Séminaire Lotharingien de Combinatoire **82B** (2019)
Article #103, 8 pp.

Proceedings of the 31st Conference on Formal Power Series and Algebraic Combinatorics (Ljubljana)

FindStat – a database and search engine for combinatorial statistics and maps

Martin Rubey*[1] and Christian Stump†[2]

[1]*Fakultät für Mathematik und Geoinformation, TU Wien, Austria*
[2]*Fakultät für Mathematik, Ruhr-Universität Bochum, Germany*

Abstract. FindStat is a collaborative online database of combinatorial statistics on combinatorial collections and of maps between such collections. A search engine allows users to identify statistics and maps given a few known values, using for example the web interface. As of April 2019, the database contains 1387 statistics and 146 maps on 18 collections by 70 contributors.

1 Introduction

The *On-Line Encyclopedia of Integer Sequences* OEIS [4] is an incredibly valuable tool in mathematical research. As a 'fingerprint database for theorems' it allows to identify interconnections and relations between theorems throughout mathematics [1].

In this spirit, FindStat is a collaborative online database of combinatorial statistics on combinatorial collections and of combinatorial maps between such collections. It is mainly used via its web interface which may be found at

www.findstat.org .

A *(combinatorial) collection*[1] is, for the purpose of this project and only loosely speaking, a collection $\mathcal{C} = \bigcup_n \mathcal{C}_n$ of finite "combinatorial" sets $\mathcal{C}_n$. Examples are

- the set of permutations of $\{1, \ldots, n\}$,
- the set of integer partitions of n,
- the set of binary words with n letters, and
- the set of simple unlabelled graphs on n vertices.

A *(combinatorial) map* is a map $\phi : \mathcal{C} \to \mathcal{C}'$ between two such collections. An example is

*Supported by the Austrian Science Fund (FWF): P 29275.
†Supported by DFG grants STU 563/2 "Coxeter-Catalan combinatorics" and STU 563/4-1 "Noncrossing phenomena in Algebra and Geometry".
[1]www.findstat.org/CollectionsDatabase

- the map sending a permutation of $\{1,\dots,n\}$ to the integer partition of n given by its *cycle type*[2].

Finally, a *(combinatorial) statistic* is a map $st : \mathcal{C} \to \mathbb{Z}$. Examples are

- the *number of descents* of a permutation[3],
- the *number of standard Young tableaux* of an integer partition[4],
- the *number of different cyclic shifts* of a binary word[5],
- the *maximal cardinality of a clique* in a simple unlabelled graph[6].

The fundamental idea of `FindStat` is to

provide a tool for a computer-aided study of statistics on combinatorial collections by taking into account combinatorial maps.

The question that started the project was: *how to search the* `OEIS` *for the number of descents of a permutation?* A first attempt at an answer might be to

- search for the generating function $\sum_{\pi \in \mathcal{S}_n} x^{\mathrm{des}(\pi)}$.

This works and reveals the *Eulerian numbers* `www.oeis.org/A008292`. But going from a statistic to its generating function means to "ignore" a lot of combinatorial information. A second attempt could be to

- list all permutations in some order and then search for the resulting sequence.

Maybe the most obvious order is lexicographic. Although this fails to reveal any information for the example at hand, the design of the `OEIS` would in principle be capable of handling such queries. It is clear, however, that this approach does not scale well.

`FindStat` provides, as its main application, the following feature.

Feature. *Given (a few values of) a combinatorial statistic $\alpha : \mathcal{C} \to \mathbb{Z}$, find combinatorial maps $\phi_1 : \mathcal{C} \to \mathcal{C}'$, $\phi_2 : \mathcal{C}' \to \mathcal{C}''$, $\phi_3 : \mathcal{C}'' \to \mathcal{C}'''$, and a combinatorial statistic $\beta : \mathcal{C}''' \to \mathbb{Z}$, all in the database, such that*

$$\alpha = \beta \circ \phi_3 \circ \phi_2 \circ \phi_1 \,. \qquad \text{(search for values)}$$

The requirement of equality may be relaxed to requiring equidistribution on the finite "pieces" $\mathcal{C}_n$ of the collection $\mathcal{C} = \bigcup_n \mathcal{C}_n$. This is, for all n,

$$\sum_{c \in \mathcal{C}_n} x^{\alpha(c)} = \sum_{c \in \mathcal{C}_n} x^{\beta \circ \phi_3 \circ \phi_2 \circ \phi_1(c)} \qquad \text{(search for distribution)}$$

