its DAG representation.

Distinguished formulas are the literals over $PS$; for any subset $V$ of $PS$, $L_V$ denotes the set of all literals built over $V$, i.e., $\{ x, \neg x \mid x \in V \}$. If a literal $l$ of $L_{PS}$ is an atom $x$ from $PS$, it is said to be positive; otherwise it has the form $\neg x$ with $x \in PS$ and it is said to be negative. If $l$ is a positive literal $x$ then its complementary literal $\neg x$ is the negative literal $\neg x$; if $l$ is a negative literal $\neg x$ then its complementary literal $\neg x$ is the positive literal $x$. Other distinguished formulas are the clauses (resp. the terms) over $PS$; a clause (resp. a term) is a finite disjunction (resp. conjunction) of literals, or the Boolean constant $\bot$ (resp. $\top$). An XOR-clause is a finite exclusive disjunction of literals or Boolean constants.

The KC Map

Among the propositional fragments considered in (Darwiche & Marquis 2002) are $\text{OBDD}, \text{DNF}, \text{DNNF}, \text{CNF}, \text{PT}, \text{IP}$. Many fragments among them can be characterized by a number of properties, restricting the admissible formulas:

- **Flatness**: A $\text{DAG}_{PS}$ formula satisfies this property if its height is at most 2.
- **Simple-disjunction**: A $\text{DAG}_{PS}$ formula satisfies this property if the children of each or-node are leaves that share no variables (the node is a clause).
- **Simple-conjunction**: A $\text{DAG}_{PS}$ formula satisfies this property if the children of each and-node are leaves that share no variables (the node is a term).
- **Decomposability**: A $\text{DAG}_{PS}$ formula satisfies this property if for each conjunction $C$ in the formula, the conjuncts of $C$ do not share variables. That is, if $C_1, \ldots, C_n$ are the children of and-node $C$, then $\text{Var}(C_i) \cap \text{Var}(C_j) = \emptyset$ for $i \neq j$.
- **Decision**: A $\text{DAG}_{PS}$ formula satisfies this property if its root is a decision node, where a decision node $N$ in a $\text{DAG}_{PS}$ formula is one which is labeled with $T, \bot$, or is an or-node having the form $(x \land \alpha) \lor (\neg x \land \beta)$, where $x$ is a variable, $\alpha$ and $\beta$ are decision nodes. In the latter case, $d\text{Var}(N)$ denotes the variable $x$.
- **Ordering**: Let $\prec$ be a total ordering on $PS$. A $\text{DAG}_{PS}$ formula satisfying Decomposability and Decision satisfies Ordering if whenever $N$ and $M$ are or-nodes in it, if $N$ is an ancestor of node $M$, then $d\text{Var}(N) < d\text{Var}(M)$.

**Definition 2**

- **DNNF** is the subset of all $\text{DAG-NNF}_{PS}$ formulas satisfying Decomposability.
- **OBDD$<$** is the subset of all $\text{DAG-NNF}_{PS}$ formulas satisfying Decomposability, Decision and Ordering (for a given total ordering $< \text{on PS}$). $\text{OBDD}$ is the union of all $\text{OBDD} <$ languages.
- **CNF** is the subset of all $\text{DAG-NNF}_{PS}$ formulas satisfying Flatness and Simple-disjunction.
- **DNF** is the subset of all $\text{DAG-NNF}_{PS}$ formulas satisfying Flatness and Simple-conjunction.
- **PT** is the subset of $\text{CNF}$ in which each clause entailed by the formula is subsumed by a clause that appears in the formula; and no clause in the formula is subsumed by another.
- **IP** is the subset of $\text{DNF}$ in which each term entailing the formula subsumes some term that appears in the formula; and no term in the formula is subsumed by another term.

The following queries and transformations have been considered in (Darwiche & Marquis 2002); since their importance has been discussed in depth, we refrain from recalling it here.

**Definition 3** Let $C$ denote any subset of $\text{DAG}_{PS}$.

- $C$ satisfies $\text{CO}$ (resp. $\text{VA}$) iff there exists a polytime algorithm that maps every formula $\alpha$ from $C$ to 1 if $\alpha$ is consistent (resp. valid), and to 0 otherwise.
- $C$ satisfies $\text{CE}$ iff there exists a polytime algorithm that maps every formula $\alpha$ from $C$ and every clause $\gamma$ to 1 if $\alpha \models \gamma$ holds, and to 0 otherwise.
- $C$ satisfies $\text{EQ}$ (resp. $\text{SE}$) iff there exists a polytime algorithm that maps every pair of formulas $\alpha, \beta$ from $C$ to 1 if $\alpha \equiv \beta$ (resp. $\alpha \models \beta$) holds, and to 0 otherwise.
C satisfies IM iff there exists a polytime algorithm that maps every formula α from C and every term γ to 1 if γ |= α holds, and to 0 otherwise.

