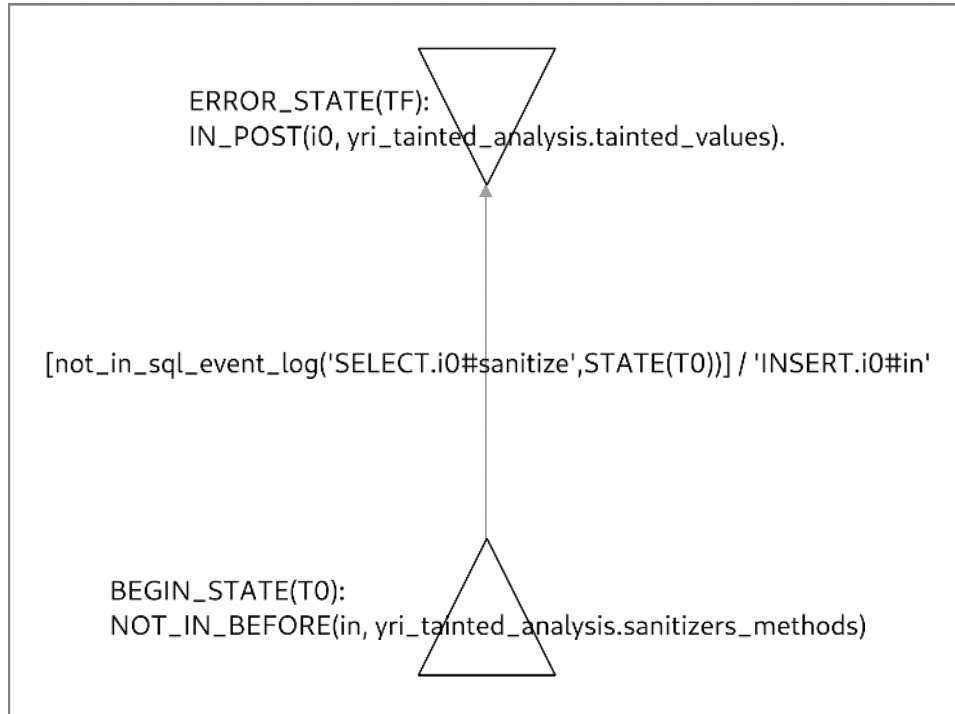


We want to have a testing for the implementation of taint analysis because it is quite complicated to have it as a modular unit within YRI–DB–RUNTIME–VERIF.

Therefore I need to test and implement it thoroughly since having it working would empower users of YRI–DB–RUNTIME–VERIF to specify parametric value in their state diagram mealy machine (SDMM).

Figure 1: A taint analysis "SDMM" to test parametric SDMM.



Test–1

- 1.) Compilation of YRI–DB–RUNTIME–VERIF and automatic creation of a '.deb' installation package is done with following command :

```
./yri_db_runtime_verif_BUILD_DEBIAN_PACKAGE.sh
```

- 2.) After implementation of taint analysis functionality, the current running instance of YRI–DB–RUNTIME–VERIF needs to be stopped & restarted with a new binary.

```
yri_DB_RUNTIME_VERIF_uninstall.sh && yri_DB_RUNTIME_VERIF_INSTALL.SH && sudo systemctl stop yri-db-runtime-verif
```

- 3.) then start a newer instance directly without calling it as a system daemon.

```
/opt/yri-db-runtime-verif/bin/yri-db-runtime-verif
```

- 4.) Then start 'YERITH–ERP–9.0'.

- 5.) Particularly we need to check for instance that any string input from a "YeritLineEdit" GUI widget is coined as a tainted input except if it was also passed into a sanitizer method;

To test this, Tainted–Analysis functionality, we created a SDMM using YRI_QVGE : Figure 1 illustrates this tainted analysis as a SDMM.