Generating maximal solutions given by closure operations

Arnaud Mary[1] and Yann Strozecki[2]

[1] Université Lyon 1, CNRS, INRIA - ERABLE
[2] Université de Versailles Saint-Quentin-en-Yvelines - DAVID Laboratories

In enumeration complexity we are interested in listing a set of elements, which can be of exponential cardinality in the size of the input. The *delay* of an enumeration algorithm is the time between the production of two consecutive solutions. Thus a good enumeration algorithm must have a bounded delay so that giving more time to the algorithm ensures to get more solutions.

Many enumeration problems can be seen as saturation problems, that is we are given a set of solutions $\mathcal{S}$ and a polynomial time function f which from a k-uple of solutions produces a solution. The problem is to generate the closure of $\mathcal{S}$ by f, denoted by $f(\mathcal{S})$. The natural algorithm is just to apply the function until we find no new solutions, therefore its delay is bounded by a polynomial in the input and the number of solutions already generated. We call such an algorithm an *incremental polytime* algorithm. There are many examples of such algorithms, for instance to generate the circuit of a matroid [6] or the minimal transversals of particular hypergraphs [5].

It turns out that when the elements of $\mathcal{S}$ are subsets of a ground set E and when the function f acts as a set operation (union, symmetric difference, majority ...), there are better algorithms to generate $f(\mathcal{S})$. The method is to solve the extension problem in polytime, that is given two subsets of E, A and B, does there exists $A \subseteq C \subseteq E \setminus B$ such that $C \in \mathcal{S}$? One can then use a backtrack search which keeps adding elements to partial solution in every possible way allowed by the extension problem. The delay of the algorithm is then polynomial in the size of the input only. In a recent article [8], we use Post's lattice [3] to list every possible set of functions f (called clone) and to prove that the associated extension problem is polynomial. We also provide better enumeration algorithm than the plain backtrack search when possible.

The natural generalization would be to consider more general functions f. For instance, we could consider functions acting coefficient-wise on a domain with more than two elements. We already provide few results in our article, and researchers from the universal algebra community proved that the extension problem is polytime for field, ring, group and some semi-group operations [9, 4]. To go beyond what is known requires an extensive knowledge of algebra and is limited since we have proved that for some very simple function f, the extension problem is NP-complete or even PSPACE-complete.

We develop here an alternative generalization. We only study closure by set operations but we add additional operators acting on the set of solutions to capture more natural enumeration problems. The problems we get may not

have incremental polytime algorithm anymore, but most of them can be solved in polynomial delay using other methods than the backtrack search.

1 Heredity operator

Let $\mathcal{S} \subseteq \mathcal{P}([n])$, we define the closure by heredity $\uparrow \mathcal{S} = \{A \mid A \supseteq B, B \in \mathcal{S}\}$, a property enjoyed by all systems of dependent sets. Given a set of solutions we now ask to generate the closure by a set of functions $\mathcal{F}$ and the heredity operator $\uparrow$. A different problem would be to enumerate $\uparrow \mathcal{F}(\mathcal{S})$. The Post's lattice becomes simpler when we add the heredity operator to the clones, since we can easily simulate a disjunction using it. If there is a way to generate a unique minimal element, for instance when there is a conjunction in the closure functions, then the solutions are elements containing the minimal element. Moreover, class containing negation are trivial and constants do not matter.

In addition to the trivial problems we have just described, there are only two cases; the first corresponds to $\mathcal{F} = \emptyset$, that is computing the set of supersets of elements in $\mathcal{S}$. It can be seen as the sets of satisfying assignments of a monotone DNF formula, where each literal represents an element of $\mathcal{S}$. This problem can be solved in polynomial delay but the polynomial depends both on the size of the subsets in $\mathcal{S}$ and on the cardinal of $\mathcal{S}$. Dropping the dependency in the cardinal is an intriguing open question.

The second case are the clones whose closure can be characterized by projections over all k-uples of the elements of $\mathcal{S}$, for instance by the Baker-Pixley theorem. It is then enough to compute the closure of the projections by the elements of the clone and by $\uparrow$ and combine them as in [8].

2 Minimal and maximal solutions

We define the set of minimal solutions by $\mathsf{Min}(\mathcal{F}(\mathcal{S})) = \{A \in \mathcal{F}(\mathcal{S}) \mid B \in \mathcal{F}(\mathcal{S}) \text{ and } B \subseteq A \Rightarrow A = B\}$. $\mathsf{Max}(\mathcal{F}(\mathcal{S}))$ is defined symmetrically. When enumeration is used to find some optimal solutions, it is useful to generate only maximal or minimal solution when they contain the optimal.

