

Towards Statically Reasoning About R Vectors

$$\mathcal{D}_{\mathcal{V}_{\mathcal{B}}^{\#}} \stackrel{\text{def}}{=} \left\langle \mathcal{V}_{\mathcal{B}}^{\#}, \sqsubseteq_{\mathcal{V}_{\mathcal{B}}^{\#}}, \sqcup_{\mathcal{V}_{\mathcal{B}}^{\#}}, \sqcap_{\mathcal{V}_{\mathcal{B}}^{\#}}, \perp_{\mathcal{V}_{\mathcal{B}}^{\#}}, \top_{\mathcal{V}_{\mathcal{B}}^{\#}} \right\rangle$$



**UNIVERSITÀ
DI PARMA**

Candidate: Manuel Di Agostino

2026-07-09

Advisor: Prof. Vincenzo Arceri

Academic Year 2025/2026

1. Motivation

Dominant in statistical computing.

- Over **23,938** community packages (CRAN 2026)
- **No type** declarations. Every variable can hold any value at any time
- **Interpreted**, dynamically typed: errors surface only at runtime

```
1 gdef <- function() {  
2   x <<- "hello!"  
3 }  
4  
5 x <- 3  
6 gdef()  
7 x + 1 # Runtime error
```

r

Everything is a vector:

- **TRUE**, *logical* of length 1
- **3** or **3.14**, *double* of length 1
- **c(1L, 2L, 3L)**, *integer* of length 3

Everything is a vector:

- **TRUE**, *logical* of length 1
- **3** or **3.14**, *double* of length 1
- **c(1L, 2L, 3L)**, *integer* of length 3

Notation

In the following, we denote vectors as $[e_1, e_2, \dots, e_n]$.

Everything is a vector:

- `TRUE`, *logical* of length 1
- `3` or `3.14`, *double* of length 1
- `c(1L, 2L, 3L)`, *integer* of length 3

Notation

In the following, we denote vectors as $[e_1, e_2, \dots, e_n]$.

So what about...

- `c()`
- `"hello"`
- `c(TRUE, 3L, 2.5)`

- `c()` is equivalent to the reserved keyword **NULL**

```
is.null(c()) # TRUE  
length(c()) # 0
```

r

- `c()` is equivalent to the reserved keyword **NULL**

r

```
is.null(c()) # TRUE  
length(c()) # 0
```

- `"hello"` is a *character* vector of length 1

r

```
length("hello") # 1
```

- `c()` is equivalent to the reserved keyword **NULL**

r

```
is.null(c()) # TRUE  
length(c()) # 0
```

- `"hello"` is a *character* vector of length 1

r

```
length("hello") # 1
```

- `c(TRUE, 3L, 2.5)` implicitly converted to a *double* vector

r

```
typeof(c(TRUE, 3L, 2.5)) # "double"
```

Remark

Type Hierarchy

Vectors elements must be homogeneous in their type:

logical $\prec$ integer $\prec$ double $\prec$ character

- Implicit **coercion** occurs:

r

```
c(TRUE, "hello")      # → ["TRUE", "hello"]
```

- Special logical value: NA, representing missing data.

r

```
c(NA, "hello")        # → [NA, "hello"]
```

Recycling

- R operations are **element-wise** across vectors.
- When two vectors have different lengths, the shorter is **recycled** to match the longer.

Example

```
c(1, 2, 3) + c(10, 20)  
# [10, 20] recycled to [10, 20, 10]
```

r

The resulting vector is [11, 22, 13].

Attributes

- Any R vector can carry extra metadata via *attributes*
- Most used are `dim`, `class` and `names`

Example

From vector to matrix
r

```
x <- 1:10
dim(x) <- c(2, 5)
#      [,1] [,2] [,3] [,4] [,5]
# [1,]    1    3    5    7    9
# [2,]    2    4    6    8   10
```

Current static analysis for R are not enough.

