A Reverse Search Method for the Enumeration of Unordered Forests using DAG Compression

Florian Ingels, Romain Azaïs

A reverse search method for the enumeration of unordered forests using DAG compression

Florian Ingels
florian.ingels@inria.fr

Romain Azaïs
romain.azais@inria.fr

Inria team MOSAIC, Lyon, France

Enumeration of trees & reverse search Although Cayley was the first to propose a formula for counting unordered trees in the mid-19th century[3, I.5.2], their exhaustive enumeration has only been tackled in the 2000s [7], using the so-called reverse search technique [1]. This method is particularly relevant for the enumeration of structures for which a notion of substructure can be defined. Indeed, it provides the space of the structures with an enumeration tree, where the children of a particular object accept it as a substructure. It is then sufficient to explore the tree from its root and recursively explore the branches.

Enumeration of forests So far the literature has only studied the enumeration of trees as single objects; we propose to solve a more general problem, the enumeration of sets of trees. We call *forest* any set of trees such that no tree in the forest is a subtree of another; and we are therefore interested in the exhaustive enumeration of forests. As trees can be considered as forests with one element, this problem generalizes the enumeration of trees, and raise it to a higher combinatorial dimension. Indeed, forests belong to the powerset of trees, exponentially bigger in nature than the set of trees. Another challenge arises from how to deal with the unordered nature of the considered objects.

Dealing with unorderedness We aim to develop an exhaustive but not redundant enumeration, i.e. we aim to avoid enumerating two isomorphic versions of the same unordered object. One way to address this problem is to find a systematic way of ordering each object, giving it a so-called canonical form. Only such canonical forms can then be enumerated, ensuring that each object is listed exactly once. This approach is notably used with unordered trees in [7]. However, when considering a forest of unordered trees, while it is always possible to order the vertices of the trees using this method, there is no total order on the set of the trees and therefore the elements of the forest cannot be ordered as such.

To circumvent this issue, we employed a method called Directed Acyclic Graph (DAG) reduction [4]. Given a forest F, taking advantage of the redundancies in the structures of the trees of F, we create a new graph, denoted $\mathfrak{R}(F)$, that is a lossless compression of F. An example is depicted in Figure 1. Since the whole forest is reduced to a unique graph without any loss of information, any ordering on the vertices of $\mathfrak{R}(F)$ induces an order on both vertices and trees composing F.

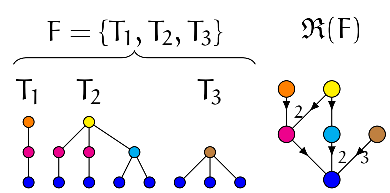

Figure 1: A forest F (left) and its DAG reduction (right).

