Little Idea from talking with Prof O

Here's the cute combinatorial proof of from what we were just talking about.

Let us count the number of parking functions for n spots and m cars with $m \leq n$ in two different ways.

Method 1

The standard way of counting parking functions for n spots and m cars is a modified version of Pollack's circular argument. I think this generalization is mentioned in [2, 1] and probably the Catherine Yan survey.

The key idea is to let m cars park in a parking lot with $n+1$ spots. There will be $n-m+1$ empty spots. Therefore, in a single coset of the diagonal subgroup, there will be $n-m+1$ resulting preference lists which result in the last spot being empty. These preference lists are then parking functions for m cars and n spots. This gives the following count:

$$(n-m+1)(n+1)^{m-1}$$

Method 2

Consider taking a parking function for m cars and m spots, and inserting additional empty spots to the parking outcome until the result is a parking function for m cars and n spots. Note that additional spots can only be inserted at some locations. For example, an empty spot cannot be inserted between spots 2 and 3 for the parking function 122 since the resulting preference list wouldn't leave the empty spot empty. These empty spots can only be inserted after break points or at the very beginning. Let us define an ordered partition λ associated with a parking function π which describes the lengths of the prime components. Therefore all parking functions can be described by assigning each part of λ a particular prime parking function of the given length.

For example if $\lambda = 313$, then we need 3 prime parking functions of length $3, 1$, and 3 respectively. Once we have chosen these parking functions, we will shuffle them to get an overall parking function. If we chose 111 , 1, and 112 as the prime parking functions, the result would be any reordering of 1114556. Now the possible places to insert an empty spot are in the resulting parking configuration are after the following spots: $0, 3, 4, 7$. Note that these are the partial sums of λ, and give the following 4 possibilities: $2225667, 1115667, 1114557, 1114556$

Note that λ is now an **ordered** partition, which is distinct but easily related to its cousin the unordered partition. Just multiply by the number of orderings of λ. ex: for $\lambda = 211$ there are 3 orderings.

Now, let us turn little construction of arbitrary parking functions into an expression counting them.

$$\sum_{\lambda \models m} \binom{m}{\lambda} \left(\!\!\binom{|\lambda|+1}{n-m}\!\!\right) \prod_{p \in \lambda} (p-1)^{p-1}$$

Here are the origins of each of the terms:

- The summation is over all **ordered** partitions of m which will be the prime parking function segments. If you instead want to sum over normal partitions, you have to multiply by the number of orderings for λ, which is also reasonable.
- The $\binom{m}{\lambda}$ comes from choosing how the different prime parking functions will be ordered for the cars when they are shuffled together in line. This comes from sequentially choosing each part of the partition. For example $\binom{4}{211} = \binom{4}{2}\binom{2}{1}\binom{1}{1}$
- The $\left(\!\binom{|\lambda|+1}{n-m}\!\right)$ comes from choosing the location of the empty spots
- The product over all of the different parts

**Note - I have been fast and loose with some of the details like showing that any parking function for $n \neq m$ can be constructed like this, and that no parking function for $n \neq m$ is created twice, but these things are true.

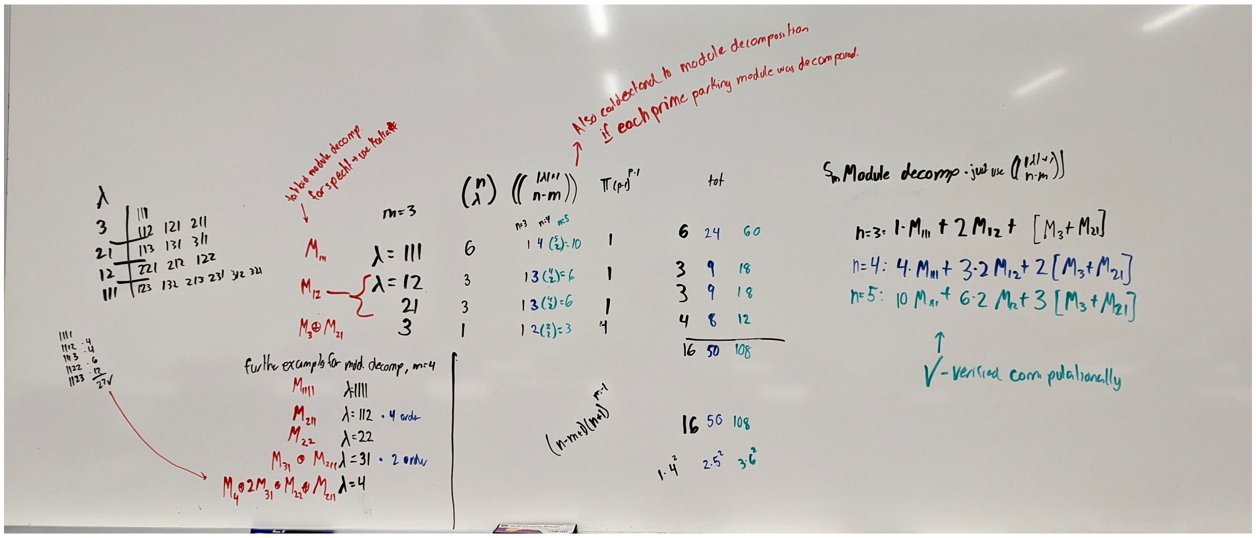
This is a combinatorial *would be proof if I stopped going on tangents* that

