Towards a Knowledge Compilation Map for Heterogeneous Representation Languages

Alexandre Niveau[1]

joint work with

Hélène Fargier[2] Pierre Marquis[3]

published in IJCAI'13

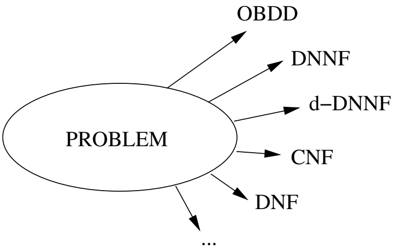
1. GREYC-CNRS, Caen, France — alexandre.niveau@unicaen.fr

2. IRIT-CNRS, Toulouse, France — fargier@irit.fr

3. CRIL-CNRS, Lens, France — marquis@cril.fr

June 4th, 2015

Choosing a Compilation Language



- What is the best language for my application?

→ use the knowledge compilation map [Dar02]

- Compares languages according to two criteria:
  1. efficiency of operations
  2. succinctness

[Dar02] Adnan Darwiche and Pierre Marquis. "A Knowledge Compilation Map". In: *Journal of Artificial Intelligence Research (JAIR)* 17 (2002), pp. 229–264

Knowledge Compilation Map: Operations

- All online manipulations boil down to elementary queries and transformations

L	**CO** (consistency)	**VA** (validity)	**CE** (clause entailmt.)	**IM** (implicant check)	**EQ** (equivalence)	**SE** (entailment)	**CT** (model count)	**ME** (model enum.)
NNF	○	○	○	○	○	○	○	○
DNNF	√	○	√	○	○	○	○	√
BDD	○	○	○	○	○	○	○	○
FBDD	√	√	√	√	?	○	√	√
OBDD	√	√	√	√	√	○	√	√
DNF	√	○	√	○	○	○	○	√
CNF	○	√	○	√	○	○	○	○

L	**CD** (conditioning)	**FO** (forgetting)	**SFO** (single forg.)	$\wedge$**C** (conjunction)	$\wedge$**BC** (bounded conj.)	$\vee$**C** (disjunction)	$\vee$**BC** (bounded disj.)	$\neg$**C** (negation)
NNF	√	○	√	√	√	√	√	√
DNNF	√	√	√	○	○	√	√	○
BDD	√	○	√	√	√	√	√	√
FBDD	√	●	○	●	○	●	○	√
OBDD	√	●	√	●	○	●	○	√
DNF	√	√	√	●	√	√	√	●
CNF	√	○	√	√	√	●	√	●

√ polynomial
○ not polynomial unless P = NP
● not polynomial

Knowledge Compilation Map: Operations

- All online manipulations boil down to elementary queries and transformations

L	CO (consistency)	VA (validity)	CE (clause entailmt.)	IM (implicant check)	EQ (equivalence)	SE (entailment)	CT (model count)	ME (model enum.)
NNF	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$
DNNF	$\surd$	$\circ$	$\surd$	$\circ$	$\circ$	$\circ$	$\circ$	$\surd$
BDD	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$
FBDD	$\surd$	$\surd$	$\surd$	$\surd$	?	$\circ$	$\surd$	$\surd$
OBDD	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\circ$	$\surd$	$\surd$
DNF	$\surd$	$\circ$	$\surd$	$\circ$	$\circ$	$\circ$	$\circ$	$\surd$
CNF	$\circ$	$\surd$	$\circ$	$\surd$	$\circ$	$\circ$	$\circ$	$\circ$

L	CD (conditioning)	FO (forgetting)	SFO (single forg.)	$\wedge$C (conjunction)	$\wedge$BC (bounded conj.)	$\vee$C (disjunction)	$\vee$BC (bounded disj.)	$\neg$C (negation)
NNF	$\surd$	$\circ$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
DNNF	$\surd$	$\surd$	$\surd$	$\circ$	$\circ$	$\surd$	$\surd$	$\circ$
BDD	$\surd$	$\circ$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
FBDD	$\surd$	$\bullet$	$\circ$	$\bullet$	$\circ$	$\bullet$	$\circ$	$\surd$
OBDD	$\surd$	$\bullet$	$\surd$	$\bullet$	$\circ$	$\bullet$	$\circ$	$\surd$
DNF	$\surd$	$\surd$	$\surd$	$\bullet$	$\surd$	$\surd$	$\surd$	$\bullet$
CNF	$\surd$	$\circ$	$\surd$	$\surd$	$\surd$	$\bullet$	$\surd$	$\bullet$

