# 1 System Requirements

We have built MASCOT on top of SCOTS [1]. Therefore, the system requirements and the installation procedure for our software are the same as for SCOTS. As for SCOTS, you need a working C++ developer environment, the CUDD library [2], and a working installation of a recent version of MATLAB (the latter is only used for the visualization of some results). Following are the requirements as detailed in the manual of SCOTS:

1. A working C++ developer environment:
   - Mac OS: Install the latest version of Xcode.app including the command line tools.
   - Linux: Most Linux OS distributions already include all the necessary tools already. We used the compiler GCC 4.9.x.
   - Windows: SCOTS has not been tested on a Windows machine.
2. A working installation of the CUDD library with
   - the C++ object oriented wrapper,
   - the dddmp library, and
   - the shared library

   option enabled. The package follows the usual configure, make and make install installation routine. We use cudd-3.0.0, with the configuration

   ```
   $ ./configure --enable-shared --enable-obj --enable-dddmp --prefix=/opt/local/
   ```

   The BDD installation can be tested by compiling a dummy program, e.g. test.cc

   ```
   #include<iostream>
   #include "cuddObj.hh"
   #include "dddmp.h"
   int main () {
   Cudd mgr(0,0);
   BDD x = mgr.bddVar();
   }
   ```

   by running

   ```
   $ g++ test.cc -I/opt/local/include -L/opt/local/lib -lcudd
   ```

   It has been reported that on some linux machines, the header files `util.h` and `config.h` were missing in `/opt/local`, and the fix is to manually copy the files to `/opt/local/include` by running

   ```
   $ sudo cp ./util/util.h /opt/local/include/
   $ sudo cp ./config.h /opt/local/include/
   ```

3. A working installation of a recent version of MATLAB. We conducted our experiments with version R2016a (9.0.0.341360) 64-bit. To compile the mex files:
   (a) open MATLAB and setup the mex compiler by

   ```
   >> mex -setup C++
   ```

   (b) Open a terminal and go to `./mfiles/mexfiles/` (within the MASCOT directory, see below). Edit the `Makefile` and adjust the variables

   `MATLABROOT and CUDDPATH`

   and run

   ```
   $ make
   ```

   (The MATLAB root directory is returned when typing `matlabroot` in the command window of MATLAB.)

## 2 Directory Structure

After you unzip MASCOT, you will see the following directory structure:

- `./bdd/` Contains the C++ source code for the SCOTS and MASCOT classes which use Binary Decision Diagrams as the underlying data structure
- `./doxygen/` C++ Class documentation directory
- `./manual/` Both SCOTS and MASCOT manuals
- `./mascot_examples/` Examples which were used in the papers [**?**, **?**] demonstrating the usage and performance of MASCOT todo: add the references
- `./mfiles` Contains an mfile as a wrapper to the mex-file functions
- `./mfiles/mexfiles/` mex-file to read the C++ output from file
- `./scots_examples/` Some C++/Matlab programs demonstrating the usage of basic SCOTS
- `./test/` Some C++/Matlab test programs
- `./utils/` Some C++ header files used by the source codes in `./bdd/`

## 3 Getting Started

For validating any of the tables/figures presented in the paper

1. Go to a sub-directory in one of the example folders.
2. Edit the `Makefile`:
   (a) adjust the compiler, and
   (b) adjust the directories of the CUDD library.
3. Compile and run the executable, for example in `./mascot_examples/safety/dcdc/LazySafe/dcdc3A/` run

```
$ make
$ ./dcdc
```

4. Use the .m files to plot results. todo: discuss different modes

## References


[1] M. Rungger and M. Zamani. SCOTS: A tool for the synthesis of symbolic controllers. In *HSCC*, pages 99–104. ACM, 2016.

[2] Fabio Somenzi. Cudd: Cu decision diagram package-release 2.4. 0. *University of Colorado at Boulder*, 2009.