$$\begin{aligned}(n-m+1)(n+1)^{m-1} &= \sum_{\lambda \models m} \binom{m}{\lambda} \left(\!\!\binom{|\lambda|+1}{n-m}\!\!\right) \prod_{p\in\lambda}(p-1)^{p-1} \\ &= \sum_{\lambda \vdash m} o(\lambda) \binom{m}{\lambda} \left(\!\!\binom{|\lambda|+1}{n-m}\!\!\right) \prod_{p\in\lambda}(p-1)^{p-1}\end{aligned}$$

Note the two expressions are different since the first one λ has an order, and the second one, λ doesn't have an order, and the additional factor of $o(\lambda)$ is to account for all of the orderings of the partition λ

Verification

I double checked that this was true for some small examples.


λ
3 | 111
21 | 112 121 211
12 | 113 131 311
111 | 221 212 122
123 132 213 231 312 321
1111
1112 · 4
1113 · 4
1122 · 6
1123 · 12
27 ✓
to tie module decomp for specht → use Kostka
m=3
M_{111} λ = 111
M_{12} λ = 12
21
$M_3 \oplus M_{21}$ 3
further examples for mod. decomp, m=4
M_{1111} λ=1111
M_{211} λ = 112 · 4 order
M_{22} λ=22
$M_{31} \oplus M_{211}$ λ=31 · 2 order
$M_4 \oplus 2M_{31} \oplus M_{22} \oplus M_{211}$ λ=4
Also could extend to module decomposition if each prime parking module was decomposed.
$\binom{n}{\lambda}$ $\left(\!\binom{|\lambda|+1}{n-m}\!\right)$ $\prod(p-1)^{p-1}$ tot
n=3 n=4 n=5
6 | 1 4 $\binom{5}{2}$=10 | 1 | 6 24 60
3 | 1 3 $\binom{4}{2}$=6 | 1 | 3 9 18
3 | 1 3 $\binom{4}{2}$=6 | 1 | 3 9 18
1 | 1 2 $\binom{3}{2}$=3 | 4 | 4 8 12
16 50 108
$(n-m+1)(n+1)^{m-1}$
16 56 108
$1\cdot4^2$ $2\cdot5^2$ $3\cdot6^2$
S_m Module decomp · just use $\left(\!\binom{|\lambda|+1}{n-m}\!\right)$
n=3: $1\cdot M_{111} + 2M_{12} + [M_3 + M_{21}]$
n=4: $4\cdot M_{111} + 3\cdot2 M_{12} + 2[M_3 + M_{21}]$
n=5: $10 M_{111} + 6\cdot2 M_{12} + 3[M_3 + M_{21}]$
✓ - verified computationally

Along the way, I noticed that this logic can extend to decomposing modules.
If you understand how prime parking modules decompose as S_m modules, then you can construct how parking modules for m cars and n spots decompose.

As in

$$PF_{n,m} \cong_{S_m} \bigoplus_{\lambda \vdash m} o(\lambda) \left(\!\!\binom{|\lambda|+1}{n-m}\!\!\right) PPF_\lambda$$

Note that $o(\lambda)$ is the number of orderings of the normal partition λ. Additionally, the PPF_λ part of this expression is hiding what used to be a product, and is still a product which takes place between the tabloid representations. Let μ_λ be the normal tabloid representation. As an example, note that because the prime parking functions of length 3 are $111, 112, 121, 211$, there are 2 orbits of S_m, and the module decomposition into tabloid representations is the following:

$$PPF_3 = M_3 \oplus M_{21}$$

Therefore if we wanted to know PPF_{33}, we would just need to find the following by concatenating partitions:

$$PPF_3 \cdot PPF_3 = (M_3 \oplus M_{21}) \cdot (M_3 \oplus M_{21}) = M_{33} \oplus 2M_{321} \oplus M_{2211}$$

Question: I am unclear on how this relates to tensor products, but it seems like they should be related somehow (or I should be able to explain why not)

A note on the associated maps

There are a couple of implicit maps that could be more explicit, just so it is clear what their properties are.

- Preference list to prime segment ordered partition. Surjective (I think), not injective
- Restricted to parking functions. Surjective, for λ, the number of times the function maps to λ is $\left(\!\binom{|\lambda|+1}{n-m}\!\right) \prod_{p\in\lambda}(p-1)^{p-1}$
- Restricted to parking functions where $n = m$. Surjective, for λ, the number of times the function maps to λ is $\prod_{p\in\lambda}(p-1)^{p-1}$
- Parking function for $n > m$ to parking function for $n = m$. Surjective, not injective
- Parking function for $n > m$ to parking function for $n = m$ with fixed spots empty. Injective, not surjective

References

[1] Peter J. Cameron, Daniel Johannsen, Thomas Prellberg, and Pascal Schweitzer. Counting defective parking functions, 2008.

[2] Richard P. Stanley and Mei Yin. Some enumerative properties of parking functions, 2023.