Counting finite categories

Geoff Cruttwell
Mount Allison University
(joint work with Rejean Leblanc)

AMS Fall 2014 Sectional Meeting Fall 2014
New Directions in Category Theory special session
Halifax, October 19th, 2014

Outline

The question we are interested in: how many categories are there with n morphisms? (Up to isomorphism).

- How far can we calculate this number exactly, either by hand or with computer assistance? (And store the relevant categories).
- Can we get a formula, either precise or asymptotic, for the growth of this function?
- Can we say anything about specific types of categories, eg, Cauchy-complete categories?

Purely for interest's sake, no application in mind!

Known results

These were the previous known results:

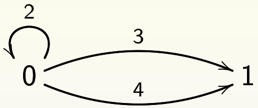
	Total	1 object	2	3	4	5	6
1 arrow	1	1					
2	3	2	1				
3	11	7	3	1			
4	55	35	16	3	1		
5	329	228	77	20	3	1	
6	2858	2,237	485	111	21	3	1
7		31,559					
8		1,668,997					
9		3,685,886,630					
10		$\sim 1.05x10^{15}$					

Finding categories as a CSP

- One can formulate the problem of "finding all categories with n morphisms" as a constraint satisfaction problem: a set of variables and constraints they must satisfy.
- For this, we use the "arrows only" version of category: objects are represented by their identity arrows.
- The category represented in this way is then simply an $n \times n$ table of how the n arrows compose, with each value in the table either another numbered arrow or a dummy value if the pair is non-composable.
- The constraints are the domain/codomain/identity/associativity axioms for a category.

Example category with five morphisms

The category:



with: $2 \circ 2 = 0$, $3 \circ 2 = 4$, $4 \circ 2 = 3$, is represented as:

	0	1	2	3	4
0	0	*	2	3	4
1	*	1	*	*	*
2	2	*	0	4	3
3	*	3	*	*	*
4	*	4	*	*	*

Minion

- There are many constraint satisfaction solvers available.
- We used Minion: a relatively recent constrant satisfaction solver that focuses on speed at the expense of some flexibility.
- Like any constraint satisfaction solver, it has many optimizations to solve CSP's; both algorithmic and hardware related.

Trimming isomorphisms

- One problem: the above process will give us many isomorphic copies of the same category.
- To resolve this problem, we run through the output we are given, and only select the categories which are lexicographically-least in each isomorphism class.
- That is, for each category given to us by Minion, we run through all permutations of the arrows in that category, and re-arrange the original category's table according to that permutation. If the original category is lexicographically greater than the permuted version, we discard it.
- This leaves only one category in each isomorphism class.

Additional optimizations, part I

- This process can find all categories with 7 morphisms in a reasonable amount of time, but then starts to take days for 8 morphisms, so additional optimization is required.
- **First optimization**: count the number of categories with n morphisms and k objects. Can remove all associativity constraints for identity morphisms. This gives all categories with 8 morphisms and most of 9.
- **Second optimization**: count only the connected categories with n morphisms and k objects. Non-connected categories can be found as a function of the previous lower connected counts. This gives counts up to 9 and 10 for all but the 2-object case.

Additional optimizations, part II

- **Third optimization**: count only the categories with 2 objects that have at least a certain number of non-endomorphic arrows (in practice, at least 3 or 4 non-endomorphic arrows).
- This avoids counting, for example, categories with two monoids with a single arrow between them: these can easily be counted by hand, but add a lot of processing time when running Minion.
- We then seperately run Minion instances with specific directed graphs with, say, 3 arrows between two monoids.