$\surd$ polynomial
$\circ$ not polynomial unless P = NP
$\bullet$ not polynomial

Knowledge Compilation Map: Operations

- All online manipulations boil down to elementary queries and transformations

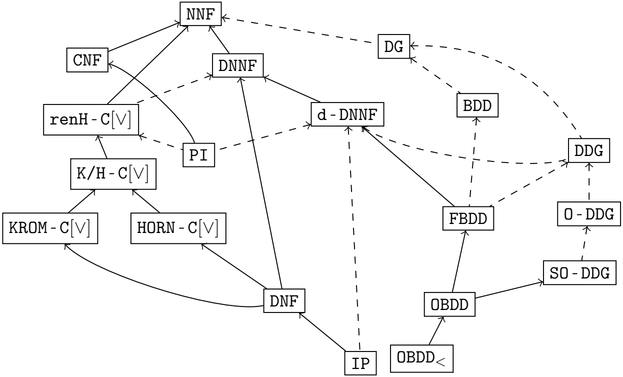
L	CO (consistency)	VA (validity)	CE (clause entailmt.)	IM (implicant check)	EQ (equivalence)	SE (entailment)	CT (model count)	ME (model enum.)
NNF	○	○	○	○	○	○	○	○
DNNF	√	○	√	○	○	○	○	√
BDD	○	○	○	○	○	○	○	○
FBDD	√	√	√	√	?	○	√	√
OBDD	√	√	√	√	√	○	√	√
DNF	√	○	√	○	○	○	○	√
CNF	○	√	○	√	○	○	○	○

L	CD (conditioning)	FO (forgetting)	SFO (single forg.)	$\wedge$C (conjunction)	$\wedge$BC (bounded conj.)	$\vee$C (disjunction)	$\vee$BC (bounded disj.)	$\neg$C (negation)
NNF	√	○	√	√	√	√	√	√
DNNF	√	√	√	○	○	√	√	○
BDD	√	○	√	√	√	√	√	√
FBDD	√	●	○	●	○	●	○	√
OBDD	√	●	√	●	○	●	○	√
DNF	√	√	√	●	√	√	√	●
CNF	√	○	√	√	√	●	√	●

√ polynomial
○ not polynomial unless P = NP
● not polynomial

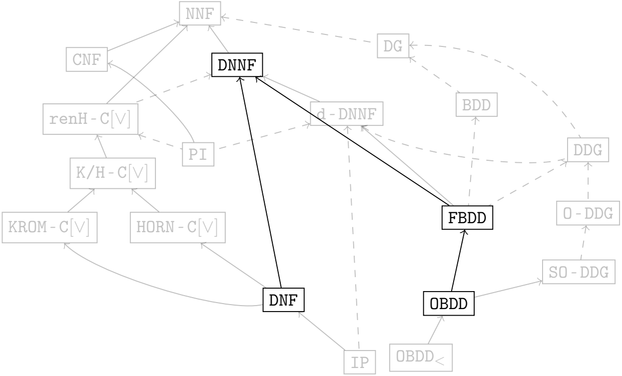
Knowledge Compilation Map: Succinctness

- Succinctness relation: orders languages w.r.t. their ability to represent knowledge compactly
- $L_1 \leq_s L_2$ means "L_1 is at least as succinct as L_2"



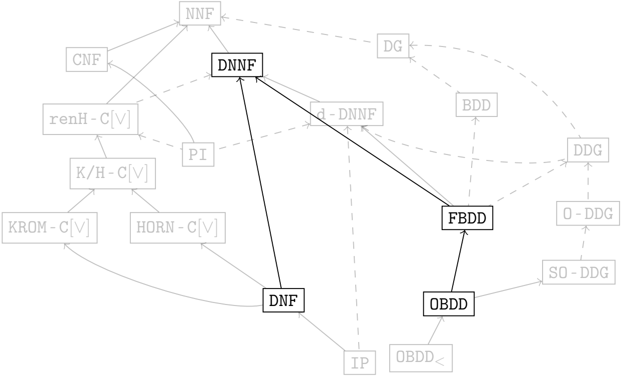
Knowledge Compilation Map: Succinctness

- Succinctness relation: orders languages w.r.t. their ability to represent knowledge compactly
- $L_1 \leq_s L_2$ means "L_1 is at least as succinct as L_2"