C satisfies CT iff there exists a polytime algorithm that maps every formula α from C to a nonnegative integer that represents the number of models of α over Var(α) (in binary notation).

C satisfies ME iff there exists a polynomial p(., .) and an algorithm that outputs all models of an arbitrary formula α from C in time p(n, m), where n is the size of α and m is the number of its models (over Var(α)).

Definition 4 Let C denote any subset of DAG

C satisfies CD iff there exists a polytime algorithm that maps every formula α from C and every consistent term γ to a formula from C that is logically equivalent to the conditioning α \mid γ of α on γ, i.e., the formula obtained by replacing every variable x of α by T (resp. ⊥) if x (resp. ¬x) is a positive (resp. negative) literal of γ.

C satisfies FO iff there exists a polytime algorithm that maps every formula α from C and every subset X of variables from P S to a formula from C equivalent to \exists X.α. C satisfies SFO if the property holds for any singleton.

C satisfies ∆C (resp. ∨C) iff there exists a polytime algorithm that maps every finite set of formulas α₁, . . . , αₙ from C to a formula of C that is logically equivalent to α₁ ∧ . . . ∧ αₙ (resp. α₁ ∨ . . . ∨ αₙ).

C satisfies ABC (resp. ∀BC) iff there exists a polytime algorithm that maps every pair of formulas α and β from C to a formula of C that is logically equivalent to α ∧ β (resp. α ∨ β).

C satisfies ¬C if there exists a polytime algorithm that maps every formula α from C to a formula of C that is logically equivalent to ¬α.

Finally, the following notion of succinctness has been considered in (Darwiche & Marquis 2002):

Definition 5 Let C₁ and C₂ be two subsets of DAG which are complete for propositional logic. C₁ is at least as succinct as C₂, denoted C₁ ≤ s C₂, if there exists a polynomial p such that for every formula α ∈ C₂, there exists an equivalent formula β ∈ C₁ where |β| ≤ p(|α|).

= is the symmetric part of ≤, defined by C₁ = C₂ iff C₁ ≤ C₂ and C₂ ≤ C₁. <ₚ is the asymmetric part of ≤, defined by C₁ <ₚ C₂ iff C₁ ≤ C₂ and C₂ ≤ C₁. The succinctness preorder ≤ can be viewed as a refinement of the expressiveness preorder over all propositional fragments, given by C₁ is at least as expressive as C₂, denoted C₁ ≤ e C₂, iff for every formula α ∈ C₂, there exists an equivalent formula β ∈ C₁. <ₑ is the asymmetric part of ≤ₑ.

Extending the KC Map

Languages

This paper focuses on the following subsets of DAG:

Definition 6

- The language KROM-C is the subset of all CNF formulas in which each clause is binary, i.e., it contains at most two literals.

- The language HORN-C is the subset of all CNF formulas in which each clause contains at most one positive literal.

- The language K/H-C is the union of KROM-C and HORN-C.

- The language renH-C is the subset of all CNF formulas α for which there exists a subset V of Var(α) (called a Horn renaming for α) such that the formula obtained by substituting in α every literal l of L₁ by its complementary literal l is a Horn CNF formula.

- The language AFF is the subset of DAG consisting of conjunctions of XOR clauses.

- The language KROM-C[

  V ] is the subset of DAG consisting of disjunctions of KROM-C formulas.

- The language HORN-C[

  V ] is the subset of DAG consisting of disjunctions of HORN-C formulas.

- The language K/H-C[

  V ] is the subset of DAG consisting of disjunctions of K/H-C formulas.

- The language renH-C[

  V ] is the subset of DAG consisting of disjunctions of renH-C formulas.

Example 1

- (a ∨ b) ∧ (¬b ∨ c) is a KROM-C formula.

- (¬a ∨ ¬b ∨ c) ∧ (¬b ∨ c ∨ ¬d) is a HORN-C formula.

- (a ∨ b ∨ c) ∧ ¬a ∨ ¬b ∨ c is a renH-C formula.

- (((a ∨ b) ∧ (¬b ∨ c)) ∨ a ∨ d) is a KROM-C[

  V ] formula.

- (((¬a ∨ c) ∧ (¬b ∨ c ∨ ¬d)) ∨ (a ∧ b)) is a HORN-C[

  V ] formula.