Updated table

	1	2	3	4	5	6
1	1					
2	2	1				
3	7	3	1			
4	35	16	3	1		
5	228	77	20	3	1	
6	2237	485	111	21	3	1
7	31559	**4013**	**716**	**127**	**21**	**3**
8	1668997	**47648**	**5623**	**862**	**131**	**21**
9	3.68×10^{10}	**1868157**	**60201**	**6739**	**926**	**132**
10	$\sim 1.05(10^{14})$	$\sim \mathbf{3.69(10^{10})}$	$\sim \mathbf{6(10)^5}$	**65922**	**7349**	**945**

This is as we far as we can go until someone counts the 11 monoids.

Monoids vs. Categories

Arrows	Monoids	All categories	Ratio
1	1	1	1.0
2	2	3	0.66
3	7	11	0.63
4	35	55	0.64
5	228	329	0.69
6	2,237	2,858	0.78
7	31,559	36,440	0.87
8	1,668,997	1,723,286	0.97
9	3,685,886,630	3,687,822,810	0.999
10	1.05986×10^{14}	1.05982×10^{14}	0.9999

Can we explain this?

Almost all semigroups are 3-nilpotent

- Almost all semigroups are 3-nilpotent: there is an element 0 with for any x, $x0 = 0x = 0$ and for any x, y, z, $xyz = 0$ (Kleitman, Rothschild, Spencer 1976).
- Why? These are very easy to construct: to construct such a semigroup on a set A, pick a subset B, an element $0 \in B$, and define xy to be 0 if x or y is in B and an arbitrary element of B otherwise. Such an operation is automatically associative.
- These types of semigroups overwhelm all other possibilities as the order of the semigroups grow.

Semigroup and monoid counts

- One can get a precise count of the number of 3-nilpotent semigroups of order n up to isomorphism (Distler and Mitchell 2012) and thus obtain an asymptotic formulae for the number of semigroups of order n up to isomorphism.
- Moreover, almost all monoids are semigroups with an identity attached (Koubek and Rodl, 1985).
- Thus one has asymptotic counts for both semigroups and monoids of order n.

Conjecture: almost all categories are 3-nilpotent semigroups

- Thus, if we can prove that almost all categories with n morphisms are monoids, then almost all categories will actually be 3-nilpotent semigroups with an identity adjoined.
- The direct known formula for the the number of 3-nilpotent semigroups, and its astonishing growth rate, should allow us to prove this.
- We would thus have an asymptoptic count for the number of categories with n morphisms (either up to isomorphism or up to equivalence).

Cauchy-complete

- Most interest in finite categories comes from looking at their associated presheaf categories.
- Thus, it makes sense to also look at the counts of Cauchy-complete categories.
- This has the additional advantage of removing all monoids which are not groups, and so avoids the ridiculous numbers which come with the 3-nilpotent semigroups.

Cauchy-complete table

	Total	1	2	3	4	5	6	7	8	9	10
1	1	1									
2	2	1	1								
3	4	1	2	1							
4	11	2	6	2	1						
5	25	1	12	9	2	1					
6	63	2	23	25	10	2	1				
7	163	1	45	69	35	10	2	1			
8	451	5	98	178	119	38	10	2	1		
9	1311	2	278	457	371	151	39	10	2	1	

Conclusions

- We have counted and stored all categories with 10 morphisms or less up to isomorphism: this is about the limit with current technology using our techniques.
- One can get an asymptotic count for the number of categories with n morphisms, based on the idea that almost all categories are monoids, and almost all monoids are 3-idempotent semigroups.
- A more interesting question, then, is how many Cauchy-complete categories there are: further investigation needed.

References

References:

- Distler, A. and Mitchell, J. The number of nilpotent semigroups of degree 3. *Electronic Journal of Combinatorics*, Vol. 19 (2), pg. 51–64, 2012.
- Kleitman, D., Rothschild, B., and Spencer, J. The number of semigroups of order n. *Proceedings of the American Mathematical Society*, Vol. 53 (1), pg. 227–232, 1976.
- Koubek, V. and Rodl,V. Note on the number of monoids of order n. *Commentationes Mathematicae Universitatis Carolinae*, Vol. 26 (2), pg. 309–314, 1985.