[2] `www.findstat.org/Mp00108`
[3] `www.findstat.org/St000021`
[4] `www.findstat.org/St000003`
[5] `www.findstat.org/St000543`
[6] `www.findstat.org/St000097`

We invite the reader to try this by visiting www.findstat.org/St000021 and to click on "search for values", and on "search for distribution". FindStat also supports a natural generalization of "search for values" and "search for distribution", and analogous features for maps are also available, see www.findstat.org/HowToUse for details.

Accessing the database

The dedicated website allows various ways of browsing the FindStat database and to use its search engine. We refer to www.findstat.org/HowToUse for instructions and to www.findstat.org/CollectionsDatabase for browsing the collections and their statistics and maps. A second, more advanced possibility is to use the interface for the computer algebra system SageMath[7]. Of course, we explicitly encourage users of other computer algebra systems to also implement FindStat interfaces.

2 Usage examples

2.1 Identifying a statistic given a few values

The original use case of FindStat is to identify a statistic given its values on a few elements of a collection. For example, the following question was raised on mathoverflow[8]:

> I've come across a function from the set of integer partitions to the natural numbers which I don't recognise but which probably ought to be familiar.

The function in question was given as $f(\emptyset) = 1$ and $f(\lambda) = \binom{i+j}{i} f(\mu) f(\nu)$, where (i, j) are the coordinates of a box such that $i + j$ is maximal, μ is obtained from λ by removing the first i rows, and ν is obtained by removing the first j columns. It does not take long to produce a few values of this statistic, for example

```
[1] => 2,
[2] => 3, [1,1] => 3,
[3] => 4, [2,1] => 6, [1,1,1] => 4,
[4] => 5, [3,1] => 8, [2,2]   => 6, [2,1,1] => 8, [1,1,1,1] => 5
```

Querying the database using these values[9] yields a handful of matches. Because so few values are given, some of these matches are *false positives*; they turn out not to match the given definition. However, we also obtain what we are looking for! The following is equivalent to the accepted answer on mathoverflow:

[7] www.findstat.org/SageInterface
[8] www.mathoverflow.net/q/132338
[9] www.findstat.org/StatisticFinder/IntegerPartitions

$$\begin{array}{rccc} \texttt{Mp00043:} & \texttt{Integer partitions} & \xrightarrow{\texttt{to Dyck path}} & \texttt{Dyck paths} \\ \texttt{Mp00026:} & \texttt{Dyck paths} & \xrightarrow{\texttt{to ordered tree}} & \texttt{Ordered trees} \\ \texttt{St000085:} & \texttt{Ordered trees} & \xrightarrow{\texttt{number of lin. extensions}} & \mathbb{Z} \end{array}$$

This means: first traverse the boundary of the Ferrers diagram of the partition from the bottom-left to the top-right to obtain a Dyck path, prepending vertical and appending horizontal steps if necessary. Then interpret the Dyck path as an ordered tree, using the traditional preorder-traversal bijection. Finally, return the number of linear extensions of this tree.

We emphasize in this example, that all involved maps and also the statistic are well-known and classical. But it is unclear at a first glance that the given definition of a statistic matches the composition

$$\texttt{St000085} \circ \texttt{Mp00026} \circ \texttt{Mp00043}\,.$$

The `FindStat` search engine enables the user to automatize such discoveries.

2.2 Identifying an equidistributed statistic

It may happen that the statistic of interest cannot be obtained as a composition of maps and a statistic known to `FindStat`. In this case one may check the relaxed condition whether an *equidistributed statistic* can be found. For example, consider the set $\mathcal{T}_n$ of (unordered, rooted) leaf-labelled binary trees with $n+1$ leaves. For a tree $T \in \mathcal{T}_n$, we are interested in the statistic measuring the distance $d(T)$ of the leaf labelled 1 from the root.

Leaf-labelled binary trees are not a collection in the database, but there is a well-known natural bijection $\mu : \mathcal{M}_n \to \mathcal{T}_n$ between $\mathcal{M}_n$, the set of perfect matchings of $\{1, \dots, 2n\}$, and these trees, see [5, Example 5.2.6]. However, querying the database for the statistic $d \circ \mu$ on perfect matchings does not yield any result.

