

CSCI-GA.3140-001 – Abstract Interpretation, Spring 2019

Optional Project

Name: Yusen Su

NetID: ys3547

University ID (UID): N15412725

Work Log

1. Project plan
 - Read the requirement and remarked the implementation hard point
 - Started writing abstract interpreter by following §19.2
 - Fixing the code error and debugging
 - Testing interpreter for abstract domain of parity
 - Defined abstract domain of constancy
 - Defined abstract domain of congruence
 - Defined abstract domain of interval
 - Reading the user manual of Apron
 - Defined abstract domain of octagon
 - Wrote the document
2. Project implementation
 - Using OCaml and referenced by example files
 - Addressed implementation hard core
 - i. How to implement abstract interpreter
 - ii. How to calculate fixpoint and define F for loop
 - iii. How to define operators for abstract domain
 - iv. How to use Apron

Testing

1. Test performance
 - Static Analysis is correct for Parity, Constancy and Interval.
 - There are some test cases failed for Congruence domain.
 - Octagon analysis would be correct (no referenced examples).
2. Test limitation
 - Only allow variable x and y
 - For each abstract domain, I only use the example from the makefile. It could extend to multiple variables.
 - No functor for abstract domain
 - No nand operator implementation
 - Octagon limitation

- The Octagon does not have function transferring abstract value to string (as we print needed). The output is not formatted.
- The Octagon does not support != operator for APRON tree constraints. It will show this error when analyze != code fragment:
 - Fatal error: exception Failure("!= not yet supported")

Code

Please find src folder for more details.

Compiling & Usage

make analyseParity: parity analysis (for variables x,y only)

make analyseConst: consistency analysis (for variables x,y only)

make analyseCong: congruence analysis (for variables x,y only)

make analyseInt: interval analysis (for variables x,y only)

make analyseOct: octagon analysis (for variables x,y only)

* Bottom and Top in octagon output is the same meaning as \perp and T.