- flowR (Sihler and Tichy 2025):
dataflow analysis, no sound vector
abstraction
- No tool models coercion, recycling,
or attributes



Sound by construction (Cousot and Cousot 1977)

- Given the set of states $\Sigma : \text{Var} \rightarrow \text{Val}$, a *program property* is an element of $\mathcal{P}(\Sigma)$

$$\Phi = \{\sigma \in \Sigma \mid \forall v \in \text{Var} : \sigma(v) \neq 0\}$$

- Concrete **Collecting Semantics** computes reachable set of states at program points
- Abstract Interpretation over-approximates via an **Abstract Domain**

2. The μ R language

2.1 Concrete Syntax

μR is a subset of R focused on vector operations, omitting environments, promises, and lazy evaluation.

P	$::=$	S		E	$::=$	$Atm \mid \triangleright E$
S	$::=$	E				$e_1 \bowtie_{arith} e_2$
		$\text{if } (E) \{S_1\} \text{ else } \{S_2\}$				$e_1 \bowtie_{rel} e_2$
		$\text{while } (E) \{S\}$				$c()$
		$S_1 S_2$				$c(E_1, \dots, E_n)$
		$\{S\}$				$x \leftarrow e$
$\bowtie_{arith}$	$::=$	$+ \mid - \mid * \mid /$				$x[E_{idx}]$
$\bowtie_{rel}$	$::=$	$> \mid < \mid = \mid \neq \mid \leq \mid \geq$				$x[E_{idx}] \leftarrow e$
$\triangleright$	$::=$	$! \mid -$				$\text{setAttr}(x, \{s_1, \dots, s_n\})$
						$\text{getAttr}(x)$

Table 1: Syntax of μR (subset of the full grammar).

$$Atm ::= Var \mid Float \mid Int \mid Bool \mid \text{NULL}$$

Definition

Concrete Vector

Given the set of atomic values $\mathbb{V}_{Atm}$ and a set of string attribute names $\mathcal{A}$, a concrete vector $v \in \mathcal{V}$ is a pair $(\mathbb{N} \rightarrow \mathbb{V}_{Atm}) \times \mathcal{A}$: a total map from indexes to atomic values, plus attribute names.

2.2 Concrete Denotational Semantics

$\mathbb{E}[\mathbf{e}] : \mathcal{E} \rightarrow (\mathcal{E} \times (\mathcal{V} \cup \mathcal{P}(\text{String})))$, where $\mathbf{x} \in \text{Var}$, $\mathbf{e} \in E$

$$\begin{aligned} \mathbb{E}[\mathbf{e}_1 \diamond \mathbf{e}_2] \rho &\stackrel{\text{def}}{=} (\rho_2, v'_1 \diamond v'_2) \\ &\text{where } (\rho_1, v_1) = \mathbb{E}[\mathbf{e}_1] \rho, \\ &\quad (\rho_2, v_2) = \mathbb{E}[\mathbf{e}_2] \rho_1, \\ &\quad (v'_1, v'_2) = \text{Recycle}((v_1, v_2)), \\ &\quad \diamond \in \{\bowtie_{arith}, \bowtie_{rel}\} \end{aligned}$$

$$\mathbb{E}[\mathbf{c}(\)] \rho \stackrel{\text{def}}{=} (\rho, [])$$

$$\begin{aligned} \mathbb{E}[\mathbf{c}(\mathbf{E}_1, \dots, \mathbf{E}_n)] \rho_0 &\stackrel{\text{def}}{=} (\rho_n, v_1 \oplus^\# \dots \oplus^\# v_n) \\ &\text{where } \forall i \in [1, n]. (\rho_i, v_i) = \mathbb{E}[\mathbf{e}_i] \rho_{i-1} \end{aligned}$$

$$\begin{aligned} \mathbb{E}[\mathbf{x}[\mathbf{e}_{idx}]] \rho &\stackrel{\text{def}}{=} (\rho', \text{Select}(\rho'(\mathbf{x}), v_{idx})) \\ &\text{where } (\rho', v_{idx}) = \mathbb{E}[\mathbf{e}_{idx}] \rho \end{aligned}$$

Table 2: Selected rules of the concrete semantics (incomplete list).

