Towards a Knowledge Compilation Map for Heterogeneous Representation Languages

Hélène Fargier[1] Pierre Marquis[2] Alexandre Niveau[2]

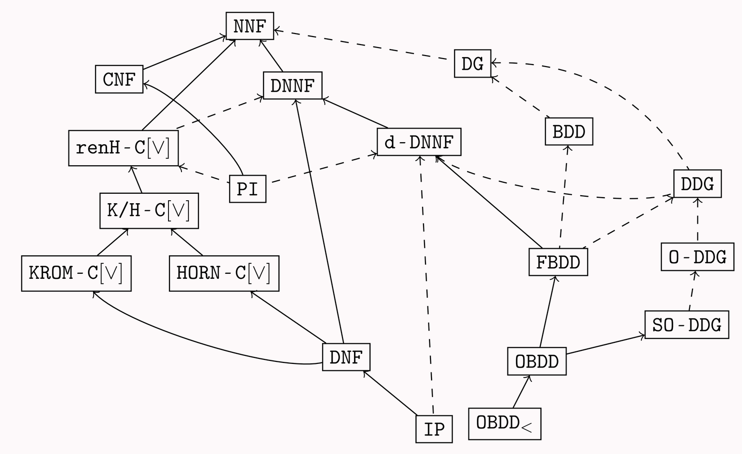
1. IRIT-CNRS, Toulouse, France — fargier@irit.fr
2. CRIL-CNRS, Lens, France — {marquis,niveau}@cril.fr

Introduction: The Knowledge Compilation Map

- Knowledge compilation consists in pre-processing information to make the online resolution of an AI problem tractable. Many target compilation languages are available; the knowledge compilation map is designed to help choosing the best language for a given application.
  - The map compares languages according to two criteria: (1) efficiency of operations, and (2) efficiency of representation.
  - (1) Online manipulations boil down to elementary queries & transformations: the map indicates whether each one is polynomial for each language.
  - (2) The 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 include expressiveness ($\leq_e$) and polynomial translatability ($\leq_p$).

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.)	∧C (conjunction)	∧BC (bounded conj.)	∨C (disjunction)	∨BC (bounded disj.)	¬C (negation)
NNF	√	○	√	√	√	√	√	√
DNNF	√	√	√	○	○	√	√	○
BDD	√	○	√	√	√	√	√	√
FBDD	√	•	○	•	○	•	○	√
OBDD	√	•	√	•	○	•	○	√
DNF	√	√	√	•	√	√	√	•
CNF	√	○	√	√	√	•	√	•



- The map is drawn for lots of languages representing Boolean functions over Boolean variables, among which the BDD family. There exists maps for languages with multivalued variables (MDD family) or continuous variables; numerous other languages exist, representing, e.g., functions with non-Boolean values (ADDs).
- **How to compare heterogeneous languages? How to unify all maps?**

Representation Language

- In the classical map, the notion of "language" designates a formal language, that is, a set of words over some alphabet. For example, the CNF language is the set of all CNFs. This notion concerns only syntax: semantics is implicitly given by the interpretation function of propositional formulæ.
- Very limited, because important features are implicit, notably the interpretation function and the domains of the variables. These implicit aspects prevent a unified presentation of heterogeneous languages.
- We define a notion of representation language as general as possible, using an universe of discourse $\mathfrak{U}$ containing all objects that we could intend to represent (Boolean functions, real functions, etc.), and a generic alphabet Σ with no restriction on formulæ $\varphi \in \Sigma^*$ (they can be any data structure).

Definition. A representation language is a pair $L = \langle \Phi_L, \mathcal{I}_L \rangle$, where $\Phi_L \subseteq \Sigma^*$ is the syntax of L, and $\mathcal{I}_L \colon \Sigma^* \to \mathfrak{U}$ is the semantics of L (partial function, defined at least on all formulæ in Φ_L).

Ex.:
- The language of propositional logic is $\text{PROP} = \langle \Phi_{\text{PROP}}, \mathcal{I}_{\text{PROP}} \rangle$, with Φ_{PROP} the set of well-formed propositional formulæ and $\mathcal{I}_{\text{PROP}}$ the classical interpretation function;
- $\text{CNF} = \langle \Phi_{\text{CNF}}, \mathcal{I}_{\text{PROP}} \rangle$, with Φ_{CNF} the set of CNFs;
- $\text{HORN-C} = \langle \Phi_{\text{HORN-C}}, \mathcal{I}_{\text{PROP}} \rangle$, with $\Phi_{\text{HORN-C}}$ the set of Horn-CNFs;
- $\text{OMDD} = \langle \Phi_{\text{OMDD}}, \mathcal{I}_{\text{MDD}} \rangle$, with Φ_{OMDD} the set of ordered MDDs and $\mathcal{I}_{\text{MDD}}$ the interpretation function of multivalued decision diagrams.

Interpretation Space

- The semantics of L is a way of interpreting some formulæ of Σ^*; it notably associates with each formula $\varphi \in \Phi_L$ in the syntax of L its interpretation $[\![\varphi]\!]_L$. But it also interprets other formulæ: for example, the semantics of CNF, $\mathcal{I}_{\text{PROP}}$, interprets any well-formed propositional formula, not only CNFs.
- Interpretation space Ω_L: set of all objects represented by the semantics of L. Ex.: $\Omega_{\text{PROP}} = \Omega_{\text{CNF}} = \Omega_{\text{HORN-C}} =$ set of Boolean functions on Boolean variables.
- The interpretation space is used to define the completeness of a language: a language is complete if its syntax covers its entire interpretation space. With this definition, CNF is complete while HORN-C is incomplete, as expected.

