This file describes the usage of the tool eVolCheck to check the upgrades of software. The provided distribution contains the following files:

- eVolCheck (Model Checker with upgrade checking capabilities)
- goto-cc (preprocessor for model extraction)
- examples (directory containing the sources of programs and corresponding precompiled models called goto-binaries)
- run_demo.sh (small running examples, representing different types of changes, with comments)

The following steps show the example of usage of the tools:

1. Remove the subsidiary files, produced during previous verification runs, i.e., __summaries and __omega:

```
~/evolcheck$ rm __summaries __omega
```

3. Create a model of the program by running the preprocessor goto-cc. Choose one of the "*_orig.c" files in "examples" directory, for example,

```
~/evolcheck$ ./goto-cc
    examples/valid/change_valid_orig.c -o examples/valid/change_valid_orig.out
```

("examples/valid/change_valid_orig.c" - the input *.c file, "-o" - flag pointing to the output file, "examples/valid/change_valid_orig.out" - the resulting model, represented by goto-binary)

4. Run the eVolCheck for the initial check of the program. For example,

```
~/evolcheck$ ./evolcheck --init-upgrade-check --unwind 10
                                              examples/valid/change_valid_orig.out
```

("--init-upgrade-check" - a parameter required for the later upgrade checking, "--unwind <N>" is optional (number of unrollings for every loop), the user can change the value of N)

5. Check the eVolCheck outputs. The following message at the end of the eVolCheck output indicates that ...

   a) ... verification was successful:

```
"ASSERTION(S) HOLD(S) AFTER INLINING."
```

In this case it is possible to run verification of an upgrade.

   b) ... verification failed:

```
"ASSERTION(S) DO(ES)N'T HOLD AFTER INLINING.
 A real bug found."
```

In this case it is required to fix the bug first, and go to the step 2.

6. For upgrade checking, create the model of the upgraded program using goto-c. Run it for the file "*_upgr.c" of an upgrade, corresponding to the "*_orig.c" file chosen at the step 3:

```
~/evolcheck$ ./goto-cc
```

```
examples/valid/change_valid_upgr.c -o examples/valid/change_valid_upgr.out
```

7. Run the eVolCheck to check the upgraded program. For example,

```
~evolcheck$ ./evolcheck --do-upgrade-check
                        examples/valid/change_valid_upgr.out --unwind 10 examples/
                        valid/change_valid_orig.out
```

("--do-upgrade-check <file>" points to the upgraded goto-binary, "--unwind <N>" must be the same as for the original check)

8. Check the eVolCheck outputs. There are several possible cases:

   a) two programs are identical, and therefore correct (examples for this case are located at "examples/ident"). The output should end by the message:

   "The programs are trivially identical."

   b) the upgraded program is correct (examples for this case are located at "examples/valid"). The output contains the following message for every function summary:

   "summary was verified"

   c) the upgraded program is incorrect (examples for this case are located at "examples/not_valid"). The output contains the following message for the summary of the function main:

   "Old summary is no more valid.
   ...
   summary cannot be renewed. A real bug found."

9. Additional information about the usage of the tool can be found by typing

```
~/evolcheck$ ./evolcheck --help
```

For further assistance please contact grigory.fedyukovich@usi.ch.