Automatic Conjecturing of P-Recursions Using Lifted Inference

Jáchym Barvínek[1] Timothy van Bremen[2] Yuyi Wang[3] Filip Železný[1] Ondřej Kuželka[1]
[1]Czech Technical Univeristy in Prague, Czechia, [2]KU Leuven, Belgium, [3]ETH Zurich, Switzerland

Czech Technical University in Prague

Abstract

Recent progress in lifted inference algorithms has made it possible to solve many non-trivial counting tasks from enumerative combinatorics in an automated fashion, by casting them as first-order model counting problems. Algorithms for this problem typically output a single number, which is the number of models of the first-order logic sentence in question on a given domain. However, in the combinatorics setting, we are more interested in obtaining a mathematical formula that holds for any given structure size. In this paper, we show that one can use lifted inference algorithms to conjecture linear recurrences with polynomial coefficients, one such class of formulas of interest.

First-order Logic and Model Counting

We deal with the function-free, finite domain fragment of first-order logic. Here we restrict ourselves to the two-variable fragment of first-order logic with counting quantifiers, which is usually referred to as the $\mathbf{C}^2$ fragment.[1] This fragment is obtained by restricting the allowed sentences to contain only two variables (here x and y) and allowing quantifiers $\exists^{=k}$, $\exists^{\leq k}$, $\exists^{\geq k}$ together with the standard $\forall$ and $\exists$ quantifiers.

First-order model count

The first-order model count (FOMC) of a sentence ϕ over a domain of size n is defined as:

$$\mathrm{FOMC}(\phi, n) = |\mathrm{models}_n(\phi)|$$

where $\mathrm{models}_n(\phi)$ denotes the set of all models of ϕ over any domain $\Delta, |\Delta| = n$. We call the sequence of numbers $a_n = \mathrm{FOMC}(\phi, n)$ the FOMC sequence of ϕ.

Example

A function $f : \Delta \to \Delta$ is called an involution if $f(f(x)) = x$ for all $x \in \Delta$. If we want to encode involutions in $\mathbf{C}^2$, we use the sentence:

$$\Psi = (\forall x \exists^{=1} y : f(x, y)) \wedge (\forall x \forall y : f(x, y) \Rightarrow f(y, x)).$$

Here, the first conjunct uses the counting quantifier $\exists^{=1}$ to force f to be a function (i.e. to have exactly one value y for every value of x) and the second conjunct forces it to be involutive. We can ask: What is the FOMC of this sentence over $\Delta = \{1, 2\}$? To answer this question, we can enumerate the models of Ψ which in this case are $\{f(1, 1), f(2, 2)\}$ and $\{f(1, 2), f(2, 1)\}$, so the answer is that FOMC is 2 in this case. Analogously, it's possible to calculate the number of involutions $\mathrm{FOMC}(\Psi, n)$ for $n = 0, 1, \ldots$ to be:

$$1, 1, 2, 4, 10, 26, 76, 232, 764, 2620, \ldots$$

In general, since we know the models of Ψ correspond to involutions, it's also known that the answer for domains of $|\Delta| = n$ is given by the recurrence

$$a_n - a_{n-1} - (n-1)a_{n-2} = 0, \quad a_0 = a_1 = 1$$

even without enumerating all models explicitly.

In our work, we show how such recurrence relations can be conjectured automatically in polynomial time given only the logical description ϕ.

Domain liftability

Lifted inference [2, 3, 4, 5] studies ways to compute the FOMC much faster than by direct enumeration of models. Here we rely on the result asserting the tractability of computing the FOMC for sentences from the $\mathbf{C}^2$ fragment of first-order logic, which builds on previous works of [4, 6, 7].