First we can remark that the reductions between saturation problems parametrized by clones established in [8] still work when one considers only the minimal elements (up to exchanging minimal and maximal elements). Therefore one has to deal with the simplified Post's lattice, that is 13 clones and two infinite families. For some clones, only one maximal or minimal element is generated by saturation which makes the problem trivial. This is the case for all clones which contain either the conjunction or the disjunction, which already settles the case of 6 clones.

Let f be the sum modulo two, then $f(\mathcal{S})$ is the vector space generated by $\mathcal{S}$. If we chose $\mathcal{S}$ to be a basis of the dependent sets of a binary matroid, then $\mathsf{Min}(\mathcal{S})$ represent the set of circuits of this matroid, a problem which can be solved with an incremental polytime algorithm, but for which no polynomial

delay algorithm is known. In particular the extension problem is NP-hard [2] therefore the backtrack search cannot be used.

A general method that works for many clones, is a variation of the method described in [7]. It was originally designed to enumerate maximal subsets of an independent set system (a set system closed under inclusion). While for most clones, $\mathcal{F}(\mathcal{S})$ does not form an independent system, the method can be used anyway.

The main idea is to enumerate all maximal (resp. minimal) restrictions of $\mathcal{F}(\mathcal{S})$ on the $i+1$ first elements of the ground set from all maximal (resp. minimal) restriction of $\mathcal{F}(\mathcal{S})$ on the i first elements. Assume for instance that we want to enumerate all minimal elements of $f(\mathcal{S})$ where f is the *maj* operator. For a given $i \leq |E|$, let us denote by $f(\mathcal{S})_i$ the set of restrictions on the i first elements of subsets of $f(\mathcal{S})$. Let T be a subset that belongs to $\mathsf{Min}(f(\mathcal{S})_i)$ Then one can produce from T subsets of $\mathsf{Min}(f(\mathcal{S})_{i+1})$ in the following way.

1. Either T itself belongs to $f(\mathcal{S})_{i+1}$ and then it belongs to $\mathsf{Min}(f(\mathcal{S})_{i+1})$
2. Or $T \cup \{i+1\}$ belongs to $\mathsf{Min}(f(\mathcal{S})_{i+1})$ and there exists a unique $T \subset T' \subseteq \{1, ..., i\}$ such that T' belongs to $\mathsf{Min}(f(\mathcal{S})_{i+1})$. Moreover such T' can be found in polynomial time.

The key point is that any subset of $\mathsf{Min}(f(\mathcal{S})_{i+1})$ can be "produced" in this way from a subset of $\mathsf{Min}(f(\mathcal{S})_i)$. This define a directed acyclic graph where vertices correspond to all restrictions of solutions and where there is an arc between $T_1 \in \mathsf{Min}(f(\mathcal{S})_i)$ and $T_2 \in \mathsf{Min}(f(\mathcal{S})_{i+1})$ if T_2 can be produced from T_1. The only source of this graph is $\emptyset$ which is the only element of $\mathsf{Min}(f(\mathcal{S})_0)$ and the sinks correspond to $\mathsf{Min}(f(\mathcal{S}))$. The enumeration algorithm consists in a simple DFS of this graph starting from the only source. This leads to a polynomial delay algorithm since the depth of the DAG is linear. To avoid an exponential storage, one can use a classical reverse search method (cf. [1])

References

1. David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65:21–46, 1993.
2. Elwyn R Berlekamp, Robert J McEliece, and Henk CA Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
3. Elmar Böhler, Steffen Reith, Henning Schnoor, and Heribert Vollmer. Bases for boolean co-clones. *Information Processing Letters*, 96(2):59–66, 2005.
4. Andrei Bulatov, Peter Mayr, and Markus Steindl. The subpower membership problem for semigroups. *arXiv preprint arXiv:1603.09333*, 2016.
5. Leonid Khachiyan, Endre Boros, Khaled Elbassioni, and Vladimir Gurvich. On the dualization of hypergraphs with bounded edge-intersections and other related classes of hypergraphs. *Theoretical Computer Science*, 382(2):139 – 150, 2007.
6. Leonid Khachiyan, Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. On the complexity of some enumeration problems for matroids. *SIAM Journal on Discrete Mathematics*, 19(4):966–984, 2005.

7. Eugene L. Lawler, Jan Karel Lenstra, and A. H. G. Rinnooy Kan. Generating all maximal independent sets: Np-hardness and polynomial-time algorithms. *SIAM J. Comput.*, 9(3):558–565, 1980.
8. Arnaud Mary and Yann Strozecki. Efficient enumeration of solutions produced by closure operations. In *33rd Symposium on Theoretical Aspects of Computer Science*, 2016.
9. Peter Mayr. The subpower membership problem for mal'cev algebras. *International Journal of Algebra and Computation*, 22(07):1250075, 2012.