Towards Practical First-Order Model Counting: Technical Appendix

Ananth K. Kidambi
Indian Institute of Technology Bombay, Mumbai, India

Guramrit Singh
Indian Institute of Technology Bombay, Mumbai, India

Paulius Dilkas
University of Toronto, Toronto, Canada
Vector Institute, Toronto, Canada

Kuldeep S. Meel
University of Toronto, Toronto, Canada

1 The Three Logics of FOMC

- Reference the table
- Make sure the rest of the paper doesn't refer to these logics (or describe examples in them)
- Move the URLs to the main text (and make sure that each algorithm name is combined with a citation at least once)
- Later (probably) move this to a different file

There are three first-order logics commonly used in FOMC: FO, C^2, and $\mathsf{UFO}^2 + \mathsf{CC}$. First, FO is the input format for ForcLift* and its extensions Crane† and Gantry. Second, C^2 is often used in the literature on FastWFOMC‡ and related methods [1, 2]. Finally, $\mathsf{UFO}^2 + \mathsf{CC}$ is the input format supported by the most recent implementation of FastWFOMC [4]. All three logics are function-free, and domains are always assumed to be finite. As usual, we presuppose the *unique name assumption*, which states that two constants are equal if and only if they are the same constant [3].

In FO, each term is assigned to a *sort*, and each predicate P/n is assigned to a sequence of n sorts. Each sort has its corresponding domain. These assignments to sorts are typically left implicit and can be reconstructed from the quantifiers. For example, $\forall x, y \in \Delta.\ P(x, y)$ implies that variables x and y have the same sort. On the other hand, $\forall x \in \Delta.\ \forall y \in \Gamma.\ P(x, y)$ implies that x and y have different sorts, and it would be improper to write, for example, $\forall x \in \Delta.\ \forall y \in \Gamma.\ P(x, y) \lor x = y$. FO is also the only logic to support constants, sentences with more than two variables, and the equality predicate. While we do not explicitly refer to sorts in subsequent sections of this paper, the many-sorted nature of FO is paramount to the algorithms presented therein.

▶ Remark. In the case of ForcLift and its extensions, support for a sentence as valid input does not imply that the algorithm can compile the sentence into a circuit or graph suitable for lifted model counting. However, it is known that ForcLift compilation is guaranteed to succeed on any FO sentence without constants and with at most two variables [5, 6].

Compared to FO, C^2 and $\mathsf{UFO}^2 + \mathsf{CC}$ lack support for constants, the equality predicate, multiple domains, and sentences with more than two variables. The advantage that C^2 brings

* https://github.com/UCLA-StarAI/Forclift
† https://doi.org/10.5281/zenodo.8004077
‡ https://github.com/jan-toth/FastWFOMC.jl

Logic	Sorts	Constants	Variables	Quantifiers	Additional atoms
FO	one or more	✓	unlimited	$\forall$, $\exists$	$x = y$
C^2	one	✗	two	$\forall$, $\exists$, $\exists^{=k}$, $\exists^{\le k}$, $\exists^{\ge k}$	—
$\mathsf{UFO}^2 + \mathsf{CC}$	one	✗	two	$\forall$	$\lvert P \rvert = m$

Table 1 A comparison of the three logics used in FOMC. The 2^{nd}–5^{th} columns refer to: the number of sorts, support for constants, the maximum number of variables, and supported quantifiers, respectively. The last column lists supported atoms in addition to those of the form $P(\mathbf{t})$ for a predicate P/n and an n-tuple of terms $\mathbf{t}$. Here: k and m are non-negative integers, with the latter depending on the domain size, P represents a predicate, and x and y are terms.

over FO is the inclusion of *counting quantifiers*. That is, alongside $\forall$ and $\exists$, C^2 supports $\exists^{=k}$, $\exists^{\le k}$, and $\exists^{\ge k}$ for any positive integer k. For example, $\exists^{=1}x.\ \phi(x)$ means that there exists *exactly one* x such that $\phi(x)$, and $\exists^{\le 2}x.\ \phi(x)$ means that there exist *at most two* such x. $\mathsf{UFO}^2 + \mathsf{CC}$, on the other hand, does not support any existential quantifiers but instead incorporates *(equality) cardinality constraints*. For example, $|P| = 3$ constrains all models to have *precisely three positive literals with the predicate* P.

1.1 Our Benchmarks in C^2 and $\mathsf{UFO}^2 + \mathsf{CC}$

- An introductory paragraph
- Figure out the right order in which the information below should be presented
- Make the text below coherent
- Refer to it in the main text (and make sure the reference sticks when I transfer this to a separate file)

Friends & Smokers in C^2 and $\mathsf{UFO}^2 + \mathsf{CC}$ is the same as in FO.

For *Bijections*, the equivalent sentence in C^2 is

$$(\forall x \in \Delta.\ \exists^{=1} y \in \Delta.\ P(x, y)) \land (\forall y \in \Delta.\ \exists^{=1} x \in \Delta.\ P(x, y)).$$

Similarly, in $\mathsf{UFO}^2 + \mathsf{CC}$ the same sentence can be written as

$$(\forall x, y \in \Delta.\ R(x) \lor \neg P(x, y)) \land (\forall x, y \in \Delta.\ S(x) \lor \neg P(y, x)) \land (|P| = |\Delta|),$$

where $w^-(R) = w^-(S) = -1$.

For *Functions*, in C^2 one would write $\forall x \in \Delta.\ \exists^{=1} y \in \Delta.\ P(x, y)$. In $\mathsf{UFO}^2 + \mathsf{CC}$, the same could be written as

$$(\forall x, y \in \Delta.\ S(x) \lor \neg P(x, y)) \land (|P| = |\Delta|), \tag{1}$$

where $w^-(S) = -1$. Although Sentence (1) has more models compared to its counterpart in C^2, the negative weight $w^-(S) = -1$ makes some of the terms in the definition of WFOMC cancel out.

References

1 Ondřej Kuželka. Weighted first-order model counting in the two-variable fragment with counting quantifiers. *J. Artif. Intell. Res.*, 70:1281–1307, 2021. doi:10.1613/JAIR.1.12320.

2 Sagar Malhotra and Luciano Serafini. Weighted model counting in FO2 with cardinality constraints and counting quantifiers: A closed form formula. In *AAAI*, pages 5817–5824. AAAI Press, 2022. doi:10.1609/AAAI.V36I5.20525.

3 Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020.

4 Jan Tóth and Ondřej Kuželka. Complexity of weighted first-order model counting in the two-variable fragment with counting quantifiers: A bound to beat. In *KR*, 2024. `doi:10.24963/KR.2024/64`.

5 Guy Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *NIPS*, pages 1386–1394, 2011. URL: `https://proceedings.neurips.cc/paper/2011/hash/846c260d715e5b854ffad5f70a516c88-Abstract.html`.

6 Guy Van den Broeck, Wannes Meert, and Adnan Darwiche. Skolemization for weighted first-order model counting. In *KR*. AAAI Press, 2014. URL: `http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/8012`.