P-Recursive Sequences

P-recursive sequences are integer sequences which can be encoded by a well-behaved recurrence relation. Formally, a sequence a_n is called P-recursive if there exist polynomials $p_0, \ldots, p_k$ such that for each $n > k$ it holds:

$$\sum_{i=0}^{k} p_i(n) a_{n-i} = 0. \tag{1}$$

The maximum degree of the polynomials p_i is denoted by d. P-recursive description of a sequence has the advantages of allowing the fast calculation of its terms, easily outperforming the existing lifted inference algorithms when it exists. Here, we propose a way to conjecture such recurrence relations for a given property or rule out it's existence for reasonable values of k and d.

Conjecturing P-recursiveness from a C^2 description

Suppose we have the first l terms of a sequence $(a_0, \ldots, a_{l-1})$, and we are trying to conjecture a P-recursive relation for given values of the metaparameters k, d. From Equation (1), we can directly obtain a system of $l - k$ linear equations where the unknowns are the coefficients of p_i. This way, the problem is reduced to finding the kernel of a certain matrix.

The first l terms are calculated using the lifted inference methods mentioned above. Using a grid-search type of algorithm, we can search the space of the metaparemeters k, d to find suitable values if they exist or exit with failure if they don't.

If a P-recurrence exists within the searched range of k, d it will be identified by our method. The required size l is proportional to $k \cdot d$ and the redundant sequence terms are used to corroborate the conjecture. The method doesn't produce formal proof that the found P-recurrence is correct, but it may disprove it's existence for small k, d. For practical problems, the values of k, d do tend to be small, however.

Results

We demonstrated the potential of the approach by showing that we can automatically rediscover non-trivial recurrence relations from the literature, as well as conjecture new ones. The rediscovered recurrences include examples such as the involutions above. A newly discovered recurrence is that the relation:

$$a_n - (n-1)(a_{n-1} + (n-2)(2a_{n-2} + 3a_{n-3} + 6(n-3)a_{n-4})) = 0, \quad a_0 = 1, a_1 = a_2 = 0, a_3 = 6$$

, describes the number of graphs on n vertices which are 2-regular and 3-colored at the same time.

TL;DR

Problem: Find an easier way to calculate numbers of object satisfying some property expressible in the $\mathbf{C}^2$ subset of first-order logic. Our approach as a pipeline:

1. Input the logical description of a collection of objects.
2. Calculate some initial numbers of objects (sequence terms) using lifted inference algorithms.
3. Construct a well behaved recurrence relation fitting the sequence terms by reducing the problem to solving a system of linear equations or conjecture it's non-existence. If a reasonable one exists, it's guaranteed to be found.
4. If not all the sequence terms were needed to identify the relation, use the extra ones to validate the conjecture.
5. The conjecture ought to be proved using other means in order for it to be able to be used to reliably predict subsequent higher terms.
6. (Optional) Use existing algorithms to solve the recurrence to obtain a closed-form expression using special functions if it exists.

Bibliography

[1] Grädel, E., Kolaitis, P. G., and Vardi, M. Y. (1997) On the decision problem for two-variable first-order logic. *Bull. Symb. Log.*, **3**(1), 53–69.

[2] Poole, D. (2003) First-order probabilistic inference. In *IJCAI* Morgan Kaufmann pp. 985–991.

[3] de Salvo Braz, R., Amir, E., and Roth, D. (2005) Lifted First-Order Probabilistic Inference. In *IJCAI* Professional Book Center pp. 1319–1325.

[4] Van den Broeck, G. (2011) On the Completeness of First-Order Knowledge Compilation for Lifted Probabilistic Inference. In *NIPS* pp. 1386–1394.

[5] Gogate, V. and Domingos, P. M. (2011) Probabilistic Theorem Proving. In *UAI* AUAI Press pp. 256–265.

[6] Van den Broeck, G., Meert, W., and Darwiche, A. (2014) Skolemization for Weighted First-Order Model Counting. In *KR* AAAI Press.

[7] Kuusisto, A. and Lutz, C. (2018) Weighted model counting beyond two-variable logic. In *LICS* ACM pp. 619–628.