CheckMATE Tutorial

1 Introduction

This tutorial will guide you through a practical example to learn how CheckMATE works [1]. We have chosen the reference model [2] which you will consecutively analyse. In addition to CheckMATE, we will also run some other tools that are useful for collider phenomenology [3, 4].

2 Virtual Machine Setup

In order to follow the instructions of the tutorial, you have to use a virtual machine that we have already set up for you. The machine makes it easy to access a variety of different tools without requiring you to install them first. That way we can focus more on the actual physics. To install and run the virtual machine, you have to follow the intructions at

https://workshops.ift.uam-csic.es/MCtutorial/Tutorial

3 Testing the CheckMATE installation

After the virtual machine is set up, we do a simple test in order to check that CheckMATE works. You should find the CheckMATE executable in the directory

```
/home/checkmate/tools/checkmate/bin
```

In the same folder you should find an example parameter file testparam.dat with a minimally working example. If you look into the file, you will find the following

```
## General Options
[Mandatory Parameters]
Name: My_New_Run
Analyses: atlas_conf_2013_047

[Optional Parameters]

## Process Information (Each new process 'X' must start with [X])
[gluinogluino]
XSect: 3.53*FB
XSectErr: 1e-5*PB
Events: testfile.hep
```

Some general options and parameters are fixed in the text file which are necessary in order to run CheckMATE. Enter

```
./CheckMATE testparam.dat
```

Running the example with the event file testfile.hep provided by the Checkmate installation should, after confirmation and some intermediate output, produce the following result (which should not be surprising as the test events effectively do not contain any signal events):

```
[...]
Test: Calculation of r = signal/(95%CL limit on signal)
Result: Allowed
Result for r: r_max = 0.0
SR: atlas_conf_2013_047 - AL
```

The most important result is given by the `Result:` line, which will either state `Allowed` or `Excluded`. This statement alone allows the user to quickly perform simple tests on his model of interest.

4 Testing the reference BSM model

In the previous exercises, you should have learned how to implement the reference model, which is a simple extension of the SM, into FeynRules [5] and to obtain a corresponding model file for Madgraph. In the following, we will generate enough events with Madgraph in order to test the model point with CheckMATE. Run the Madgraph binary

```
cd /home/checkmate/tools/mg5_amc/bin
./mg5_aMC
```

First we have to load the model file

```
import model MC4BSM_2012_UFO
```

We want to instruct Madgraph to generate the following process

$$pp \to U\bar{U}, \tag{1}$$

where $U \to u\phi_1$ and $\bar{U} \to u\phi_2$ with $\phi_2 \to e^+e^-\phi_1$. First we define the two objects

```
define l = e+ e-
define lv = ev ev~
```

and then generate the process

```
generate p p > uv uv~, uv > u p1, (uv~ > u~ p2, (p2 > l lv, lv > l p1))
```

and finally create the output directory.

```
output /home/checkmate/tutorial/MC4BSM/mc4bsm
```

You can view a graph for our process in `HTML` directory. Within the output directory there is a designated folder that contains all files where user accessible parameters can be defined. It is located under

```
/home/checkmate/tutorial/MC4BSM/mc4bsm/Cards
```

We want to keep the default parameters of the model. Please check the model parameters in the file

```
param_card.dat
```

What are the masses of the heavy quark U, the heavy lepton E and both scalars Φ_1 and Φ_2? Change the widths of BSM particles to `Auto` so they will be properly calculated.

Before we generate events, we have to fix some run parameters for the event generation which are given in the

```
run_card.dat
```

We want to test the model point at the LHC with a beam energy of 8 TeV and thus changing the value in the corresponding card.

With the setup files ready, go back to the MadGraph terminal, type

```
launch /home/checkmate/tutorial/MC4BSM/mc4bsm
```

answer 1 afterwards to activate pythia=ON and then hit enter twice to start the event generation. It should take a couple of minutes to complete. You can simply close the firefox window that pops up automatically. As soon as MadGraph is done, run gunzip

```
/home/checkmate/tutorial/MC4BSM/mc4bsm/Events/run_01/tag_1_pythia_events.hep.gz
```

to unpack the compressed event file. Also, write down the cross section MadGraph quotes at the very end.

Since the final state includes 2 leptons, jets and missing transverse energy, we want to test the model point with the LHC search `atlas_conf_2013_089` (2 leptons + jets + MET) [6].

We make a copy of the testparam.dat card and name it mc4bsm.dat in the binary directory of CheckMATE. We modify the relevant entries in the CheckMATE card.

- change the name of the CheckMATE run
- set the correct ATLAS analysis
- change the cross section and assume a 10% error on the cross section
- set the right link to the hep file

Run CheckMATE again and then will see if the model point is excluded or not. What other production processes can contribute to this final state?

References

[1] M. Drees, H. Dreiner, D. Schmeier, J. Tattersall and J. S. Kim, Comput. Phys. Commun. **187** (2014) 227 [arXiv:1312.2591 [hep-ph]].

[2] S. Ask, N. D. Christensen, C. Duhr, C. Grojean, S. Hoeche, K. Matchev, O. Mattelaer and S. Mrenna *et al.*, arXiv:1209.0297 [hep-ph].

[3] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao and T. Stelzer *et al.*, JHEP **1407** (2014) 079 [arXiv:1405.0301 [hep-ph]].

[4] T. Sjostrand, S. Mrenna and P. Z. Skands, JHEP **0605** (2006) 026 [hep-ph/0603175].

[5] A. Alloul, N. D. Christensen, C. Degrande, C. Duhr and B. Fuks, Comput. Phys. Commun. **185** (2014) 2250 [arXiv:1310.1921 [hep-ph]].

[6] Search for strongly produced supersymmetric particles in decays with two leptons at 8 TeV https://cds.cern.ch/record/1595272?ln=es