Static Analyses for the Properties, Programs, and People of Tomorrow

HDR Defense

30 September 2025

Caterina Urban
Inria & École Normale Supérieure | Université PSL

Static Analysis by Abstract Interpretation

Intuition

ABSTRACTION

SOUNDNESS

€ 10 +
€ 40 +
€ 30 +
€ 10

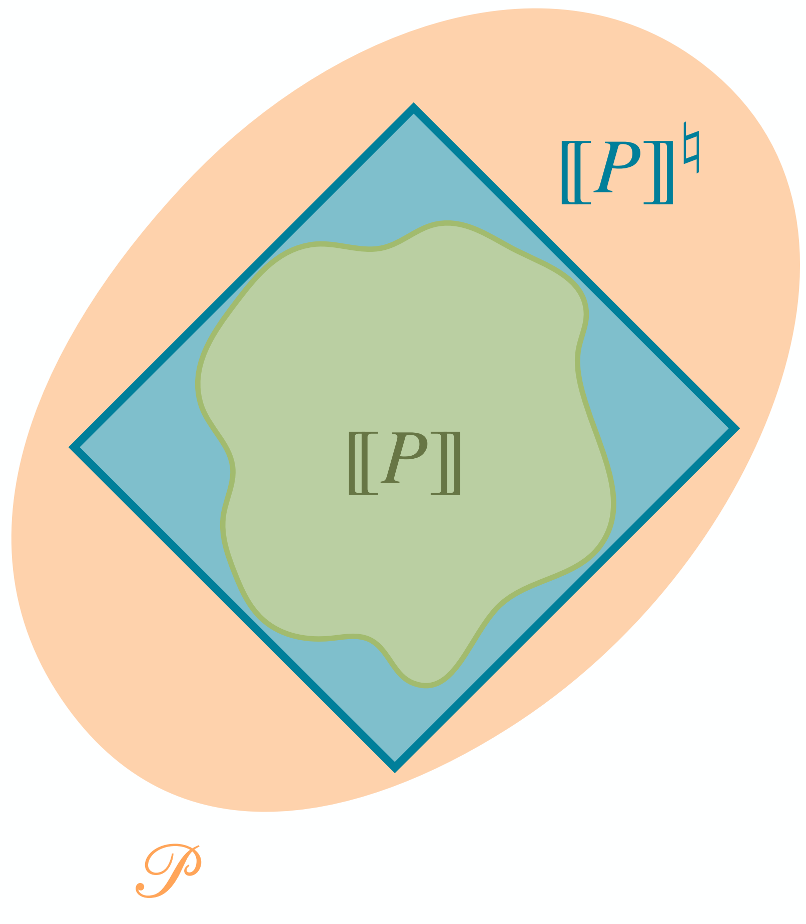