- ((a ∨ b) ∧ (¬b ∨ c ∨ ¬d)) ∨ ((a ∨ b) ∧ c) is a K/H-C[

  V ] formula.

- (((a ∨ c) ∧ (b ∨ c ∨ ¬d)) ∨ (a ∨ b) ∧ c) is a renH-C[

  V ] formula.

- (((a ∨ b) ∧ (¬b ∨ c)) ∨ (a ∨ b) ∧ c) is an AFF[

  V ] formula.

Obviously enough, we have the following inclusions:

HORN-C ⊆ K/H-C

HORN-C ⊆ HORN-C[

  V ] ⊆ K/H-C[

  V ]

HORN-C[

  V ] ⊆ renH-C[

  V ]

KROM-C ⊆ K/H-C

KROM-C ⊆ KROM-C[

  V ] ⊆ K/H-C[

  V ].

Note that there exists linear time algorithms for recognizing renamable Horn CNF formulas (see e.g. (Hébrard 1994; del Val 2000)); furthermore, such recognition algorithms typically give a Horn renaming when it exists. It is also known that every satisfiable KROM-C formula is a renH-C formula and that KRON-C satisfies CO; so, to every K/H-C[

  V ] formula we can associate in polynomial time an equivalent renH-C[

  V ] formula.

Expressiveness

It is well-known that none of the fragments KROM-C, HORN-C, K/H-C, renH-C, or AFF is complete for propositional logic. For instance, there is no formula from any of these fragments which is equivalent to the CNF formula (a ∨ b ∨ c) ∧ (¬a ∨ ¬b ∨ ¬c).

As to expressiveness, it is easy to see that we have
Proposition 2
The results in Table 2 hold.

### Table 1: Subsets of the \( \text{DAGPS} \) language and their corresponding polytime queries. \( \checkmark \) means "satisfies," \( \circ \) means "does not satisfy unless \( P = NP \)."

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>CD</th>
<th>FO</th>
<th>SFO</th>
<th>&amp;C</th>
<th>&amp;BC</th>
<th>vC</th>
<th>vBC</th>
<th>ME</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFF[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>renH-C[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>K/H-C[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>HORN-C[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>KROM-C[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>AFF</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>renH-C</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>K/H-C</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>HORN-C</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>KROM-C</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
</tbody>
</table>

Most of the results given in Proposition 1 are well-known or easy. We mainly provide them for the sake of completeness. As to transformations, we have obtained the following results (which, like succinctness results, are typically less easy):

Proposition 2
The results in Table 2 hold.

### Table 2: Subsets of the \( \text{DAGPS} \) language and their corresponding polytime transformations. \( \checkmark \) means "satisfies," \( \bullet \) means "does not satisfy," \( \circ \) means "does not satisfy unless \( P = NP \)."

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>CD</th>
<th>FO</th>
<th>SFO</th>
<th>&amp;C</th>
<th>&amp;BC</th>
<th>vC</th>
<th>vBC</th>
<th>ME</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFF[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>renH-C[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>K/H-C[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>HORN-C[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>KROM-C[v]</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>AFF</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>renH-C</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>K/H-C</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>HORN-C</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
<tr>
<td>KROM-C</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
<td>\checkmark</td>
</tr>
</tbody>
</table>

In the light of the results reported in Propositions 1 and 2, we can draw the following remarks:

- Focusing on the queries only, the fragments we have considered can be gathered into two classes: one contains all the incomplete fragments and the other one contains all complete fragments. Within a class, all fragments have the same tractable queries (among those considered here) except AFF in the class of incomplete fragments which satisfies also MC. The incomplete fragments satisfy more tractable queries than complete fragments (in some sense, this balances their loss w.r.t. expressiveness).

- Taking transformations into account renders the comparison more complex: except KROM-C and AFF which satisfy the same feasible and tractable transformations (but are incomparable w.r.t. expressiveness), the incomplete fragments exhibit pairwise distinct sets of feasible and tractable transformations, justifying further that each of them has its own interest.

Notice that KROM-C[v] and AFF[v] satisfy FO, just like DNNF does (recall that OBDD does not). Each of AFF[v], HORN-C[v], and KROM-C[v] satisfies \( \&BC \) while DNNF does not. renH-C[v] and K/H-C[v] loose this property; this seems to be the price to be paid for the gain in succinctness offered by these fragments, compared to HORN-C[v] and KROM-C[v], as shown in the next section.