3. The Abstract Domain of R Vectors

Definition

Raw Abstract Element

A *raw abstract vector* in $\tilde{\mathcal{V}}_{\mathcal{B}}^{\#}$ is a 4-tuple

$$\nu = ([l, u], \langle p_1 \cdots p_k \rangle, s, a) \in \text{ltv}_0 \times \mathcal{B}^{\#*} \times \mathcal{B}^{\#} \times \text{Attr}^{\#}$$

with the invariant:

$$l \leq k \leq u \wedge u \neq +\infty \Rightarrow s = \perp_{\mathcal{B}^{\#}} \wedge u = k$$

Definition

Abstract Domain of R Vectors

The abstract domain is the complete lattice:

$$\mathcal{D}_{\mathcal{V}_{\mathcal{B}}^{\#}} \stackrel{\text{def}}{=} \left\langle \mathcal{V}_{\mathcal{B}}^{\#}, \sqsubseteq_{\mathcal{V}_{\mathcal{B}}^{\#}}, \sqcup_{\mathcal{V}_{\mathcal{B}}^{\#}}, \sqcap_{\mathcal{V}_{\mathcal{B}}^{\#}}, \perp_{\mathcal{V}_{\mathcal{B}}^{\#}}, \top_{\mathcal{V}_{\mathcal{B}}^{\#}} \right\rangle$$

where $\mathcal{V}_{\mathcal{B}}^{\#} = \left\{ [\nu]_{\equiv} \mid \nu \in \tilde{\mathcal{V}}_{\mathcal{B}}^{\#} \right\}$ where:

- $\perp_{\mathcal{V}_{\mathcal{B}}^{\#}} = \left(\perp_{\text{Itv}_0}, \varepsilon, \perp_{\mathcal{B}^{\#}}, \perp_{\text{Attr}^{\#}} \right)$
- $\top_{\mathcal{V}_{\mathcal{B}}^{\#}} = \left(\top_{\text{Itv}_0}, \varepsilon, \top_{\mathcal{B}^{\#}}, \top_{\text{Attr}^{\#}} \right)$

Definition

The concretization function $\gamma_{\mathcal{V}_{\mathcal{B}}^{\#}} : \mathcal{V}_{\mathcal{B}}^{\#} \rightarrow \mathcal{P}(\mathcal{V})$ is defined as:

$$\gamma_{\mathcal{V}_{\mathcal{B}}^{\#}}([l, u], \mathbf{p}, s, a) \stackrel{\text{def}}{=} \left\{ [c_1, \dots, c_n]_{\mathcal{A}} \in \mathcal{V} \left| \begin{array}{l} n \in \gamma_{\text{Itv}_0}([l, u]), \mathcal{A} \in \gamma_{\text{Attr}^{\#}}(a), \\ \forall i \in [1, \min(n, |\mathbf{p}|)] : c_i \in \gamma_{\mathcal{B}^{\#}}(p_i), \\ \forall j \in [|\mathbf{p}| + 1, n] : c_j \in \gamma_{\mathcal{B}^{\#}}(s) \end{array} \right. \right\}$$

Theorem

Meet Preservation

The concretization $\gamma_{\mathcal{V}_{\mathcal{B}}^{\#}}$ preserves the meet:

$$\gamma_{\mathcal{V}_{\mathcal{B}}^{\#}}(\nu_1 \sqcap_{\mathcal{V}_{\mathcal{B}}^{\#}} \nu_2) = \gamma_{\mathcal{V}_{\mathcal{B}}^{\#}}(\nu_1) \cap \gamma_{\mathcal{V}_{\mathcal{B}}^{\#}}(\nu_2)$$

Theorem

Galois Connection

$$\langle \mathcal{P}(\mathcal{V}), \subseteq \rangle \stackrel{\gamma_{\mathcal{V}_{\mathcal{B}}^{\#}}}{\underset{\alpha_{\mathcal{V}_{\mathcal{B}}^{\#}}}{\rightleftarrows}} \langle \mathcal{V}_{\mathcal{B}}^{\#}, \sqsubseteq_{\mathcal{V}_{\mathcal{B}}^{\#}} \rangle$$

where $\alpha_{\mathcal{V}_{\mathcal{B}}^{\#}}(C) \stackrel{\text{def}}{=} \sqcap_{\mathcal{V}_{\mathcal{B}}^{\#}} \left\{ \nu \in \mathcal{V}_{\mathcal{B}}^{\#} \mid C \subseteq \gamma_{\mathcal{V}_{\mathcal{B}}^{\#}}(\nu) \right\}$.

