

Software Verification

Bounded Box

Mattia Bottaro - Mauro Carlin
May, 2018



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

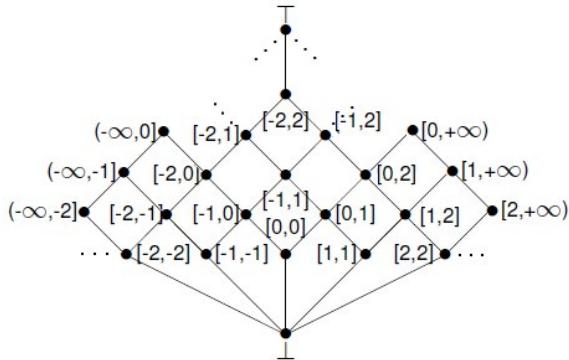
- 1 Project
- 2 Our Contribution
- 3 Example
- 4 Improvements that could be made

We have chosen the **Bounded Box Domain**, which is a parametric restriction of the interval abstract domain *Int*:

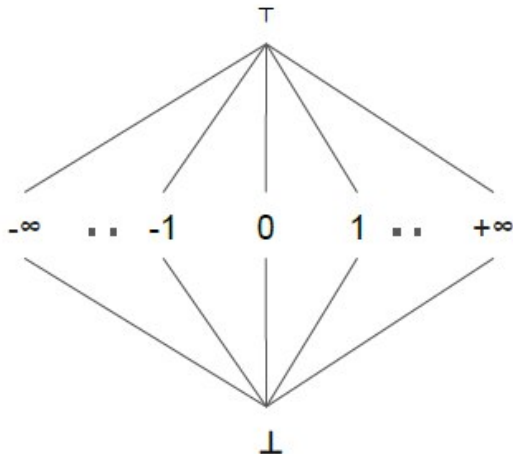
Given $m, n \in \mathbb{Z} \cup \{-\infty, +\infty\}$, then

$$\text{Int}_{m,n} := \{\emptyset, \mathbb{Z}\} \cup \{[k, k] \mid k \in \mathbb{Z}\} \cup \{[a, b] \mid a < b, [a, b] \subseteq [m, n]\} \cup \{(-\infty, k] \mid k \in [m, n]\} \cup \{[k, +\infty) \mid k \in [m, n]\}$$

Bounded Box with $m = -2$, $n = 2$

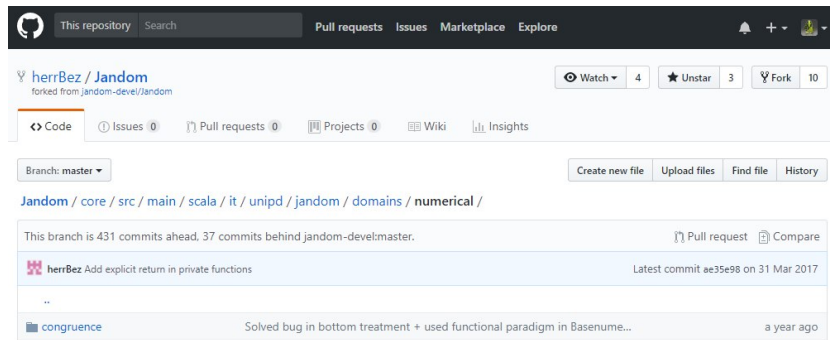


Bounded Box with $m > n$



- Static Analyzer for **Numerical** and Object Domains (forse togliere object)
- Jandom was created at the University of Chieti-Pescara
- It's a buildup of **RANDOM**, which analyzes **R** code
- Jandom is written in **Scala**
- Jandom analyzes JVM bytecode using **Soot**

We've extended Jandom from this repository created by University of Padua' students



The screenshot shows the GitHub interface for the repository **herrBez / Jandom**, which is a fork of **jandom-devel/Jandom**. The repository has 4 watchers, 3 unstars, and 10 forks. The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The repository's main branch is **master**. The commit history shows two recent commits: one by **herrBez** titled "Add explicit return in private functions" (latest commit `ae35e98` on 31 Mar 2017) and another by **congruence** titled "Solved bug in bottom treatment + used functional paradigm in Basenum..." (committed a year ago).

We have:

- 1** Implemented the Integer Interval Domain
- 2** Implemented Bounded Box Domain specializing the previous domain, because abstract operators of both domains are very similar

Abstract **sum** operator algorithm in Bounded Box Domain.

- 1 Execute sum operator of Interval Domain.

$$[a, b] +_b^{\#} [c, d] = [a + c, b + d] = [e, f]$$

- 2 $[e, f]$ must be represented as an element of Bounded Box Domain

$$[e, f] = \begin{cases} \top^{\#} & e < m \wedge f > n \\ [n, +\infty) & e \geq n \wedge e \neq f \\ [e, +\infty) & e < n \wedge f > n \\ (-\infty, m] & f \leq m \wedge e \neq f \\ (-\infty, f] & f > m \wedge e < m \\ [e, f] & \text{otherwise} \end{cases}$$

Abstract **reminder** operator algorithm in Box Domain

$$[a, b] \%_b^\# [c, d] = \begin{cases} \top^\# & [c, d] = \top^\# \\ \perp^\# & [a, b] = \perp^\# \vee [c, d] = \perp^\# \vee [c, d] = [0, 0] \\ [0, 0] & [a, b] = [0, 0] \\ [0, d - 1] & c \geq 0 \\ [c + 1, 0] & d \leq 0 \\ [c + 1, d - 1] & \textit{otherwise} \end{cases}$$

We have defined a new type, called *Inflnt*, to:

- 1 model infinity values with Integer type
- 2 overload operations between integer number
- 3 simplify further contribution

Example

$$(+\infty) + n = +\infty$$

$$(+\infty) \times (-\infty) = -\infty$$

$$(+\infty) \div (+\infty) = 0$$

```
int i0, i1, i2, i3;

/*[ [ i0 = T , i1 = T , i2 = T , i3 = T ]types: int,int,int,int ]*/
    i0 = -5;

/*[ [ i0 = [-5,-5] , i1 = T , i2 = T , i3 = T ]types: int,int,int,int ]*/
    i1 = 15;

label1:
/*[ [ i0 = [-5,6] , i1 = T , i2 = T , i3 = T ]types: int,int,int,int ]*/
    if i0 > 5 goto label2;

/*[ [ i0 = [-5,5] , i1 = T , i2 = T , i3 = T ]types: int,int,int,int ]*/
    i1 = i0 * i1;

/*[ [ i0 = [-5,5] , i1 = T , i2 = T , i3 = T ]types: int,int,int,int ]*/
    i0 = i0 + 1;

/*[ [ i0 = [-4,6] , i1 = T , i2 = T , i3 = T ]types: int,int,int,int ]*/
    goto label1;

label2:
/*[ [ i0 = [5,6] , i1 = T , i2 = T , i3 = T ]types: int,int,int,int ]*/
    if i0 < 7 goto label3;

/*[ [ empty ]types: int,int,int,int ]*/
    i2 = i0 - 10;

/*[ [ empty ]types: int,int,int,int ]*/
    goto label4;

label3:
/*[ [ i0 = [5,6] , i1 = T , i2 = T , i3 = T ]types: int,int,int,int ]*/
    i3 = i0 + 10;

label4:
/*[ [ i0 = [5,6] , i1 = T , i2 = T , i3 = [10,+] ]types: int,int,int,int */
```