€ 90

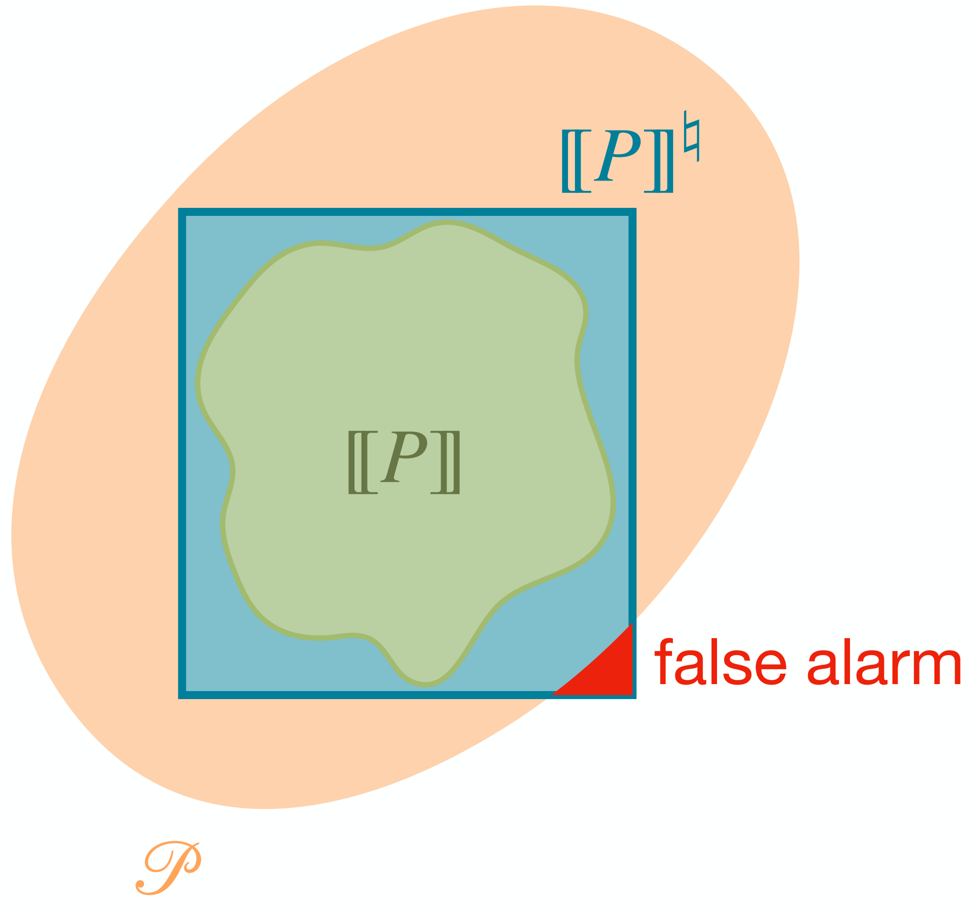
PROPERTY OF INTEREST

€ 9.95 +
€ 35.85 +
€ 27.95 +
€ 4.85

€ 78.60

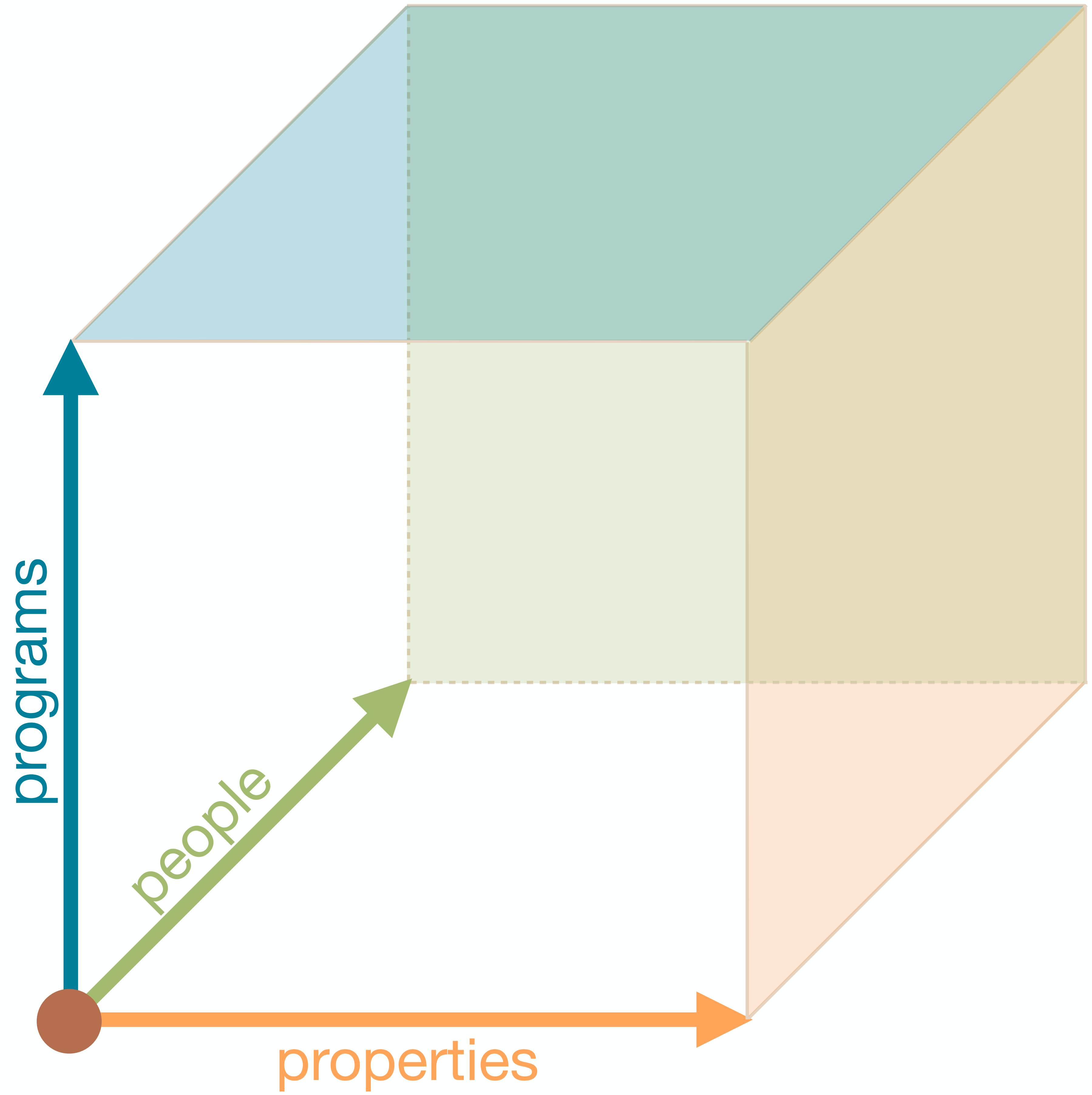
~~COMPLETENESS~~





Static Analysis by Abstract Interpretation

Where It Started, Where I am Going



CONCEPTUAL SHIFT

- from **safety (trace) properties**
  through liveness (trace) properties [PhD]
  to **program (hyper)properties**

APPLICATION SHIFT

- from **safety-critical (embedded) software**
  to **high-stakes decision-making software**

COMMUNITY SHIFT

- from **static analysis** (or formal methods) **experts**
  to **domain experts** (e.g., data scientists)