Knowledge Compilation Map: Succinctness

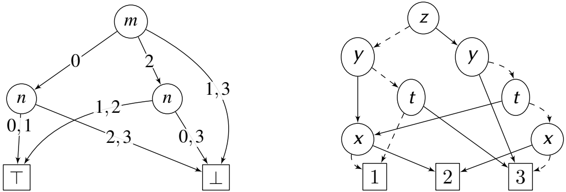
- Succinctness relation: orders languages w.r.t. their ability to represent knowledge compactly
- $L_1 \leq_s L_2$ means "L_1 is at least as succinct as L_2"



- Other relations: expressiveness ($\leq_e$),
  polynomial translatability ($\leq_p$)

Beyond Boolean Languages

- The map is drawn for lots of languages representing Boolean functions over Boolean variables
- There exists maps for languages with multivalued variables (family of MDDs) or continuous variables, and for languages representing functions with non-Boolean values (VDDs)



- Languages close in essence: generalizations of the BDD family in several directions
  - ⟶ some are "equivalent"
  - ⟶ similarities in maps

Motivation

- However, these languages are heterogeneous, i.e., they represent different kinds of objects
  - their maps are distinct
  - their "equivalence" is not formally stated within the framework
- We would like to unify the maps, in order to
  - allow the comparison of heterogeneous languages
  - factorize the common parts of the maps
  - inherit results between "close" heterogeneous languages
  - enable the diversification of the KC map setting

$\longrightarrow$ We propose a generalized framework for comparing representation languages

Plan

1 Introduction

2 Representation Languages

3 Comparing Heterogeneous Languages

4 Result Inheritance

Languages of the Classical Compilation Map

- In the classical compilation map, the notion of “language” designates a formal language:
  - A propositional formula is a word over the alphabet $P_S \cup \{\vee, \wedge, \neg, (,)\}$
  - It is in CNF if it verifies some specific properties
  - The `CNF` language is the set of all CNFs
- The notion of “language” concerns syntax only

$\longrightarrow$ the semantics is implicitly given by the interpretation function of propositional formulæ

Limitations

- This notion of language is limited:
    - implicit interpretation function
    - implicit variable domains
- Easily adaptable to other families of data structures...
- ... but implicit aspects prevent a unified presentation
- We need a more general notion

Representation Language

- Definition of a representation language, as general as possible
- Universe of discourse $\mathfrak{U}$: contains all objects that we could intend to represent (Boolean functions, real functions, etc.)
- Generic alphabet Σ: no *a priori* restriction on formulæ $\varphi \in \Sigma^*$

Definition

A representation language is a pair $\mathrm{L} = \langle \Phi_{\mathrm{L}}, \mathcal{I}_{\mathrm{L}} \rangle$, where

- Φ_{L} is the syntax of L: $\Phi_{\mathrm{L}} \subseteq \Sigma^*$;
- $\mathcal{I}_{\mathrm{L}}$ is the semantics of L: $\mathcal{I}_{\mathrm{L}} : \Sigma^* \to \mathfrak{U}$ (partial function, defined at least on all formulæ in Φ_{L}).

Examples

- Language of propositional logic: $\texttt{PROP} = \langle \Phi_{\texttt{PROP}}, \mathcal{I}_{\texttt{PROP}} \rangle$
  - $\Phi_{\texttt{PROP}}$: set of well-formed propositional formulæ
  - $\mathcal{I}_{\texttt{PROP}}$: usual interpretation function
- $\texttt{CNF} = \langle \Phi_{\texttt{CNF}}, \mathcal{I}_{\texttt{PROP}} \rangle$, with $\Phi_{\texttt{CNF}}$ the set of CNFs
- $\texttt{HORN-C} = \langle \Phi_{\texttt{HORN-C}}, \mathcal{I}_{\texttt{PROP}} \rangle$, with $\Phi_{\texttt{HORN-C}}$ the set of Horn-CNFs
- $\texttt{OMDD} = \langle \Phi_{\texttt{OMDD}}, \mathcal{I}_{\texttt{MDD}} \rangle$
  - $\Phi_{\texttt{OMDD}}$: set of ordered MDDs
  - $\mathcal{I}_{\texttt{MDD}}$: interpretation function of multivalued decision diagrams

Interpretation Space

- Semantics of L: way of interpreting some formulæ of Σ^*
- Associates with each formula $\varphi \in \Phi_L$ its interpretation $[\![\varphi]\!]_L$

