3. A General Static Analysis Framework Based on a Compositional Semantics

20190065 김기환 (kimkihwan@kaist.ac.kr)

* Input-Output Semantic = Input set → Output set (Non-deterministic)
* Program State = Memory State + Control State
* Memory State = Describe all assigned value of variables.
* Semantics of scalar =

* Semantics of Boolean =

* Filtering function =
* Semantics of command =

*note. semantic of while statement can be replaced by least fixpoint (Kleene’s Thm.)*

* Concrete domain = = domain of all concrete behavior
* Abstract domain = = domain of all abstraction
* Abstraction relation =

= ‘’ means abstraction ‘a’ describe concrete behavior ‘c’

* Concretization =
* Abstraction =
* Galois connection = pair of concrete and abstraction

1. and are monotone function.

2. loses precision

3. gives more information

* Non-relational abstraction = abstraction with no info. about variable’s relationship

*ex. sign, interval, etc.*

* Definition of Non-relational abstraction :=

each abstraction only describes about single variable at once

order relation follows order over value

concretization is union of concrete values

*note. is memory state and is abstraction element*

least/greatest element = respectively

get set of memory state, return abstraction

* Relational abstraction = abstraction that maintains relationship between some variables

*ex. linear equality, convex polyhedron, etc.*

* = analysis which returns post-condition of program p with pre-condition a

ex.

*note. It is also called ‘abstract semantics*

* if and then => soundness of
* Theorem. Approximation of composition =

Let are monotone. Let are approx. of

Then is is approx. of

*note. this theorem guarantees composition of approx. of subexpression is available*

* Abstract interpretation of expression =

where

(It means, operator returns abstraction of result of with all concrete value given)

*note. is an abstraction function but maybe not a best-approximation.*

* Analysis of assignment =

* Algorithm for analyzing assignments with relational abstraction =

1. Introduce new variable x’ and assign E to x’.

2. Represent x’=E precisely as good as possible.

3. Replace all x to expression with x’. Rename all x’ to x.

* Abstract filtering function = where
* Abstract filtering of Boolean expression (in interval abstract domain) =

where

* Abstract join = where
* Analysis of condition = union of two branches (true and false) =
* Sequence of concrete =

Let where

Sequence satisfy and for given

* Sequence of abstraction =

Let where

Sequence satisfy and for given

*note. holds for all*

* Abstract iterates for analysis of loops =

Union all iterates until it become stabilize (it means for some )

* How to guarantee abstract iterates converge =

1. Finite Height Lattice =

Since abstract domain has top and bottom, it is clear that domain is lattice.

If it has finite height, there exists stabilized term at point less than height.

2. Widening = where

Widening remove not stabilized abstract element.

Requires some skills to make abstraction more precisely.

* Analysis of loop =

*note. above formula can be expressed by least fixpoint of*

* is not monotone. It means sometimes more precise pre-condition doesn’t give more precise post-condition.
* Framework of abstract interpretation =

1. Concrete semantic = define and fix concrete semantics

2. Abstraction = select abstract domain

3. Abstract semantic = derive analysis algorithm