4. Examples of Abstract Operators

$$\text{Recycle}^\# : \mathcal{V}_B^\# \times \mathcal{V}_B^\# \rightarrow \mathcal{V}_B^\# \times \mathcal{V}_B^\#$$

Extends shorter vector to match longer one. Join over all possible target lengths in $[l', u'] = [\max(l_1, l_2), \max(u_1, u_2)]$.

The two helper functions that perform the actual recycling are:

- $\rho_c^\#$: cycles known content cyclically to a fixed target length.
- $\rho_f^\#$: joins $\rho_c^\#$ over all possible original lengths.

$$\nu'_i = \left([l', u'], \bigcup_{d=l'}^{\max(m,n)} \mathcal{B}^{\#*} \rho_f^\#(\mathbf{p}_i, s_i, l_i, d), s'_i, a_i \right)$$

$$\text{Select}^\# : \mathcal{V}_{\mathcal{B}}^\# \times \mathcal{V}_{\text{ltv}}^\# \rightarrow \mathcal{V}_{\mathcal{B}}^\#$$

The concrete `Select` handles three indexing modes. The abstract operator $\text{Select}^\#$ analyzes each mode separately:

Positive $\text{Select}_{\text{pos}}^\#$

Indices ≥ 0 or NA.
Extracts elements by
position.

Negative $\text{Select}_{\text{neg}}^\#$

Indices ≤ -0 .
Excludes elements by
position.

Logical $\text{Select}_{\text{log}}^\#$

Boolean mask $\in \{\text{NA}, \text{True}, \text{False}\}^+$.
Selects where mask is
true.

Abstract Filtering (positive/negative) : The selector content p_2 is partitioned by concretization sign:

$$p_2^+ = \langle p \in p_2 \mid \gamma_{\text{ltv}}(p) \subseteq \mathbb{Z}_{\geq 0} \cup \{\text{NA}\} \rangle$$

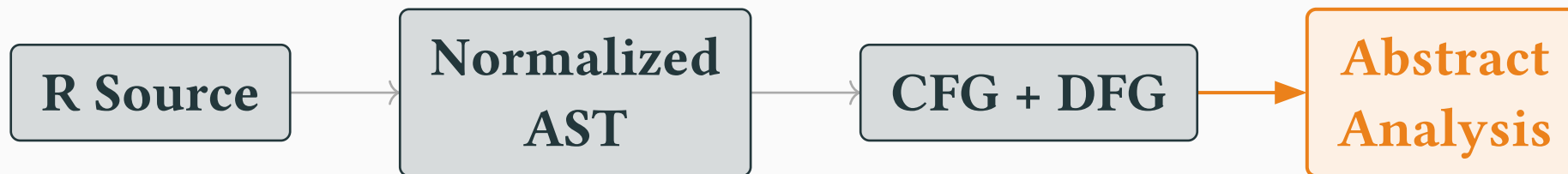
$$p_2^- = \langle p \in p_2 \mid \gamma_{\text{ltv}}(p) \subseteq \mathbb{Z}_{\leq 0} \rangle$$

The overall operator joins the two integer-indexing branches:

$$\text{Select}^\#(\nu_1, \nu_2) = \text{Select}_{\text{pos}}^\#(\nu_1, \nu_2^+) \sqcup_{\mathcal{V}_B^\#} \text{Select}_{\text{neg}}^\#(\nu_1, \nu_2^-)$$

Zero indices are handled by `AdjustForZeros` (discards definite zeros, joins over possible zeros).

