Index notation in Lean 4

Joseph Tooby-Smith

Reykjavik University

October 29, 2024

Abstract

Index notation is a tool commonly used in physics to manipulate tensors. In physics, we use index notation for three different types of tensors: Einstein tensors (e.g., ordinary vectors and matrices), Lorentz tensors, and Van der Waerden tensors. In this paper, we discuss how these are implemented in Lean 4 using a general mathematical theory based on category theory, and related to the notation of an operad.

1. INTRODUCTION

A previous work by the current author, presented the first steps of digitalizing (or formalizing) results from high energy physics into the interactive theorem prover Lean 4. This project is called HepLean. Lean is a programming language whose syntax looks similar to what we pen-and-paper mathematics, and is used to write and automatically check statements of definitions and theorems, and proofs of theorems. HepLean has four main motivations: js: sorry

When writing and proving results on paper, the physicists is accustumed to using notation, and no notation abounds in physics more than index notation. Thus as part of the larger HepLean project, index notation has been implemented by the author in to Lean 4. Any such implementation must satisfy two basic requirements:

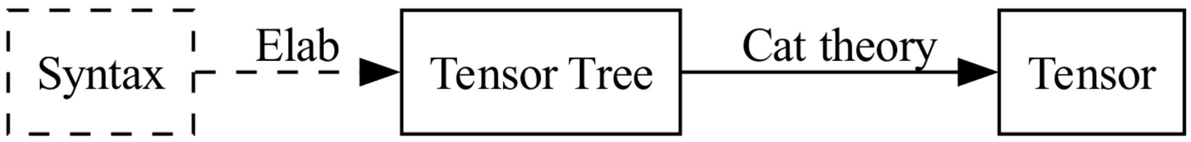
1. Must be mathematically rigourous.
2. Must be easy to use.

The first requirement is a consequence of Lean being a proof assistant, and not able to accept anything else but formally defined results. The second requirement is due to the fact that many other results will depend on the implementation of index notation. We beleive the implementation dicussed within this paper does satisfy both of these requirements.

- Notational conventions abound in physics, and non-more so then index notation.
- Notation, in general, is a way for the writer to compactly write down a term in a mathematical expression. The reader can implicitly unfold or elborate the compactly written term back into its full underlying meaning.
- Index notation is a compact way of writing expressions involving tensors.
- With tensors there is a notion of contraction, evaluation and permutation.

- All of these notions can defined independently of index notation.
- Index notation is a way of writing these operations in a compact way.
- In a previous work by the current author, the first foray of formalizing high energy physics in Lean 4 was undertaken, in a project called ‘HepLean’.
- One aim of that project is to make Lean easier for the high-energy phsycisists to use.
- Motiviated by this aim, we have implemented index notation in Lean 4.
- In this paper, we will discuss how this is done.
- This is, of course, not the first paper dicussing the implmentation of index notation in a programming language. However, Lean, being a proof assistant, has a different set of requirements.
- In particular, Lean has to be provided a proof of everything.
- We also believe that the underlying mathematics used to implement index notation here is novel.

2. IMPLEMENTATION OF INDEX NOTATION INTO LEAN 4



3. COMPLEX LORENTZ TENSORS

There are three different species of tensors that physicists deal with:

- Einstien tensors (i.e., ordinary vectors and matrices transforming under $SO(n)$)
- Real Lorentz tensors (i.e., tensors in a real vector space transforming under the Lorentz group)
- Complex Lorentz tensors (i.e., tensors in a complex vector space transforming under the $SL(2,\mathbb{C})$)

In this paper, since it is the most complicated, we will use the complex Lorentz tensors as an example.

In a tensor species there are different kinds of indices. We shall call the different kinds of indices ‘colors’. For example, for complex Lorentz tensors there are six colors of indices. In Lean we put these colors into an inductive type

```
inductive Color
  | upL : Color
  | downL : Color
  | upR : Color
  | downR : Color
  | up : Color
  | down : Color
```

3.1. SYMMETRIC AND ANTI-SYMMETRIC TENSOR

Let $S_{\mu\nu}$ be complex Lorentz tensors where μ and ν are 4-vector indices. The corresponding

```
(S : complexLorentzTensor.F.obj (OverColor.mk ![Color.down, Color.down]))
```

To explain this notation let us work from the right to the left.

4. FUTURE WORK

REFERENCES