Verification

Explainability

Verification

Explainability

Input Data Usage

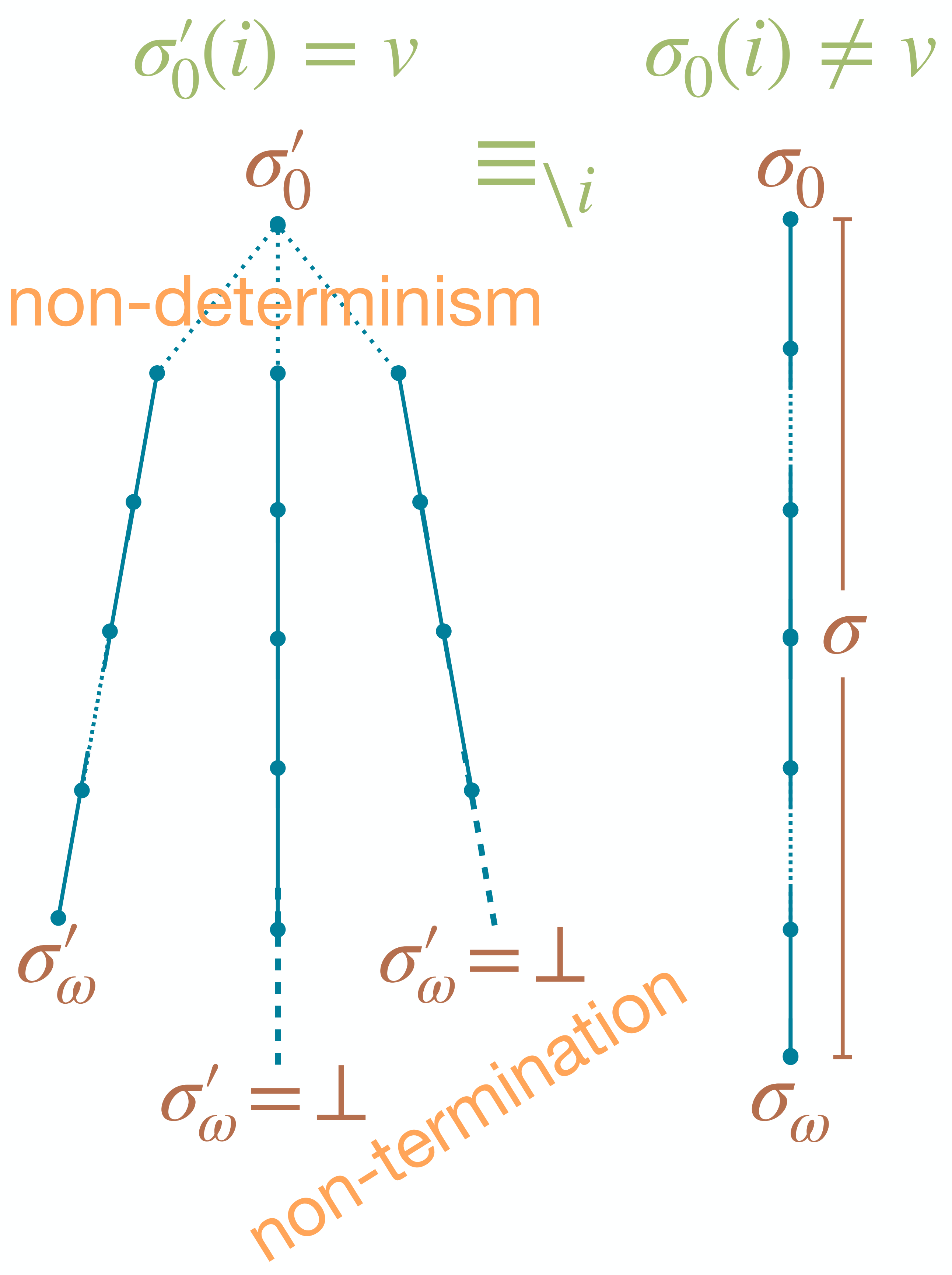
USED$_i$

[ESOP 2018]

Input Data Usage [ESOP 2018]

A Certain Outcome is Not Possible with a Certain Input Value

$$\underbrace{\text{A Certain Outcome}}_{\sigma_\omega} \text{ is Not Possible with } \underbrace{\text{a Certain Input Value}}_{v}$$



$$\text{USED}_i \stackrel{\text{def}}{=} \exists \sigma v \colon A_1 \wedge \forall \sigma' \colon A_2 \wedge B \Rightarrow C$$

$$A_1 \stackrel{\text{def}}{=} \sigma_0(i) \neq v$$

$$A_2 \stackrel{\text{def}}{=} \sigma'_0(i) = v$$

$$B \stackrel{\text{def}}{=} \sigma_0 \equiv_{\setminus i} \sigma'_0$$

$$C \stackrel{\text{def}}{=} \sigma_\omega \neq \sigma'_\omega$$

Global Prediction Stability

[OOPSLA 2020, SAS 2021, WFVML 2022]

Liveness Non-Exploitability

Termination Resilience

Data Leakage

[TASE 2024, SCP 2025]

Input Data Usage

USED_i

[ESOP 2018]

Quantitative Data Usage

[NFM 2024, SAS 2024]

[SAS 2025]

Partial Abstract Non-Interference

Partial Completeness

Global Prediction Stability

[OOPSLA 2020, SAS 2021, WFVML 2022]

Liveness Non-Exploitability

Termination Resilience

Data Leakage

[TASE 2024, SCP 2025]

Input Data Usage

USED$_i$

[ESOP 2018]

Quantitative Data Usage

[NFM 2024, SAS 2024]