5. Implementation & Validation



VectorInferenceVisitor

- Extends flowR's AbstractInterpretationVisitor
- Generic fixpoint engine: worklist algorithm, join at merge points, widening $\nabla_{\mathcal{V}_B^\#}$ at loops
- Dispatches each CFG vertex to the corresponding abstract operator

r

```
1 v <- 1
  ↳ v := ([1, 1], ⟨[1, 1]⟩, ⊥ltv)
2 if (*) {
3   v <- c(1, 2, 3)
4 } else {
5   v <- c(0, 1)
6 }
  ↳ v := ([2, 3], ⟨[0, 1][1, 2][3, 3]⟩, ⊥ltv)
7 r <- v + c(9, 10, 11)
  ↳ r := ([3, 3], ⟨[9, 10][11, 12][11, 14]⟩, ⊥ltv)
```

Figure 1: An example from the handcrafted test-suite.

6. Conclusion

Contributions

1. μR : formal core calculus $\rightarrow$ complete concrete semantics
2. $\mathcal{V}_{\mathcal{B}}^{\#}$: a parametric abstract domain $\rightarrow$ lattice, widening, Galois connection
3. Abstract operators for all vector ops $\rightarrow$ soundness proofs
4. Proof-of-concept in flowR $\rightarrow$ 61 test cases validated

Limitations & Future Work

- Not modeled: lazy evaluation, metaprogramming, S3/S4 dispatch
- Future: relational domain, real R packages, full-language analysis

Bibliography

- Cousot, Patrick, and Radhia Cousot. 1977. “Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints.” *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, 238–52. <https://doi.org/10.1145/512950.512973>.
- CRAN. 2026. “Contributed Packages.”. <https://cran.r-project.org/web/packages/>.
- Sihler, Florian, and Matthias Tichy. 2025. “Statically Analyzing the Dataflow of R Programs.” *Proc. ACM Program. Lang.* 9 (OOPSLA2): 1034–62. <https://doi.org/10.1145/3763087>.

7. Appendix

Definition

 $\sqsubseteq_{\mathcal{V}_{\mathcal{B}}^{\#}}$

Let $\nu_1 = (L_1, \mathbf{p}_1, s_1, a_1)$ and $\nu_2 = (L_2, \mathbf{p}_2, s_2, a_2)$ be minimum representatives with $k_1 = |\mathbf{p}_1|$, $k_2 = |\mathbf{p}_2|$. Then:

$$\nu_1 \sqsubseteq_{\mathcal{V}_{\mathcal{B}}^{\#}} \nu_2 \iff L_1 \sqsubseteq_{\text{ltv}_0} L_2 \wedge s_1 \sqsubseteq_{\mathcal{B}^{\#}} s_2 \wedge a_1 \sqsubseteq_{\text{Attr}^{\#}} a_2 \wedge K(\nu_1, \nu_2)$$

where $K(\nu_1, \nu_2)$ distinguishes three cases on the known content:

$$K(\nu_1, \nu_2) \stackrel{\text{def}}{=} \begin{cases} \forall i \in [1, k_1] : p_{1,i} \sqsubseteq_{\mathcal{B}^{\#}} p_{2,i} & \text{if } k_1 \leq k_2 \wedge u_1 \neq +\infty \\ \forall i \in [1, k_1] : p_{1,i} \sqsubseteq_{\mathcal{B}^{\#}} p_{2,i} \\ \wedge \forall j \in [k_1 + 1, k_2] : s_1 \sqsubseteq_{\mathcal{B}^{\#}} p_{2,j} & \text{if } k_1 \leq k_2 \wedge u_1 = +\infty \\ \forall i \in [1, k_2] : p_{1,i} \sqsubseteq_{\mathcal{B}^{\#}} p_{2,i} \\ \wedge \forall j \in [k_2 + 1, k_1] : p_{1,j} \sqsubseteq_{\mathcal{B}^{\#}} s_2 & \text{if } k_1 > k_2 \end{cases}$$

