Instructions for Using `EFTGen`

Joshua Rosaler

June 30, 2020

To use `EFTGen`, you should have Python 3 installed on your computer. Python comes preinstalled on most recent Macs, although some come with Python 2, with which the program will not work. To install Python 3, visit https://www.python.org/downloads/. You may also need to install a few dependencies by entering the following into the command line:

```
>> pip3 install itertools
>> pip3 install deepcopy
>> pip3 install itertools
>> pip3 install frozendict
```

To run `EFTGen`, navigate into the directory called “EFTGen" from your console and run the command

```
>> python3 main.py
```

You will then be prompted to input the mass dimension for which you would like `EFTGen` to generate an operator basis. In practice, this works up to mass dimension 15, which takes about an hour on an ordinary laptop. After entering the mass dimension, you will be prompted with several yes/no questions concerning which criteria you would like to use to reduce the operator basis, and also how explicit you want `EFTGen` to be in outputting contents and showing the progress of its operation. `EFTGen` will then display the operator basis, along with any other output requested by the user, in its simple, human-readable encoding scheme. The output will both be displayed on the user's console and written to a `.txt` file whose name begins with the string 'operators_'. `EFTGen`'s encoding scheme for operators is explained below, but can be guessed easily enough by inspecting the given examples.

1 How to Read `EFTGen` Output

A Lagrangian term in `EFTGen` is encoded as a triple of data structures:

- `field_multiset`: since an operator may contain multiple copies of each field type and multiple covariant derivatives, the assortment of basic objects that compose to form the operator may be specified formally as a multiset (rather than as a set, which does not allow for multiple instances of each of its elements). Such objects will be referred to below as "field/derivative multisets." A multiset can be encoded as a hash map, or dictionary, in which the keys of the dictionary are strings designating the various types of object - here, a covariant derivative, or one of various field types - and the values in the dictionary specify the number of copies of the object type. In the context of an Abelian gauge theory considered here, the field types consist of a gauge field strength tensor and the five types of Dirac bilinear.

- `derivative_assignment`: an array or list specifying the number of derivatives acting on each copy of each field type, referred to below as a "derivative assignment."

- `contraction_multiset`: a hash map or dictionary specifying the manner in which Lorentz indices are contracted, referred to below as a "full contraction" or "contraction multiset." For the case of a non-Abelian gauge theory, an additional dictionary of contractions encoding the contraction of gauge indices would be needed.

For a given field/derivative multiset, there are generally multiple inequivalent ways to assign derivatives to fields. Likewise, for each field/derivative multiset and derivative assignment, there are generally multiple inequivalent ways to contract the free indices of the fields and derivatives.

`EFTGen` generates operators by specifying these three objects for each operator that it outputs. The first two of these objects, the field/derivative multiset and derivative assignment, specify an operator with uncontracted Lorentz indices. This operator in turn can be factorized as a product of Lorentz tensors, and the indices of each such tensor can in turn be divided into groups of indices where indices in a group are interchangeable. Lorentz contractions are specified as pairs of integers, where the integers designate the groups of indices that are contracted, and the groups are numbered according to an arbitrarily chosen ordering of field types.

To give a sense of the translation between ordinary tensor notation and `EFTGen`'s notation, which is explained in further detail below, consider the following dimension-7 operators, which are shown both in tensor notation and in the `EFTGen` format.

$(\bar{\psi} i D_\mu \psi)(\bar{\psi} \gamma^\mu \psi)$

```
CONSTITUENT FIELDS:
{'D': 1, 'S': 1, 'V': 1}
DERIVATIVE ASSIGNMENT:
(((0, 1),), ((0, 0),))
```

```
GROUPING OF LORENTZ OBJECTS:
[['S', 'D'], ['V']]
LORENTZ INDEX CONTRACTIONS:
{(1, 2): 1}
```

$F_{\mu\nu}F^{\mu\nu}(\bar{\psi}\psi)$

```
CONSTITUENT FIELDS:
{'F': 2, 'S': 1}
DERIVATIVE ASSIGNMENT:
((0, 0), ((0, 0),))
GROUPING OF LORENTZ OBJECTS:
[['F'], ['F'], ['S']]
LORENTZ INDEX CONTRACTIONS:
{(0, 1): 2}
```

$F_{\mu\nu}[(D_\alpha\bar{\psi})\sigma^{\mu\nu}(D^\alpha\psi)]$

```
CONSTITUENT FIELDS:
{'D': 2, 'F': 1, 'T': 1}
DERIVATIVE ASSIGNMENT:
((0,), ((1, 1),))
GROUPING OF LORENTZ OBJECTS:
[['F'], ['D', 'T', 'D']]
LORENTZ INDEX CONTRACTIONS:
{(1, 3): 1, (0, 2): 2}
```

$(D_\mu D^\mu\bar{\psi})(D_\nu D^\nu\psi)$

```
CONSTITUENT FIELDS:
{'D': 4, 'S': 1}
DERIVATIVE ASSIGNMENT:
(((2, 2),),)
GROUPING OF LORENTZ OBJECTS:
[['DD', 'S', 'DD']]
```

```
LORENTZ INDEX CONTRACTIONS:
{(2, 2): 1, (0, 0): 1}
```

