Model Catiegories

Aliaksandra Novik aka Sasha N

Outline

The main motivation for studying the theory of model categories is that they provide an abstract framework for doing homotopy theory later. What does this mean? Well, in different categories, one can define maps called weak equivalences, which are not isomorphisms but still powerful enough to preserve many important properties. The main problem is that they are not necessarily invertible. What one wants to do is somehow formally invert them and obtain a new category in which weak equivalences are isomorphisms. Unfortunately, we cannot do this in a naive way since we lose track of maps in the new category we obtain. This is exactly where model categories come into play. They provide extra information on the structure of morphisms, allowing us to actually perform the trick described above, and we call the result the homotopy category.

Plan (at least we hope so)

Class 1: We will begin by introducing the definition of model categories on a purely categorical level, without requiring any advanced prerequisites. This definition will be followed by two main examples: model categories defined on the category of topological spaces and on the category of chain complexes.

Additionally, we will include some exercises aimed at enhancing understanding of the topic.

Class 2: In this session, we will define the homotopy category and demonstrate its existence for any model category. This is a key result that underscores the significance of the model structure. Through this construction, we will hopefully gain insight into why the model structure was necessary.

Prerequisites

Honestly, no particular difficult prerequisites are required. You don't need to be an expert in category theory, topology, or homotopy theory. Having some knowledge in these fields would help you understand the examples provided and follow the lectures more easily. However, we will strive to make the lectures self-consistent if time permits and provide brief summaries of necessary topology concepts as well.