Definition
 $\sqcup_{\mathcal{V}_{\mathcal{B}}^{\#}}$

Let $\nu_1 = (L_1, \mathbf{p}_1, s_1, a_1)$ and $\nu_2 = (L_2, \mathbf{p}_2, s_2, a_2)$ with $k_1 = |\mathbf{p}_1|$, $k_2 = |\mathbf{p}_2|$. Then:

$$\nu_1 \sqcup_{\mathcal{V}_{\mathcal{B}}^{\#}} \nu_2 \stackrel{\text{def}}{=} (L_1 \sqcup_{\text{ltv}_0} L_2, \mathbf{p}_1 \sqcup_{\mathcal{B}^{\#}^*} \mathbf{p}_2, s_1 \sqcup_{\mathcal{B}^{\#}} s_2, a_1 \sqcup_{\text{Attr}^{\#}} a_2)$$

where $\mathbf{p}_1 \sqcup_{\mathcal{B}^{\#}^*} \mathbf{p}_2 \stackrel{\text{def}}{=} \langle p_{1,i} \sqcup_{\mathcal{B}^{\#}} p_{2,i} \rangle_{i=1}^m \oplus \text{tail}$ and:

$$\text{tail} = \begin{cases} \langle p_{1,m+1} \sqcup_{\mathcal{B}^{\#}} s_2 \cdots p_{1,k_1} \sqcup_{\mathcal{B}^{\#}} s_2 \rangle & \text{if } k_1 > k_2 \\ \langle p_{2,m+1} \sqcup_{\mathcal{B}^{\#}} s_1 \cdots p_{2,k_2} \sqcup_{\mathcal{B}^{\#}} s_1 \rangle & \text{if } k_2 > k_1 \\ \varepsilon & \text{otherwise} \end{cases}$$

with $m = \min(k_1, k_2)$. The shorter known content is padded with the other vector's summary. The attribute join is $a_1 \sqcup_{\text{Attr}\#} a_2 = (\text{MustHave}_1 \cap \text{MustHave}_2, \text{MayHave}_1 \cup \text{MayHave}_2)$.

Definition
 $\sqcap_{\mathcal{V}_{\mathcal{B}}^{\#}}$

Let $\nu_1 = (L_1, \mathbf{p}_1, s_1, a_1)$ and $\nu_2 = (L_2, \mathbf{p}_2, s_2, a_2)$ with $L_1 = [l_1, u_1]$, $L_2 = [l_2, u_2]$, $k_1 = |\mathbf{p}_1|$, $k_2 = |\mathbf{p}_2|$, $M = \max(k_1, k_2)$. Then:

$$\nu_1 \sqcap_{\mathcal{V}_{\mathcal{B}}^{\#}} \nu_2 \stackrel{\text{def}}{=} \begin{cases} \perp_{\mathcal{V}_{\mathcal{B}}^{\#}} & \text{if } \max(l_1, l_2) > u' \\ \xi_1 & \text{if } u_1 = u_2 = +\infty \wedge s_1 \sqcap_{\mathcal{B}^{\#}} s_2 \neq \perp_{\mathcal{B}^{\#}} \wedge u' = M \\ \xi_2 & \text{otherwise} \end{cases}$$

where:

$$\xi_1 \stackrel{\text{def}}{=} ([\max(l_1, l_2), +\infty], \mathbf{p}', s_1 \sqcap_{\mathcal{B}^{\#}} s_2, a_1 \sqcap_{\text{Attr}^{\#}} a_2),$$

$$\xi_2 \stackrel{\text{def}}{=} ([\max(l_1, l_2), u'], \mathbf{p}', \perp_{\mathcal{B}^{\#}}, a_1 \sqcap_{\text{Attr}^{\#}} a_2).$$

The resulting known content is $\mathbf{p}' = \langle p'_i \rangle_{i=1}^{u'}$ built with elements $p'_i = p'_{1,i} \sqcap_{\mathcal{B}^\#} p'_{2,i}$, and:

$$p'_{1,i} = \begin{cases} p_{1,i} & \text{if } i \leq k_1 \\ s_1 & \text{if } i > k_1 \end{cases} \quad p'_{2,i} = \begin{cases} p_{2,i} & \text{if } i \leq k_2 \\ s_2 & \text{if } i > k_2 \end{cases}$$

with $u' = \max\{i \in [0, \max(l_1, l_2, M)] \mid \forall j \in [1, i] : p'_j \neq \perp_{\mathcal{B}^\#}\}$. The known content is truncated at the first position where the component meet is $\perp_{\mathcal{B}^\#}$.