One may then try to find an equidistributed statistic. In the case at hand, we query the database for a statistic β such that

$$\sum_{M \in \mathcal{M}_n} x^{\beta(M)} = \sum_{M \in \mathcal{M}_n} x^{d \circ \mu(M)},$$

and obtain the very promising coincidence

$$\texttt{St000838:} \quad \texttt{Perfect matchings} \xrightarrow{\texttt{number of terminal right-hand endpoints}} \mathbb{Z}\,.$$

In the following, it is more convenient to use the statistic recording the number of *initial left-hand endpoints* of a perfect matching when the vertices are written in order. It is not hard to see that this statistic, regarded as a statistic $s(T)$ on leaf labelled binary trees using the bijection μ, records the smallest label of a leaf whose sibling is also a leaf, minus 1.

Thus, it remains to find a bijection $\tau : \mathcal{T}_n \to \mathcal{T}_n$ such that $s(T) = d \circ \tau(T)$. It turns out that this is not the end of the story. In fact, using a (not completely obvious) variation of the bijection μ, one can prove, for all n, that

$$\sum_{T \in \mathcal{T}_n} x^{s(T)} y^{d(T)} = \sum_{T \in \mathcal{T}_n} x^{d(T)} y^{s(T)}.$$

2.3 Refining equidistribution results

Given two equidistributed statistics α and β, it may be interesting to find two further statistics γ and δ, such that even (α, γ) and (β, δ) have the same distribution, that is

$$\sum_{c \in \mathcal{C}_n} x^{\alpha(c)} y^{\gamma(c)} = \sum_{c \in \mathcal{C}_n} x^{\beta(c)} y^{\delta(c)}, \quad \text{for all } n.$$

In the past, such refinements were often overlooked, as witnessed by the fact that the symmetry of the joint distribution of the number of crossings and the number of nestings in perfect matchings was discovered about 20 years after the equidistribution of the two statistics was first demonstrated. Indeed, lacking a bijection that shows the equidistribution of α and β, how could one find statistics γ and δ except by luck or great intuition?

To illustrate, the following question was raised in the Viennese combinatorics seminar in November 2018:

> Are there any equidistribution results for triples of permutation statistics $(\varepsilon, \sigma, \mu)$, such that ε is Eulerian (as, for example, the number of descents), σ is a Stirling statistic (that is, equidistributed with the number of cycles) and μ is a Mahonian statistic (that is, equidistributed with the Major index)?

One may use `FindStat` to discover such coincidences. Using the `SageMath` interface, one compiles three lists containing the Eulerian statistics, the Stirling statistics and the Mahonian statistics, respectively. One then checks for each pair from the first list whether there are two statistics from the second list refining the equidistribution, by testing equality of the first few bivariate generating functions. One then repeats this procedure with the third list and arrives at the following conjecture which has, to the best of our knowledge, not appeared in the literature.

Conjecture 2.1. *Let*

- cyc *be the number of cycles*[10] *and* sal *be the number of saliances*[11] *of a permutation,*

[10] `www.findstat.org/St000031`
[11] `www.findstat.org/St000007`

- srt *be the sorting index*[12] *and* ninv *be the number of non-inversions*[13] *of a permutation, and*
- dfc *be the number of deficiencies*[14] *and* leh *be the number of repeated entries in the Lehmer code*[15] *of a permutation.*

Then

$$\sum_{\pi \in \mathcal{S}_n} x^{\mathrm{cyc}(\pi)} y^{\mathrm{srt}(\pi)} z^{\mathrm{dfc}(\pi)} = \sum_{\sigma \in \mathcal{S}_n} x^{\mathrm{sal}(\pi)} y^{\mathrm{ninv}(\pi)} z^{\mathrm{leh}(\pi)}.$$

2.4 Two concrete research examples

Guessing asymptotic behaviour

In [2], Thomas Kahle and the second author present experimental investigations of the asymptotics of permutation statistics of the following form. Assume FindStat contains a permutation statistic $\beta : \mathcal{S}_n \longrightarrow \mathbb{Z}$ with all values given for $2 \leqslant n \leqslant N$ for some N, say $N \in \{6, 7, 8\}$. One may then

- compute the generating functions $\sum_{\pi \in \mathcal{S}_n} x^{\beta(\pi)}$ and the mean and variance of associated random variable $X_\beta^{(n)}$ for $2 \leqslant n \leqslant N$, and
- use Lagrange interpolation with the computed data points to guess (Laurent) polynomial formulas for the mean and variance of $X_\beta^{(n)}$ as a function of n.