$(D_\mu D_\nu \bar{\psi})(D^\mu D^\nu \psi)$

```
CONSTITUENT FIELDS:
{'D': 4, 'S': 1}
DERIVATIVE ASSIGNMENT:
(((2, 2),),)
GROUPING OF LORENTZ OBJECTS:
[['DD', 'S', 'DD']]
LORENTZ INDEX CONTRACTIONS:
{(0, 2): 2}
```

In these examples we see that the derivative and field contents of the operator are encoded as a dictionary specifying the multiplicity of each type of element. In the simple example of an Abelian gauge theory of a single Fermi field, the different types of element are the covariant derivatives `'D'`, gauge field strength tensors `'F'`, scalar Dirac bilinears `'S'` (representing $\bar{\psi}\psi$), vector Dirac bilinears `'V'` (representing $\bar{\psi}\gamma^\mu\psi$), tensor Dirac bilinears `'T'` (representing $\bar{\psi}\sigma^\mu\psi$, pseudovector Dirac bilinears `'V_p'` (representing $\bar{\psi}\gamma^\mu\gamma_5\psi$, and pseudoscalar Dirac bilinears `'S_p'` (representing $\bar{\psi}i\gamma_5\psi$. A derivative assignment indicates how the covariant derivatives are distributed among the various copies of the various fields. `EFTGen` relies on a specific ordering of the field types to encode the distribution of derivatives: namely, `'F"`, `'S'`, `'V'`, `'T'` `'V_p'`, `'S_p'`. In the second example, the derivative assignment $((0,),((1,1),))$ indicates that there are no derivatives acting on F, one derivative acting on the $\bar{\psi}$ part of `'T'`, and one derivative acting on the ψ part of `'T'`. There is an extra layer of parentheses where the distribution of derivatives among copies of Dirac bilinears is concerned, to allow for different distributions between $\bar{\psi}$ and ψ of each copy of each bilinear.

To generate Lorentz contractions, `EFTGen` employs a special encoding of a derivative assignment that groups together derivatives and field objects whose indices can be freely interchanged without changing the generated operator. Consider the following example

```
field_multiset = { 'D':12, 'F':4, 'V':2, 'T':1 }

derivative_assignment = ((2,2,2,0), ((0,1),(0,1)), ((1,3)))
```

Together these objects encode the structure of an operator with uncontracted Lorentz indices, which can be factorized into independent Lorentz tensors. The factorization can be written schematically as

$$([D^2][F])([D^2][F])([D^2][F])([F])(\bar{\psi}[V][D]\psi)(\bar{\psi}[V][D]\psi)([D]\bar{\psi}[T][D^3]\psi) \tag{1}$$

Lorentz contractions are then between groups of interchangeable indices associated with different Lorentz objects in squared brackets, where each group has a Lorentz rank associated with the number of interchangeable indices. To encode the grouping, `EFTGen` uses nested lists, in this case

$$[[D^2, F], [D^2, F], [D^2, F], [F], [V, D], [V, D], [D, T, D^3]]. \tag{2}$$

Now we index the 14 basic operators corresponding to the groups of interchangeable Lorentz indices, $D^2, F, D^2, F, D^2, F, F, V, D, V, D, D, T, D^3$ in order from 0, 1, 2, ..., 13. To encode the set of contractions,

$$(D_{\mu_1}D^{\mu_1}F^{\mu_2\mu_3})(D_{\mu_4}D_{\mu_5}F_{\mu_2\mu_3})(D^{\mu_4}D_{\mu_6}F^{\mu_5\mu_7})(F_{\mu_7\mu_8})(\bar{\psi}V^{\mu_6}D^{\mu_{10}}\psi)(\bar{\psi}V_{\mu_9}D_{\mu_{10}}\psi)(D_{\mu_{11}}\bar{\psi}T^{\mu_8\mu_9}D^{\mu_{11}}D_{\mu_{12}}D^{\mu_{12}}\psi) \tag{3}$$

we write

$$[(0,0),(1,3),(1,3),(2,4),(2,5),(4,7),(5,6),(6,12),(8,10),(9,12),(11,13),(13,13)]. \tag{4}$$

We encode each contraction as a pair of integers in non-decreasing order. Since contractions can be occur multiple times, a set of contractions can be written in dictionary form to indicate the multiplicity of each individual contraction. In the case at hand, there is only one repeated contraction, and this dictionary reads,

{(0,0):1 (1,3):2, (2,4):1, (2,5):1, (4,7):1, (5,6):1, (6,12):1, (8,10):1, (9,12):1, (11,13):1, (13, 13):1}.

It is worth noting that even though covariant derivative operators acting on the same field do not commute - since $[D_\mu, D_\nu] = -ieF_{\mu\nu}$ - it is possible for our purposes here to treat them as if they do commute, and therefore to treat all indices associated with covariant derivatives acting on the same field as interchangeable. The reason is that in commuting one covariant derivative past another, one picks up an additional term in which the two covariant derivatives are replaced by $F_{\mu\nu}$. But this term will already be contained among the set of operators resulting from the generation of all field multisets and derivative assignments at the specified mass dimension. Thus, while an operator resulting from rearrangement of covariant derivatives acting on the same field is generally distinct from the original operator, the former can be expressed as a sum of the latter and additional terms with fewer derivatives that are already in the basis. Therefore, rearrangements of covariant derivatives acting on the same field do not produce new independent operators and for this reason such rearrangements can be ignored.