Definition

 $\gamma_{\mathcal{V}_B^\#}$

The concretization function $\gamma_{\mathcal{V}_B^\#} : \mathcal{V}_B^\# \rightarrow \mathcal{P}(\mathcal{V})$ is defined as:

$$\gamma_{\mathcal{V}_B^\#}([l, u], \mathbf{p}, s, a) \stackrel{\text{def}}{=} \left\{ [c_1, \dots, c_n]_{\mathcal{A}} \in \mathcal{V} \left| \begin{array}{l} n \in \gamma_{\text{Itv}_0}([l, u]), \mathcal{A} \in \gamma_{\text{Attr}^\#}(a), \\ \forall i \in [1, \min(n, |\mathbf{p}|)] : c_i \in \gamma_{\mathcal{B}^\#}(p_i), \\ \forall j \in [|\mathbf{p}| + 1, n] : c_j \in \gamma_{\mathcal{B}^\#}(s) \end{array} \right. \right\}$$

A Galois connection holds: $\mathcal{P}(\mathcal{V}) \xrightleftharpoons[\alpha_{\mathcal{V}_B^\#}]{\gamma_{\mathcal{V}_B^\#}} \mathcal{V}_B^\#$, with $\alpha_{\mathcal{V}_B^\#}(C) = \bigcap_{\nu \in \mathcal{V}_B^\#} \left\{ \nu \in \mathcal{V}_B^\# \mid C \subseteq \gamma_{\mathcal{V}_B^\#}(\nu) \right\}$.

Definition

 $\nabla_{\mathcal{V}_{\mathcal{B}}^{\#}}$

Let $\nu_1 = (L_1, \mathbf{p}_1, s_1, a_1)$ and $\nu_2 = (L_2, \mathbf{p}_2, s_2, a_2)$ with $k_1 = |\mathbf{p}_1|$, $k_2 = |\mathbf{p}_2|$, $m = \min(k_1, k_2)$, $M = \max(k_1, k_2)$. Then:

$$\nu_1 \nabla_{\mathcal{V}_{\mathcal{B}}^{\#}} \nu_2 \stackrel{\text{def}}{=} ([l', u'], \mathbf{p}', s', a_1 \sqcup_{\text{Attr}^{\#}} a_2)$$

where $[l', u'] = L_1 \nabla_{\text{Itv}_0} L_2$. The content extends to the maximum length when the widened length is finite or both inputs are finite:

$$\mathbf{p}' = \begin{cases} \langle p'_1 \cdots p'_M \rangle & \text{if } u' \neq +\infty \vee (u_1 \neq +\infty \wedge u_2 \neq +\infty) \\ \langle p_{1,1} \nabla_{\mathcal{B}^{\#}} p_{2,1} \cdots p_{1,m} \nabla_{\mathcal{B}^{\#}} p_{2,m} \rangle & \text{otherwise} \end{cases}$$

where $p'_j = p_{1,j} \nabla_{\mathcal{B}^{\#}} p_{2,j}$ for $j \leq m$, and $p'_j = p_{i,j}$ for elements from the longer vector i when $m < j \leq M$. The summary:

$$s' = \begin{cases} \perp_{\mathcal{B}^\#} & \text{if } u' \neq +\infty \\ s_1 \nabla_{\mathcal{B}^\#} \left(s_2 \sqcup_{\mathcal{B}^\#} \bigsqcup_{j=m+1}^{k_1} p_{1,j} \sqcup_{\mathcal{B}^\#} \bigsqcup_{j=m+1}^{k_2} p_{2,j} \right) & \text{otherwise} \end{cases}$$

Well-founded: the widening guarantees termination of fixpoint iteration.