We refer to [2] for concrete results and further considerations.

Finding bijective proofs

In [3], the authors and René Marczinzik give combinatorial interpretations of several homological properties of Nakayama algebras in terms of Dyck path statistics. The starting point was that several generating functions for homological properties on Nakayama algebras were found in the OEIS. Moreover, a natural bijection between certain Nakayama algebras and Dyck paths is well-known. Using FindStat, we discovered bijections on Dyck paths that map these homological properties to classical Dyck path statistics. We refer to [3] for details.

[12] www.findstat.org/St000224
[13] www.findstat.org/St000246
[14] www.findstat.org/St000703
[15] www.findstat.org/St001298

2.5 Searching for keywords

One of the most straightforward (and useful) ways to use FindStat is to search for a keyword. In case you have forgotten how the precise definition of a statistic or map goes, chances are good that you can identify it using a suitable keyword. This could be a word occurring in its name (such as 'promotion'), an algebraic structure you recall to be necessary for the definition (such as 'character'), or perhaps a mathematician commonly associated with the statistic (such as 'Euler').

3 Status of the database

The original version of the FindStat database was initiated in August 2011 by Chris Berg and the second author. At the previous FPSAC presentation in 2014, the database contained 175 statistics. Since then, we putted a lot of effort into adding new statistics; over the past years *we added roughly one statistic per day*. We refer to www.findstat.org/Contributors for a growth chart of the combinatorial statistics in the database. As of April 2019, the database contains

1387 statistics and 146 maps on 18 collections by 70 contributors.

This allows to test any given statistic data against millions of compositions of maps ϕ_1, ϕ_2, ϕ_3 and statistics β of the form $\beta \circ \phi_3 \circ \phi_2 \circ \phi_1$ using the search engine. For articles referencing the database and for additional ways of using it in research, we refer to www.findstat.org/Citations.

3.1 Perspectives

The main challenge we need to tackle is that of scalability. We would like to make many more combinatorial collections available, and we would like to include many more combinatorial maps.

More combinatorial collections. Technically, there are rather few prerequisites necessary to add a new collection, and supporting a collection does not cost much in terms of performance and memory. However, FindStat works best with collections whose finite pieces have moderate size (with Permutations perhaps being a reasonable upper bound). Even more important is the availability of a number of natural maps, which establish connections to the other collections in the database.

More combinatorial maps. Adding more combinatorial maps poses larger technical difficulties. Currently, upon initializing the database, FindStat has to

- determine, for each collection, the set of elements that are actually considered when processing queries, and
- precompute the compositions of all combinations of (currently) 4 individual maps on these elements.

Unfortunately, adding new maps may result in a significant slowdown of this initialization process, and also a significant increase in memory usage. It is one of our main short-term goals to solve this issue.

3.2 Participation

This is a collaborative project! It is very simple to contribute to the database. Everyone is welcome to add new statistics at `www.findstat.org/NewStatistic`, suggest new combinatorial maps, enhance the wiki pages of the collections, and to propose improvements, suggestions and additional ideas.

References

[1] S. C. Billey and B. E. Tenner. "Fingerprint databases for theorems". *Notices Amer. Math. Soc.* **60**.8 (2013), pp. 1034–1039. Link.

[2] T. Kahle and C. Stump. "Counting inversions and descents of random elements in finite Coxeter groups". *Math. Comp.* (2019). to appear.

[3] R. Marczinzik, M. Rubey, and C. Stump. "A combinatorial classification of 2-regular simple modules for Nakayama algebras". 2018. arXiv:1811.05846.

[4] N. J. A. Sloane. "The On-Line Encyclopedia of Integer Sequences". *Notices Amer. Math. Soc.* **50**.8 (2003). `http://www.ams.org/notices/200308/200308-toc.html`, pp. 912–915.

[5] R. Stanley. *Enumerative Combinatorics. Vol. 2*. Cambridge Studies in Advanced Mathematics 62. Cambridge Univ. Press, 1999.