Interpretation Space

- Semantics of $\mathtt{L}$: way of interpreting some formulæ of Σ^*
- Associates with each formula $\varphi \in \Phi_{\mathtt{L}}$ its interpretation $[\![\varphi]\!]_{\mathtt{L}}$
- ... but it also interprets other formulæ (semantics of $\mathtt{CNF}$: $\mathcal{I}_{\mathtt{PROP}}$, interprets also DNFs, for example)

$\longrightarrow$ interpretation space $\Omega_{\mathtt{L}}$: set of all objects represented by the semantics of $\mathtt{L}$

- Example : $\Omega_{\mathtt{PROP}} = \Omega_{\mathtt{CNF}} = \Omega_{\mathtt{HORN\text{-}C}} =$ set of Boolean functions over Boolean variables

Interpretation Space

- Semantics of $\mathtt{L}$: way of interpreting some formulæ of Σ^*
- Associates with each formula $\varphi \in \Phi_{\mathtt{L}}$ its interpretation $[\![\, \varphi \,]\!]_{\mathtt{L}}$
- ... but it also interprets other formulæ
  (semantics of $\mathtt{CNF}$: $\mathcal{I}_{\mathtt{PROP}}$, interprets also DNFs, for example)

$\longrightarrow$ interpretation space $\Omega_{\mathtt{L}}$: set of all objects represented by the semantics of $\mathtt{L}$

- Example : $\Omega_{\mathtt{PROP}} = \Omega_{\mathtt{CNF}} = \Omega_{\mathtt{HORN\text{-}C}} =$ set of Boolean functions over Boolean variables
- Completeness of $\mathtt{L}$: relative to its interpretation space
  ($\mathtt{CNF}$ is complete, $\mathtt{HORN\text{-}C}$ is incomplete)

Plan

1. Introduction
2. Representation Languages
3. Comparing Heterogeneous Languages
4. Result Inheritance

Encoding MDDs into BDDs

- In practice, MDDs are often compiled into BDDs
- Use of classical encodings (also used to go from CSP to SAT [Wal00; Pre04])
  - Direct encoding: one Boolean variable per multivalued variable and per value in the domain
  - Multivalued encoding: like the direct encoding, but no "at-most-one" constraint
  - Log encoding: Boolean variables used as bits
- Encoding an MDD into a BDD is polynomial

[Wal00; Pre04] Toby Walsh. "SAT v CSP". . In: *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP)*. 2000, pp. 441–456; Steven Prestwich. "Local Search on SAT-Encoded Colouring Problems". In: *Theory and Applications of Satisfiability Testing* (2004), pp. 26–29

Translatability of `MDD` into `BDD`

- MDDs can thus be “translated” into BDDs in polynomial time
- One would like to write $\texttt{MDD} \geq_p \texttt{BDD}$...
- But it is not the case: $\texttt{MDD} \not\geq_p \texttt{BDD}$, because they represent different kinds of functions
- The classical relation of polynomial translatability requires languages to have the same interpretation space
- We would like the compilation map to take translations into account

Translation

- We extend classical comparison relations
- Possibility of using a semantic correspondence between interpretation spaces: $\mathcal{T} \subseteq \Omega_{\mathrm{L}_1} \times \Omega_{\mathrm{L}_2}$

$\longrightarrow$ indicates objects considered as “equivalent”

- Example: given $f\colon \mathbb{N}^n \to \mathbb{B}$ and $g\colon \mathbb{B}^m \to \mathbb{B}$,

$$f\, \mathcal{T}_{dir}\, g \iff g \text{ is a direct encoding of } f$$

- Similarly for multivalued encoding $\mathcal{T}_{multi}$, log encoding $\mathcal{T}_{log}$
- $\mathcal{T}$ induces a syntactic translation between formulæ of L_1 and formulæ of L_2

Extended Polynomial Translatability

- If there exists a polynomial algorithm transforming any formula φ_1 of $\mathtt{L}_1$ into a formula φ_2 of $\mathtt{L}_2$ such that $[\![\, \varphi_1 \,]\!]_{\mathtt{L}_1} \; \mathcal{T} \; [\![\, \varphi_2 \,]\!]_{\mathtt{L}_2}$, then $\mathtt{L}_1$ is said to be polynomially translatable into $\mathtt{L}_2$ modulo $\mathcal{T}$
- We denote it as $\mathtt{L}_1 \geq_p^{\mathcal{T}} \mathtt{L}_2$