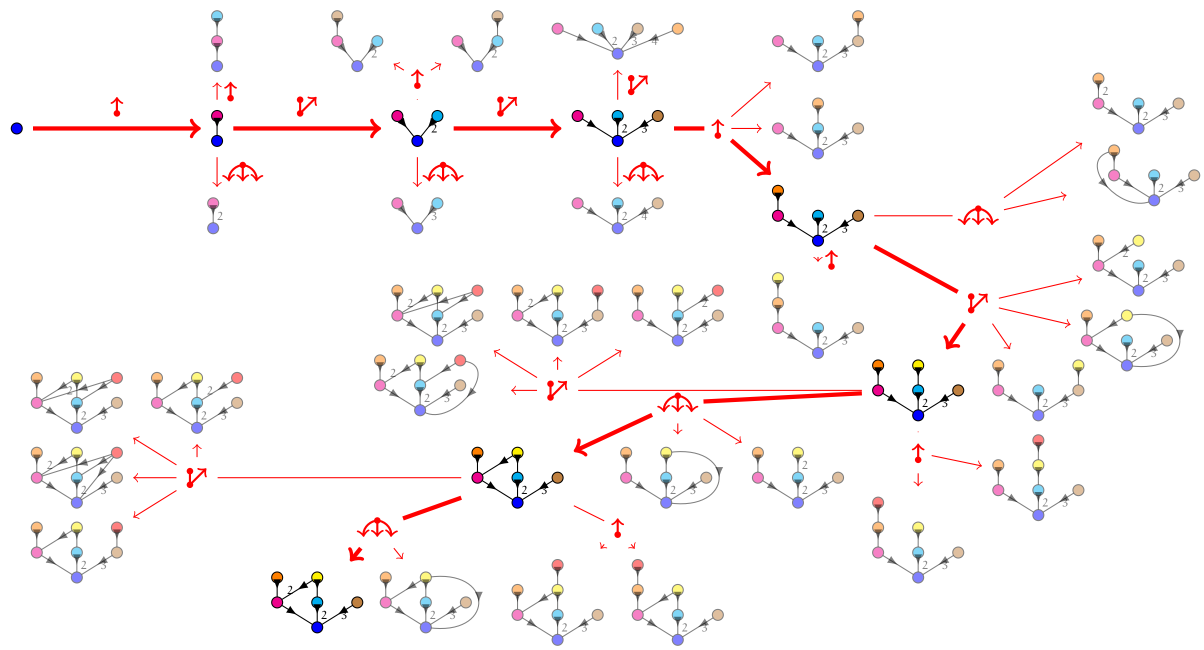

Figure 2: The path (in bold) in the FDAG enumeration tree that leads to the construction of the FDAG presented in Figure 1. The expansions rules are denoted with symbols (↥) ,(⩗) and (⩙) .

A DAG generally admits multiple ordering on its vertices; we proved that, by imposing carefully chosen topological constraints, it is possible to define a vertex ordering that is unique if and only if the DAG compresses a forest. We call such a graph FDAG and the induced order canonical. Therefore, enumerating forests is equivalent to enumerate canonical FDAGs.

Enumeration of FDAGs To address this problem, we resort to the reverse search technique. This method relies on finding *expansion rules*, that allow to "expand" an object so that its expanded versions contain it as a substructure, and also preserve some desired properties. Since FDAG operations are very indirectly related to tree operations, we had to develop completely different rules compared to the ones presented in [7] for trees. We designed three expansions rules, for which we proved that they preserve the canonical ordering and suffice to define an enumeration tree on the set of FDAGs – part of which is shown in Figure 2.

Properties of the enumeration We proved that any FDAG admits, via all three rules of expansion, a number of children in the enumeration tree *linear* in its number of vertices. This property ensures that the growth of the tree is rather controlled and therefore tractable to be explored by incremental steps. In addition, denoting E_k the set of FDAGs accessible in exactly k steps from the root, we have, as $k \to \infty$,

$$\#E_k = k! \left(\frac{12}{\pi^2}\right)^k \left(\beta + O\left(\frac{1}{k}\right)\right)$$

This formula is actually found in the literature [6, 2] for counting *row-Fisburn matrices*, for which we have proved that there exist a one-to-one correspondance between them and FDAGs.

Final note An extended version of this work, with demonstrations and details of the algorithms, can be found in [5].

References

[1] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1-3):21–46, 1996.

[2] Kathrin Bringmann, Yingkun Li, and Robert C Rhoades. Asymptotics for the number of row-fishburn matrices. *European Journal of Combinatorics*, 41:183–196, 2014.

[3] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. cambridge University press, 2009.

[4] Christophe Godin and Pascal Ferraro. Quantifying the degree of self-nestedness of trees: application to the structural analysis of plants. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(4):688–703, 2010.

[5] Florian Ingels and Romain Azaïs. Enumeration of unordered forests. *arXiv preprint arXiv:2003.08144*, 2020.

[6] Vít Jelínek. Counting general and self-dual interval orders. *Journal of Combinatorial Theory, Series A*, 119(3):599–614, 2012.

[7] Shin-ichi Nakano and Takeaki Uno. Efficient generation of rooted trees. *National Institute for Informatics (Japan), Tech. Rep. NII-2003-005E*, 8, 2003.