### Succinctness

Because they are incomplete fragments, we do not put KROM-C, HORN-C, K/H-C, renH-C, or AFF into the succinctness picture. We split our results into two propositions (and two tables). In the first table, we compare the complete fragments we have considered w.r.t. spatial efficiency \( \leq_s \).

Proposition 3
The results in Table 3 hold.

### Table 3: Succinctness of target compilation languages.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
</tr>
<tr>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
</tr>
<tr>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
</tr>
<tr>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
<td>( m_a )</td>
</tr>
</tbody>
</table>

Proposition 3 shows in particular that
\[
\text{renH-C[v]} \leq_s \text{K/H-C[v]} \leq_s \text{HORN-C[v]} \leq_s \text{KROM-C[v]} \leq_s \text{AFF[v]}
\]
while HORN-C[v] and KROM-C[v] are incomparable w.r.t. \( \leq_s \), and AFF[v] is incomparable w.r.t. succinctness with any fragment among KROM-C[v], HORN-C[v], K/H-C[v], and renH-C[v].

A direct consequence of Proposition 3 is that it makes sense to consider each of HORN-C[v], KROM-C[v], AFF[v], and renH-C[v] as a target fragment for knowledge compilation; indeed, HORN-C[v], KROM-C[v], AFF[v] are pairwise incomparable from the point of view of succinctness. This tells, from the practical side, that each of them can lead to exponentially smaller compiled forms than the other ones, depending on the instance at hand. renH-C[v] is shown strictly more succinct than HORN-C[v] and KROM-C[v], but, unless \( P = NP \) it does not satisfy \( \&BC \) while HORN-C[v] and KROM-C[v] do. Finally, renH-C[v] seems to be a better choice than \( K/H-C[v] \) in the sense that it is strictly more succinct than it while it has the same set of tractable queries and transformations.

In the second table, we compare w.r.t. \( \leq_s \) the fragments KROM-C[v], HORN-C[v], K/H-C[v], renH-C[v].

\(^1\text{One can observe that this succinctness picture is similar to the expressiveness picture for the "corresponding" incomplete fragments AFF, renH-C, K/H-C, HORN-C, KROM-C.}\)
and AFF[^V] with many of the complete fragments considered in (Darwiche & Marquis 2002).

**Proposition 4** The results in Table 4 hold.

<table>
<thead>
<tr>
<th>Fragment</th>
<th>AFF[^V]</th>
<th>renH[^C][^V]</th>
<th>K/H[^C][^V]</th>
<th>HORN[^C][^V]</th>
<th>KROM[^C][^V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNF</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
</tr>
<tr>
<td>DNF</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
</tr>
<tr>
<td>1V</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
</tr>
<tr>
<td>OBD</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
<td>S, S, S</td>
</tr>
</tbody>
</table>

Table 4: Succinctness of target compilation languages. * means that the result holds unless the polynomial hierarchy collapses.

The results given in Propositions 3 and 4 are synthesized on Figure 1, which can be interpreted as the Hasse diagram collapsing. It appears that each of the five complete fragments we considered (namely AFF[^V], renH[^C][^V], K/H[^C][^V], HORN[^C][^V], KROM[^C][^V]) is strictly more succinct than DNF and incomparable w.r.t. \( \leq_s \) to OBDD (and possibly to PL – this is known for sure for AFF[^V], K/H[^C][^V], HORN[^C][^V] and KROM[^C][^V]). The main open question concerns the relationships between AFF[^V], renH[^C][^V], K/H[^C][^V], HORN[^C][^V], KROM[^C][^V] and DNF w.r.t. succinctness. We do not know whether DNF is at least as succinct as any of them. We conjecture that this is not the case.

**Discussion**

Our results show AFF[^V], renH[^C][^V], K/H[^C][^V], HORN[^C][^V], KROM[^C][^V] as interesting alternatives to many of the complete fragments considered in (Darwiche & Marquis 2002) for the knowledge compilation purpose. Indeed, in the light of the results reported in Propositions 1, 2, 3, and 4, the following conclusions can be drawn:

- AFF[^V] and KROM[^C][^V] are strictly more succinct than DNF while satisfying the same tractable queries and the same tractable transformations; similarly, renH[^C][^V], K/H[^C][^V], and HORN[^C][^V] are strictly more succinct than DNF^2 while satisfying the same tractable queries, and possibly the same tractable transformations (depending on FO). Thus, AFF[^V] and KROM[^C][^V] prove better fragments than DNF in a perspective of knowledge compilation, while HORN[^C][^V], K/H[^C][^V], and renH[^C][^V] are at least challenging alternatives to it.
- Each of AFF[^V], renH[^C][^V], K/H[^C][^V], HORN[^C][^V], KROM[^C][^V] is strictly more succinct than IP, but satisfies less tractable queries than it. But each of them challenges IP w.r.t. transformations: AFF[^V], HORN[^C][^V], and KROM[^C][^V] satisfy more tractable transformations than IP, the set of tractable transformations satisfied by renH[^C][^V] or K/H[^C][^V] and the set of tractable transformations satisfied by IP being incomparable w.r.t. \( \subseteq_s \).
- PI and any of AFF[^V], K/H[^C][^V], HORN[^C][^V] are incomparable w.r.t. succinctness. PI satisfies more tractable queries than any of them; however it satisfies less tractable transformations than AFF[^V] and KROM[^C][^V], and possibly of renH[^C][^V], K/H[^C][^V], and HORN[^C][^V] depending on FO.
- Similarly, OBDD and any of AFF[^V], renH[^C][^V], K/H[^C][^V], HORN[^C][^V], KROM[^C][^V] are incomparable w.r.t. succinctness, as well as w.r.t. their sets of tractable queries or transformations.
- AFF[^V], renH[^C][^V], K/H[^C][^V], HORN[^C][^V], KROM[^C][^V] and DNF satisfy exactly the same set of tractable queries. DNF satisfies more tractable transformations than renH[^C][^V] or K/H[^C][^V], but can be challenged by aff[^V], HORN[^C][^V] and KROM[^C][^V] when \( \land BC \) is required: the latter fragments satisfy this transformation while DNF does not.

Finally, let us remark that, when they are expressive enough and no transformation is required by the application under consideration, the incomplete fragments renH[^C] and AFF prove as good choices for the compilation purpose: they are the most expressive fragments among the five incomplete ones (KROM[^C], HORN[^C], K/H[^C], renH[^C] and AFF) and they offer many tractable queries among those considered in the KC map (all of them for AFF). When transformations are required, the performance of renH[^C] gets lower; AFF is still a valuable candidate, provided than neither \(^-C\) nor \(\lor C\) (or \(\lor BC\)) are required: all the other transformations (and all the queries) can be performed in polynomial time within this fragment. When it proves expressive enough, AFF may even challenge OBDD since it satisfies FO and \(\land C\) while OBDD does not (unless P = NP).

**Conclusion**

In this paper, we have extended the KC map introduced by Darwiche and Marquis with ten propositional fragments based on the Krom CNF one, the Horn CNF one, and the affine one. Each fragment has been evaluated w.r.t. several criteria, including the complexity of basic queries and transformations, and its spatial efficiency has also been analyzed. As discussed in the previous section, AFF[^V], renH[^C][^V], HORN[^C][^V], KROM[^C][^V] prove valuable alternatives to many of the complete fragments considered in (Darwiche & Marquis 2002). AFF[^V], in particular, satisfies both FO and \(\land BC\), while being strictly more succinct than each of the fragments in (Darwiche & Marquis 2002) offering both transformations. When expressive enough, the incomplete fragments renH[^C] and AFF are also valuable as target languages for the compilation purpose.

This paper opens a number of perspectives. One of them consists in determining whether DNF is incomparable or not with AFF[^V], renH[^C][^V], K/H[^C][^V], HORN[^C][^V], KROM[^C][^V] w.r.t. succinctness. Another perspective consists in extending further the KC map by applying the prin-
Figure 1: The succinctness picture. A solid edge from $C_1$ to $C_2$ indicates that $C_1$ is strictly more succinct than $C_2$ (edges stemming from transitivity of strict succinctness are not explicitly represented; some of the edges are conditioned on the polynomial hierarchy not collapsing). The new fragments are in shadowed boxes. Bold edges are new results. Dashed arrows indicate unknown relationships. The lack of edges, up to transitivity, between two fragments reflects the fact that they are incomparable.

Cinciple of $\lor$ closure at work here to other fragments than $\mathsf{KROM-C}$, $\mathsf{HORN-C}$, or $\mathsf{AFF}$, leading to new fragments (e.g. $\mathsf{OBDD}$ and $\mathsf{PI}$), and evaluating them. Another line for further research concerns the design of compilation algorithms targeting the $\mathsf{AFF}[\lor]$ fragment (interestingly, for the four other complete fragments such algorithms are already available (Boufkhad et al. 1997)).

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. They have been partly supported by the ANR project PHAC (ANR-05-BLAN-0384). Pierre Marquis has also been partly supported by the IUT de Lens and the Région Nord/Pas-de-Calais.

References