$\longrightarrow$ Generalization of the classical polynomial translatability: $\mathtt{L}_1 \geq_p \mathtt{L}_2$ corresponds to $\mathtt{L}_1 \geq_p^{\mathsf{Id}} \mathtt{L}_2$

- We also extend the succinctness and expressiveness relations to the use of a correspondence: $\mathtt{L}_1 \geq_s^{\mathcal{T}} \mathtt{L}_2$ and $\mathtt{L}_1 \geq_e^{\mathcal{T}} \mathtt{L}_2$

Examples

- Thanks to the extended relations, one can compare heterogeneous languages:
  - $\texttt{MDD} \geq_p^{\mathcal{T}_{dir}} \texttt{BDD}$ and $\texttt{MDD} \geq_p^{\mathcal{T}_{log}} \texttt{BDD}$
  - $\texttt{MDD} \not\geq_s^{\mathcal{T}_{dir}} \texttt{CNF}$
- One can also compare homogeneous languages of incomparable expressiveness (e.g., HORN-C and AFF), via a well-chosen semantic correspondence
- One can extend succinctness results from one family of languages to another via some translation:

$$\texttt{BDD} <_s \texttt{OBDD}$$
$$\Downarrow$$
$$\texttt{MDD} <_s \texttt{OMDD}$$

Plan

1 Introduction

2 Representation Languages

3 Comparing Heterogeneous Languages

4 Result Inheritance

Polynomial Translatability and Operations

- The classical polynomial translatability allows one to easily infer results about queries and transformations
  - MODS $\geq_p$ OBDD
    $\Rightarrow$ MODS satisfies all queries that OBDD satisfies
  - NNF $\sim_p$ PROP
    $\Rightarrow$ NNF and PROP satisfy the exact same set of queries and transformations
- What properties of this kind hold on languages "equivalent modulo some translation", like OBDD and OMDD?

Query Inheritance

- Classical case: if $\mathtt{L}_1 \geq_p \mathtt{L}_2$, then all queries satisfied by $\mathtt{L}_2$ are satisfied by $\mathtt{L}_1$.
- Extended case: suppose $\mathtt{L}_1 \geq_p^{\mathcal{T}} \mathtt{L}_2$.
  What can we say about queries satisfied by $\mathtt{L}_1$?

Query Inheritance

- Classical case: if $\mathtt{L}_1 \geq_p \mathtt{L}_2$, then all queries satisfied by $\mathtt{L}_2$ are satisfied by $\mathtt{L}_1$.
- Extended case: suppose $\mathtt{L}_1 \geq_p^{\mathcal{T}} \mathtt{L}_2$.
  What can we say about queries satisfied by $\mathtt{L}_1$?

$\longrightarrow$ Nothing in the general case: it depends on the $\mathcal{T}$ used

  - Let $\mathtt{L}_2$ be a language satisfying **CT**
  - $\mathcal{T}_{dir}$ maintains the number of models, so if $\mathtt{L}_1 \geq_p^{\mathcal{T}_{dir}} \mathtt{L}_2$ holds, then $\mathtt{L}_1$ also satisfies **CT**
  - $\mathcal{T}_{multi}$ does not maintain the number of models: $\mathtt{L}_1 \geq_p^{\mathcal{T}_{multi}} \mathtt{L}_2$ can hold without $\mathtt{L}_1$ satisfying **CT**

- Same problem for transformations

Inheritance Theorem

- We define (in the paper) a notion of suitability to a semantic correspondence for queries and transformations
  - **CT** is suitable to $\mathcal{T}_{dir}$, but not to $\mathcal{T}_{multi}$
  - **CO** and **CD** are suitable to both
  - **SFO** is not suitable to any of the two

Inheritance Theorem

- We define (in the paper) a notion of suitability to a semantic correspondence for queries and transformations
  - **CT** is suitable to $\mathcal{T}_{dir}$, but not to $\mathcal{T}_{multi}$
  - **CO** and **CD** are suitable to both
  - **SFO** is not suitable to any of the two

Theorem

If $\mathtt{L}_1 \geq_p^{\mathcal{T}} \mathtt{L}_2$, then all queries suitable to $\mathcal{T}$ and satisfied by $\mathtt{L}_2$ are satisfied by $\mathtt{L}_1$.
If $\mathtt{L}_1 \sim_p^{\mathcal{T}} \mathtt{L}_2$, then all transformations suitable to $\mathcal{T}$ and satisfied by $\mathtt{L}_2$ are satisfied by $\mathtt{L}_1$.

