

Apron and Taylor1Plus

COSYNUS, Laboratoire d'informatique de l'Ecole Polytechnique, CNRS

July 11, 2019

Abstract

This document is a detailed explanation about: how to install and use Taylor1Plus.

1 Apron

The first and the most significant step is to have Apron (library of numerical abstract domains for static analysis by abstract interpretation) and all it's prerequisites installed. The prerequisites for C API are

- GCC
- GMP 5.x.y and MPFR 3.x.y or up

If someone wants to use the OCaml bindings, the prerequisites for OCaml API are

- MLGMPIDL (<http://www.inrialpes.fr/pop-art/people/bjeannet/mlxxxidl-forge/mlgmpidl>, SVN: [svn://scm.gforge.inria.fr/svn/mlxxxidl/mlgmpidl/](http://scm.gforge.inria.fr/svn/mlxxxidl/mlgmpidl/))
- OCaml 3.11 or up (<http://www.caml.org>)
- CamlIDL 1.05 (<http://caml.inria.fr/camlidl>)
- perl

One can install Apron through OPAM package manager or via the source files. In order to use OPAM, the command is **opam install apron**. The prerequisites can also be installed using OPAM. However, it is highly recommended to install Apron using the source files if one is intending to rework with the abstract domains. Kindly, pursue the following steps for installing Apron through the source.

- Depending on the way one installs OCaml, one might end up missing OCaml native-code compiler (OCAMLOPT).
- Please be sure to install **ocaml-native-compilers**
 - For Ubuntu, sudo apt-get install ocaml-native-compilers should do
 - Otherwise, you can find information in this link: <https://packages.ubuntu.com/xenial/ocaml-native-compilers>

- Once all the prerequisites are installed, one can download Apron through this SVN link: `svn co svn://scm.gforge.inria.fr/svnroot/apron/apron/trunk apron`
- The distribution through the above SVN link is the last committed version.
- Once Apron has been downloaded, go inside the Apron directory
- Run the command `./configure` followed by **make**, **sudo make install**
- This should install the Apron library

2 Taylor1plus

Taylor1+ is an efficient implementation of the affine forms-based numerical abstract domain or zonotopes in Apron [1]. The affine forms-based numerical abstract domain was introduced by Eric Goubault and Sylvie Putot in [2]. Taylor1+ defines the meet operation of two affine sets as a logical product of standard zonotopes and boxes [3]. It implements several join operations over such logical products. It also implements test for inclusion check, test for intersection check and splitting operation for zonotopes [4]. The domain requires the APRON library wrappers to work properly.

Inside the directory of Taylor1plus there are few test files like “t1p_test_eval_texp.c”. If someone wants to test the file or use it for his/her own purpose, first one should uncomment the line in the makefile of Taylor1plus saying:

```
test%: t1p_test_eval_texp%.o
$(CC) $(CFLAGS) -DNUM_DOUBLE -o $$ $(LIBS) -L. $< -lt1p$* -lbox$*
-lpolkaMPQ_debug -lapron t1p_internal$*.o t1p_representation$*.o
```

Now, compiling by “make” may produce an error claiming “undefined reference to box_manager_alloc”. This can be taken care of by re-ordering the above syntax as:

```
test%: t1p_test_eval_texp%.o
$(CC) $(CFLAGS) -DNUM_DOUBLE -o $$ t1p_internal$*.o
t1p_representation$*.o $(LIBS) -L. $< -lt1p$* -lbox$* -lpolkaMPQ_debug -lapron
```

References

- [1] Ghorbal, K., Goubault, E., & Putot, S. (2009, June). The zonotope abstract domain taylor1+. In International Conference on Computer Aided Verification (pp. 627-633). Springer, Berlin, Heidelberg.
- [2] Goubault, E., & Putot, S. (2006, August). Static analysis of numerical algorithms. In International Static Analysis Symposium (pp. 18-34). Springer, Berlin, Heidelberg.
- [3] Ghorbal, K., Goubault, E., & Putot, S. (2010, July). A logical product approach to zonotope intersection. In International Conference on Computer Aided Verification (pp. 212-226). Springer, Berlin, Heidelberg.

- [4] https://swim2016.sciencesconf.org/conference/swim2016/pages/Kabi_Goubault_Putot.pdf