Sublanguages

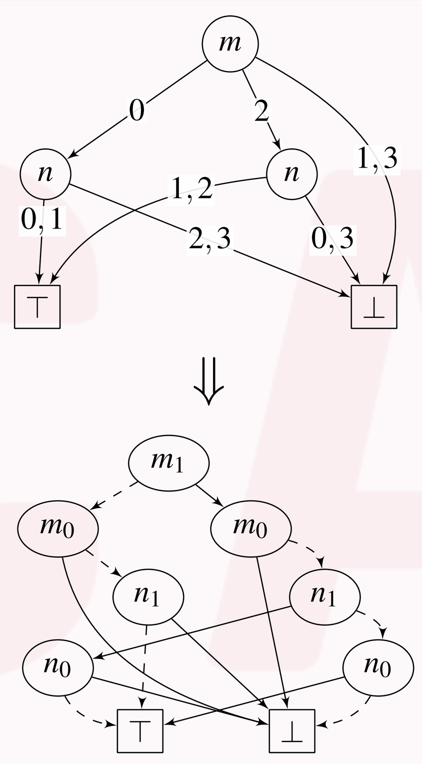
- In the classical map, inclusion constitutes a natural hierarchy over languages: for example, $\text{OBDD} \subseteq \text{FBDD} \subseteq \text{BDD}$. Now, a representation language is not simply a set of formulæ, so we need to define more specifically what a sublanguage is.

Definition. L_1 is a sublanguage of L_2 iff (i) all formulæ of L_1 are formulæ of L_2, and (ii) the semantics of L_1 is a restriction of the semantics of L_2.

- We recover the fact that HORN-C is a sublanguage of CNF, but now we can also state that OBDD is a sublanguage of OMDD.

Comparing Heterogeneous Languages

- In practice, MDDs are often compiled into BDDs (to exploit the efficiency of BDD libraries), using classical encodings (also used to go from CSP to SAT):
  - 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 a polynomial operation; however $\text{MDD} \geq_p \text{BDD}$ is not true! Indeed, 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: we extend classical comparison relations to the possibility of using a semantical correspondence between interpretation spaces, $\mathcal{T} \subseteq \Omega_{L_1} \times \Omega_{L_2}$, indicating objects considered as "equivalent".
- Each $\mathcal{T}$ induces a syntactic translation between formulæ of L_1 and formulæ of L_2. Examples: direct encoding $\mathcal{T}_{dir}$, multivalued encoding $\mathcal{T}_{multi}$, log encoding $\mathcal{T}_{log}$…

Definition. If there exists a polynomial algorithm transforming any formula φ_1 of L_1 into a formula φ_2 of L_2 such that $[\![\varphi_1]\!]_{L_1} \mathcal{T} [\![\varphi_2]\!]_{L_2}$, then L_1 is said to be polynomially translatable into L_2 modulo $\mathcal{T}$, denoted $L_1 \geq_p^{\mathcal{T}} L_2$.

- Generalization of the classical polynomial translatability: $L_1 \geq_p L_2$ corresponds to $L_1 \geq_p^{\text{Id}} L_2$. We also extend succinctness and expressiveness relations to the use of a correspondence: $L_1 \geq_s^{\mathcal{T}} L_2$ and $L_1 \geq_e^{\mathcal{T}} L_2$.
- Thanks to the extended relations, one can formally compare heterogeneous languages: for example, it holds that $\text{OMDD} \geq_p^{\mathcal{T}_{dir}} \text{OBDD}$ and $\text{MDD} \not\geq_s^{\mathcal{T}_{dir}} \text{CNF}$.
- One can also compare homogeneous languages of incomparable expressiveness (e.g., HORN-C and AFF), which is not possible in the classical framework, via a well-chosen semantical correspondence.

Result Inheritance

- The classical polynomial translatability allows one to easily infer results about queries and transformations:
  - $\text{MODS} \geq_p \text{OBDD} \Rightarrow$ MODS satisfies all queries that OBDD satisfies;
  - $\text{NNF} \sim_p \text{PROP} \Rightarrow$ 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? Suppose that $L_1 \geq_p^{\mathcal{T}} L_2$. What can we say, e.g., about queries satisfied by L_2? Are they always satisfied by L_1?
- No, in the general case: it depends on the $\mathcal{T}$ used.
  - Let L_2 be a language satisfying **CT**.
  - $\mathcal{T}_{dir}$ maintains the number of models, so if $L_1 \geq_p^{\mathcal{T}_{dir}} L_2$ holds, then L_1 also satisfies **CT**.
  - $\mathcal{T}_{multi}$ does not maintain models: $L_1 \geq_p^{\mathcal{T}_{multi}} L_2$ can hold without L_1 satisfying **CT**.
- We define a notion of suitability to a semantical correspondence for queries and transformations. For example, **CT** is suitable to $\mathcal{T}_{dir}$, but not to $\mathcal{T}_{multi}$; **CO** and **CD** are suitable to both; and **SFO** is not suitable to any of the two.

Theorem. *(1) If $L_1 \geq_p^{\mathcal{T}} L_2$, then all queries suitable to $\mathcal{T}$ and satisfied by L_2 are satisfied by L_1. (2) If $L_1 \sim_p^{\mathcal{T}} L_2$, then all transformations suitable to $\mathcal{T}$ and satisfied by L_2 are satisfied by L_1.*

- Most queries and transformations of the map are suitable to $\mathcal{T}_{dir}$ and/or $\mathcal{T}_{multi}$: one can extend the results of some language over Boolean variables to some language over multivalued variables.