- Most queries and transformations in the map are suitable to $\mathcal{T}_{dir}$ and/or $\mathcal{T}_{multi}$

$\longrightarrow$ One can extend the results of some language over Boolean variables to some language over multivalued variables

Example of Application

- Family of "bounded MDDs"
  - k-MDD: restriction of MDD to domains of cardinality k;
  - k-FMDD: read-once fragment of k-MDD;
  - k-OMDD and k-OMDD$_<$: ordered fragments of k-MDD
- $\mathcal{T}_k$: direct encoding on domains of cardinality k
  - $\mathcal{T}_k$ is a bijection
  - all queries and transformations are suitable to $\mathcal{T}_k$

Example of Application

- Families of $\mathtt{BDD}$ and k-$\mathtt{MDD}$ are equivalent modulo $\mathcal{T}_k$ (k-$\mathtt{MDD} \sim_p^{\mathcal{T}_k} \mathtt{BDD}$, k-$\mathtt{FMDD} \sim_p^{\mathcal{T}_k} \mathtt{FBDD}$, k-$\mathtt{OMDD} \sim_p^{\mathcal{T}_k} \mathtt{OBDD}$, k-$\mathtt{OMDD}_< \sim_p^{\mathcal{T}_k} \mathtt{OBDD}_<$)
- Compilation map of $\mathtt{BDD}$:

$$\mathtt{BDD} <_s \mathtt{FBDD} <_s \mathtt{OBDD} <_s \mathtt{OBDD}_<$$

L	CO	VA	CE	IM	EQ	SE	CT	ME	CD	FO	SFO	∧C	∧BC	∨C	∨BC	¬C
BDD	○	○	○	○	○	○	○	○	√	○	√	√	√	√	√	√
FBDD	√	√	√	√	?	○	√	√	√	●	○	●	○	●	○	√
OBDD	√	√	√	√	√	√	√	√	√	●	√	●	○	●	○	√
$\mathtt{OBDD}_<$	√	√	√	√	√	○	√	√	√	●	√	●	√	●	√	√

Example of Application

- Families of BDD and k-MDD are equivalent modulo $\mathcal{T}_k$ ($k\text{-}\mathtt{MDD} \sim_p^{\mathcal{T}_k} \mathtt{BDD}$, $k\text{-}\mathtt{FMDD} \sim_p^{\mathcal{T}_k} \mathtt{FBDD}$, $k\text{-}\mathtt{OMDD} \sim_p^{\mathcal{T}_k} \mathtt{OBDD}$, $k\text{-}\mathtt{OMDD}_< \sim_p^{\mathcal{T}_k} \mathtt{OBDD}_<$)
- Compilation map of k-MDD :

$$k\text{-}\mathtt{MDD} <_s k\text{-}\mathtt{FMDD} <_s k\text{-}\mathtt{OMDD} <_s k\text{-}\mathtt{OMDD}_<$$

L	CO	VA	CE	IM	EQ	SE	CT	ME	CD	FO	SFO	$\wedge$C	$\wedge$BC	$\vee$C	$\vee$BC	$\neg$C
k-MDD	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\surd$	$\circ$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
k-FMDD	$\surd$	$\surd$	$\surd$	$\surd$	?	$\circ$	$\surd$	$\surd$	$\surd$	$\bullet$	$\circ$	$\bullet$	$\circ$	$\bullet$	$\circ$	$\surd$
k-OMDD	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\bullet$	$\surd$	$\bullet$	$\circ$	$\bullet$	$\circ$	$\surd$
k-OMDD$_<$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\circ$	$\surd$	$\surd$	$\surd$	$\bullet$	$\surd$	$\bullet$	$\surd$	$\bullet$	$\surd$	$\surd$

Conclusion

- General framework for the comparison of representation languages
- Adaptation of concepts of the knowledge compilation map

$\longrightarrow$ makes it possible to formally compare heterogeneous languages

- Mechanism to extend results from one language hierarchy to another
- First step towards a general compilation map, presenting the various hierarchies of heterogeneous languages in a unified manner (quad-trees and $R^\star$-trees, qualitative formalisms, languages representing preferences...)