

Formale Grundlagen der Informatik III: Softwaretutorial



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Wintersemester 2014/2015

Prof. Dr. Stefan Katzenbeisser, Christian Schlehuber

Aufgabe 1 SPIN einrichten

In diesem Abschnitt wird die Installation von SPIN auf dem lokalen Rechner erläutert. Am Ende des Abschnitts erhalten Sie außerdem die Links zu einem Web-Interface für SPIN sowie ein Plugin für Eclipse. (*Hinweis: Wir testen Beispiele und Labs mit dem Kommandozeilen-System; für das Plugin und das Web-Interface übernehmen wir keinen Support!*)

- Lesen Sie sich die Anleitung unter <http://spinroot.com/spin/Man/README.html> durch.
- Laden Sie sich die Software für Ihr Betriebssystem herunter und installieren Sie diese (siehe Punkt 2 der Anleitung),
- Sollten Sie bisher kein GCC auf Ihrem System installiert haben, so müssen Sie dies nun installieren. Unter Windows kann dies durch MinGW erfolgen
<https://sourceforge.net/projects/mingw/files/Installer/>
Nach dem Download des Installers kann man in dem Install Manager das **base**, **gcc-g++** und **gcc-objc** Paket auswählen und installieren. Nach der Installation muss noch das `/bin` Verzeichnis den Windows Path-Variablen hinzugefügt werden. Bitte informieren Sie sich selbständig darüber, wie dies für Ihre Distribution durchgeführt wird.
- Testen Sie abschließend die Funktionsfähigkeit Ihres Systems mit dem folgenden kleinen Test Model. Erstellen Sie eine Datei `hello.pr` mit folgendem Inhalt:

```
/* A "Hello World" Promela model for SPIN. */
active proctype Hello() {
    printf("Hello process, my pid is: %d\n", _pid);
}
init {
    int lastpid;
    printf("init process, my pid is: %d\n", _pid);
    lastpid = run Hello();
    printf("last pid was: %d\n", lastpid);
}
```

Führen Sie anschließend den folgenden Befehl aus (Windows-Beispiel):

```
C:\Users\xyz\pc_spin642>spin_64bits.exe -n2 models/hello.pr
```

Anschließend sollten Sie das folgende Ergebnis in der Kommandozeile sehen können:

```
C:\Users\xyz\pc_spin642>spin_64bits.exe -n2 models/hello.pr
init process, my pid is: 1
last pid was: 2
Hello process, my pid is: 0
Hello process, my pid is: 2
3 processes created
```

Alternativ können Sie auch das EpiSpin Plugin der TU Delft für Eclipse verwenden. Eine Anleitung zur Installation finden Sie unter: <http://epispin.ewi.tudelft.nl/>

Außerdem ist ein Web-Interface für SPIN verfügbar: <http://abu.se.informatik.tu-darmstadt.de/sefm/spin/>

Aufgabe 2 CBMC einrichten

Für die Installation von CBMC gehen Sie bitte auf die folgende Website:

<http://www.cprover.org/cbmc/>

Folgen Sie dort den Anweisungen für Ihr Betriebssystem unter dem Punkt *CBMC Download*.

Anschließend kann die Funktionsfähigkeit durch die folgenden zwei Beispiele getestet werden. Legen Sie hierfür eine Datei *test.c* mit folgendem Inhalt an:

```
int puts(const char *s) { }

int main(int argc, char **argv) {
    int i;
    if(argc>=1)
        puts(argv[2]);
}
```

Und eine Datei *test2.c* mit dem folgenden Code:

```
int array[10];

int sum() {
    unsigned i, sum;

    sum=0;
    for(i=0; i<10; i++)
        sum+=array[i];
}
```

Anschließend kann man folgenden Befehl ausführen:

```
cbmc test.c --show-properties --bounds-check --pointer-check
```

Nun sollte man von CBMC Hinweise auf mögliche Null-Pointer innerhalb des Codes der *test.c* erhalten.

Die Verifikation kann man noch mit folgendem Befehl testen:

```
cbmc test2.c --function sum
```

Hierbei sollte man ein abschließendes **VERIFICATION SUCCESSFUL** erhalten.

Bitte beachten Sie, dass unter Windows die Befehle aus der Visual Studio Kommandozeile ausgeführt werden müssen.