The Banach space database project:  
a presentation

Noé de Rancourt

June 12, 2020

1 The general idea

This project is inspired by the book *Counterexamples in Topology*, by Steen and Seebach. This book contains a list of numerous classical and pathological topological spaces along with their main properties. At the end of the book, a table enables one to find, given a boolean combination of properties, if it is satisfied by some of the listed spaces. Do I need to find a topological space that is zero-dimensional, separable, but not second-countable? A look in the table, and I see that the strong parallel line topology satisfies this requirements.

A large part of the geometry of Banach spaces also relies on finding and studying counterexamples. Spaces that were at once constructed to provide counterexamples to some properties, such as the Tsirelson space, the Schlumprecht space, or HI spaces, are now central in Banach space theory. I have, personally, found myself several times in the situation of wondering whether a given combination of properties was satisfied by some known space or proved impossible, without really knowing where to look for answers. I think that a resource like *Counterexamples in Topology* would be useful for many of us.

My project is to build a database of known Banach spaces, classical or pathological, and of usual properties of Banach spaces. This database, freely available on the web, would be equipped with a search engine allowing one to find, given a combination of properties, a list of known spaces satisfying these properties.

2 How I see this

I explain here in more details how I see such a database. I am of course open to modifications.

We have a “wiki part”, containing a page for each known example of Banach space (or known family of Banach spaces), and a page for each known property. The page of a given space would give a brief presentation of the space, and list its main properties. There is no need to be too precise, to give detailed constructions of the spaces, or proofs of its properties: all of this can be found in books and articles, and the goal is not to write a new handbook. In my sense, the most important is to give a reference for the construction of the space, and for each property, a reference for the proof of this property (or a short argument if it is very easy): thus, the reader knows where to look for more details if needed. Similarly, the wiki page of a given property could give general information about it (in what context it has been introduced, its interest...),

1
Aside to the “wiki part”, we have a “search engine part”. The idea is that each space, and each property, listed on the wiki, is also listed on a database, and for each pair (space, property), the database can contain yes, no, undecidable, or unknown. The database is equipped with a search engine in which we can enter a boolean combination of properties, and which answers with the list of spaces satisfying this boolean combination. For instance, if I enter “does not contain $c_0$”, “does not contain $\ell_1$”, and “has no reflexive subspaces”, the search engine returns me a list containing the Gowers tree space, with a link to its wiki page where I can find a reference for its construction.

An other possible feature (but probably more complex to set up) of the search engine would be, when the combination of properties entered by the user is proved incompatible, to return a reference for the proof of this incompatibility. For instance, if I enter “has an unconditional basis”, “is not reflexive”, “does not contain $c_0$”, and “does not contain $\ell_1$”, the engine answers me that it is impossible and gives me a reference to a proof of James’ theorem.

Of course, filling in such a database would be a huge work, and I don’t expect that the few people maintaining it would have enough knowledge to know about all interesting Banach spaces that have been built in the history. So my idea is to make something collaborative, and peer-reviewed. Our role, as people maintaining the database, would not be to write entries, but rather to act as editors. Anybody could submit a new wiki page, a change in an existing one, a new correspondance space $\leftrightarrow$ property, etc., with references included, and our role would be to review these submissions, in order to accept or reject them.

3 Difficulties and problems

Here, I list the possible problems and difficulties that could occur in course of setting up this project. The obvious one is to code and maintain the database, and I don’t have any skills in that matter. But there are also non-technical difficulties.

One of these difficulties is that the number of spaces and of properties we would like to enter into the database is of course infinite, and even uncountable. For each $1 \leq p \leq \infty$, there is an $\ell_p$. The main properties of these spaces differ according whether $p = 1$, $1 < p < 2$, $p = 2$, $2 < p < \infty$, and $p = \infty$. Of course, one wiki page is enough for the family of all the $\ell_p$’s, detailing in the text which properties are satisfied for which $p$’s. It would also be tempting to think, at first glance, that five database entries, corresponding to the five cases listed above, would be enough. But think to the following database request: “find spaces having cotype 3”. The search engine should then be able to return as a result the $\ell_p$’s for $p \leq 3$, but not those for $p > 3$. This case, and many others, raise the following tricky questions: what kind of requests do we want the user to be able to make? Only conjunctions of properties without parameters, or any boolean combination of properties without parameters, or combinations of properties that are possibly parametrized (by real numbers, ordinals, etc.), maybe with quantifiers over these parameters? In the last case, in which language will the interface allow the user for such requests? And in which way should the database answer to such requests?

On the technical side, a difficulty comes if we choose to implement the possibility of redirecting to theorems in case of incompatibility of the requested properties. As an example, suppose that

---

1An example I see that would require such a task is finding a space that is not isomorphic to a Hilbert space, but having type $p$ and cotype $q$ for every $p < 2 < q$. Making such requests possible would be useful.
we have four properties $A$, $B$, $C$ and $D$, and the following three well-known theorems, proved in separate papers:

**Theorem 1.** $A \implies B$.

**Theorem 2.** $C \implies D$.

**Theorem 3.** $B$ and $D$ are incompatible.

We probably just want to enter well-known theorems in the database, and not their numerous consequences, otherwise the number of theorems to enter will become huge. So, the three latter theorems will probably be in the database, but not their consequence “$A$ and $C$ are incompatible”. Nevertheless, if I ask the database “Is there a space simultaneously satisfying $A$ and $C$”, I want it to answer me “No, and it’s because of Theorems 1, 2 and 3”. This implies that the search engine should support propositional logic, should be able to determine the satisfiability of a propositional equation given constraints, and should be able to return a small set of obstructions if unsatisfied.

If, as described above, we want to allow requests involving properties parametrized by real/ordinal parameters, then we will even need our software to support some fragment of first-order logic for simple structures, for example real numbers as an ordered field\(^2\). This requirement probably remains even if we don’t implement the “theorem-finding” functionality, for instance we will need it to express such things as “having all cotypes $> 2$”.

### 4 Next steps

As I said, I don’t think I’m able, and I don’t think I want, to work on this project alone. If you’re interested to work on this with me, don’t hesitate! A discord server has been set up to discuss about this project and organise ourselves; you can join it by the link https://discord.gg/tQqtWJm.

The first step will probably be to figure out more precisely how we want the interface to be, how we want to deal with the logical problems presented in the last section on the user side, etc. Then we’ll need to choose a software to implement this concretely, and to find someone able to do this. For my part, I know nothing about databases, and more generally I have almost no skills in coding.

This project has been discussed on the facebook group “Banach-space theorists worldwide” and during the Banach spaces webinar (I encourage you to join both!) People made me the following suggestions:

- Some suggested me to use SQL. Others told me that this is an old technology.
- Another option is to use the database technology from Amazon Web Services, which is more powerful.
- An option is to do something similar as the n-Category wiki. I looked at it, but it seems to have wiki functionalities only and is nothing like a database, so in my opinion it’s not a good option.

\(^2\)And not only as an ordered set, because operations are also useful sometimes! You could want, for instance, to express conjugate exponents...
• TiddlyWiki could be a god starting point. This is mostly a wiki, but it includes a tagging system allowing one to do searches, so it could be used as some kind of database (this would only allow very basic search functionalities, though).

• Maybe coding and maintaining such a database, on the technical side, would be a full-time work. It could be wise to try to get money from a grant and to hire a specialist, who would be able to deal with all the technical side of the project.