

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

FOMC commonly utilises three types of first-order logic: FO, C^2 , and $UFO^2 + CC$. Table 1 summarises the key differences among them. FO is the input format for FORCLIFT and its extensions CRANE and GANTRY. C^2 is often used in the literature on FASTWFOMC and related methods [1, 2]. (Note that no algorithm accepts C^2 as input.) Finally, $UFO^2 + CC$ 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 has a designated *sort*, and each predicate P/n corresponds to a sequence of n sorts. Each sort has its corresponding domain. These assignments to sorts are typically left implicit and follow from the quantifiers, e.g., $\forall x, y \in \Delta$. $P(x, y)$ implies that the 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, $\forall x \in \Delta$. $\forall y \in \Gamma$. $P(x, y) \vee x = y$. 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 the 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, FORCLIFT compilation always succeeds on any FO sentence without constants and with at most two variables [5, 6].

Compared to FO, C^2 and $UFO^2 + CC$ lack support for constants, the equality predicate, multiple domains, and sentences with more than two variables. The advantage that C^2 brings over FO is the inclusion of *counting quantifiers*. That is, alongside \forall and \exists , C^2 supports $\exists^{=k}$, $\exists^{\leq k}$, and $\exists^{\geq k}$ for any positive integer k . For example, $\exists^{=1}x$. $\phi(x)$ means that there exists *exactly one* x such that $\phi(x)$, and $\exists^{\leq 2}x$. $\phi(x)$ means that there exist *at most two* such x . $UFO^2 + CC$, 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* .

Our Benchmarks in C^2 and $UFO^2 + CC$

For completeness and reproducibility, let us translate the benchmark sentences from FO to C^2 and $UFO^2 + CC$. Since *Friends & Smokers* is a relatively simple sentence, it remains the

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

■ **Table 1** A comparison of the three logics used in FOMC. The 2nd–5th 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, where m depends on the domain size, P is a predicate, and x and y are terms.

45 same in C^2 and $UFO^2 + CC$. For *Functions*, in C^2 , one would write

46
$$\forall x \in \Delta. \exists^{=1} y \in \Delta. P(x, y).$$

47 In $UFO^2 + CC$, the equivalent formulation is

48
$$(\forall x, y \in \Delta. S(x) \vee \neg P(x, y)) \wedge (|P| = |\Delta|), \quad (1)$$

49 where $w^-(S) = -1$. Although sentence (1) has more models than its counterpart in C^2 , the
50 negative weight $w^-(S) = -1$ causes some of the terms in the definition of WFOMC to cancel
51 out. The translation of *Bijections* is similar to that of *Functions*. In C^2 , one could write

52
$$(\forall x \in \Delta. \exists^{=1} y \in \Delta. P(x, y)) \wedge (\forall y \in \Delta. \exists^{=1} x \in \Delta. P(x, y)).$$

53 Similarly, in $UFO^2 + CC$, the equivalent formulation is

54
$$(\forall x, y \in \Delta. R(x) \vee \neg P(x, y)) \wedge (\forall x, y \in \Delta. S(x) \vee \neg P(y, x)) \wedge (|P| = |\Delta|),$$

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

56 — References —

- 57 **1** Ondřej Kuželka. Weighted first-order model counting in the two-variable fragment with
58 counting quantifiers. *J. Artif. Intell. Res.*, 70:1281–1307, 2021. doi:10.1613/JAIR.1.12320.
- 59 **2** Sagar Malhotra and Luciano Serafini. Weighted model counting in FO2 with cardinality
60 constraints and counting quantifiers: A closed form formula. In *AAAI*, pages 5817–5824. AAAI
61 Press, 2022. doi:10.1609/AAAI.V36I5.20525.
- 62 **3** Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*.
63 Pearson, 2020.
- 64 **4** Jan Tóth and Ondřej Kuželka. Complexity of weighted first-order model counting in the
65 two-variable fragment with counting quantifiers: A bound to beat. In *KR*, 2024. doi:
66 10.24963/KR.2024/64.
- 67 **5** Guy Van den Broeck. On the completeness of first-order knowledge compilation for lifted
68 probabilistic inference. In *NIPS*, pages 1386–1394, 2011. URL: [https://proceedings.neurips.
69 cc/paper/2011/hash/846c260d715e5b854ffad5f70a516c88-Abstract.html](https://proceedings.neurips.cc/paper/2011/hash/846c260d715e5b854ffad5f70a516c88-Abstract.html).
- 70 **6** Guy Van den Broeck, Wannes Meert, and Adnan Darwiche. Skolemization for weighted
71 first-order model counting. In *KR*. AAAI Press, 2014. URL: [http://www.aaai.org/ocs/
72 index.php/KR/KR14/paper/view/8012](http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/8012).