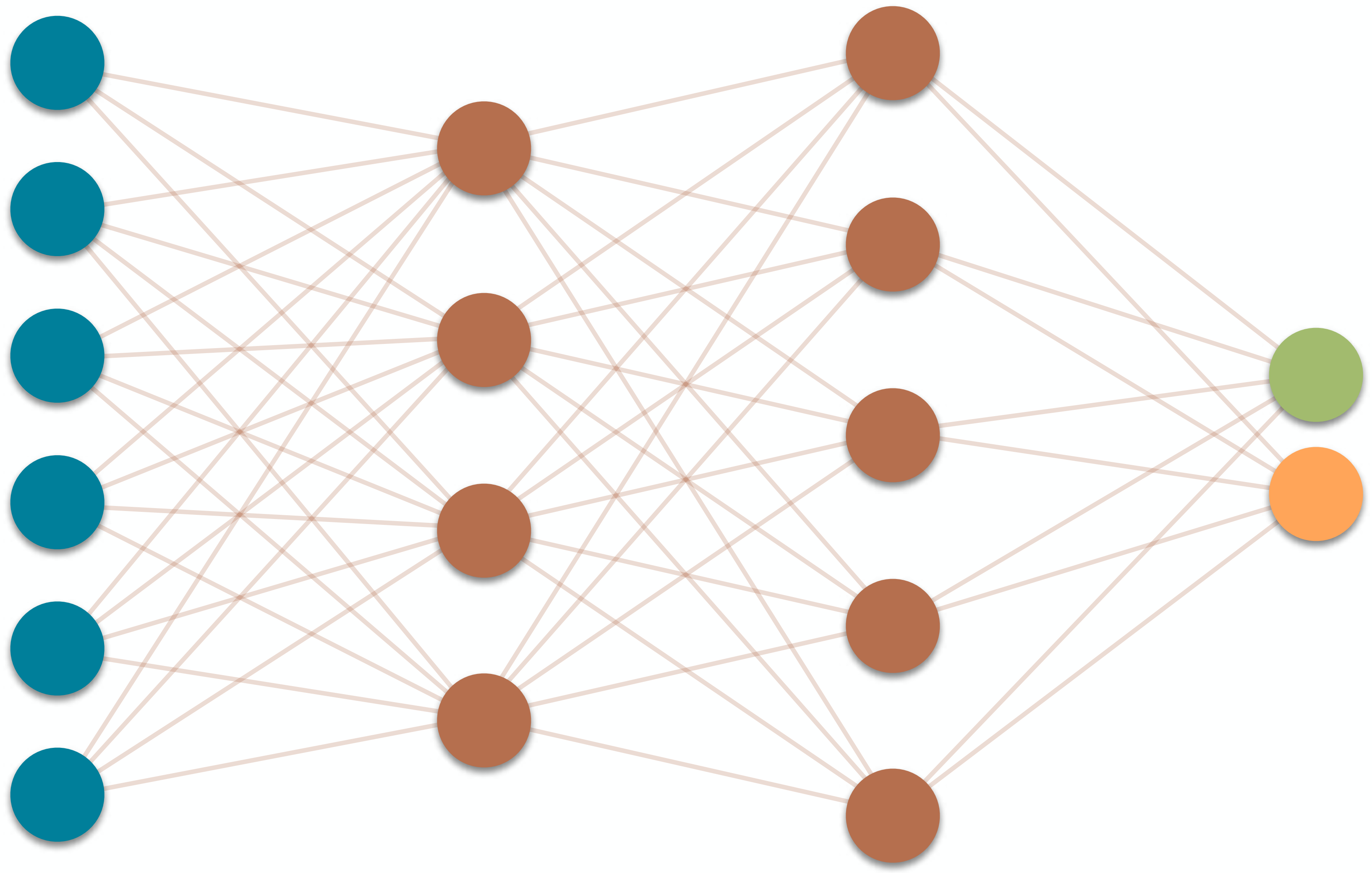
[SAS 2025]

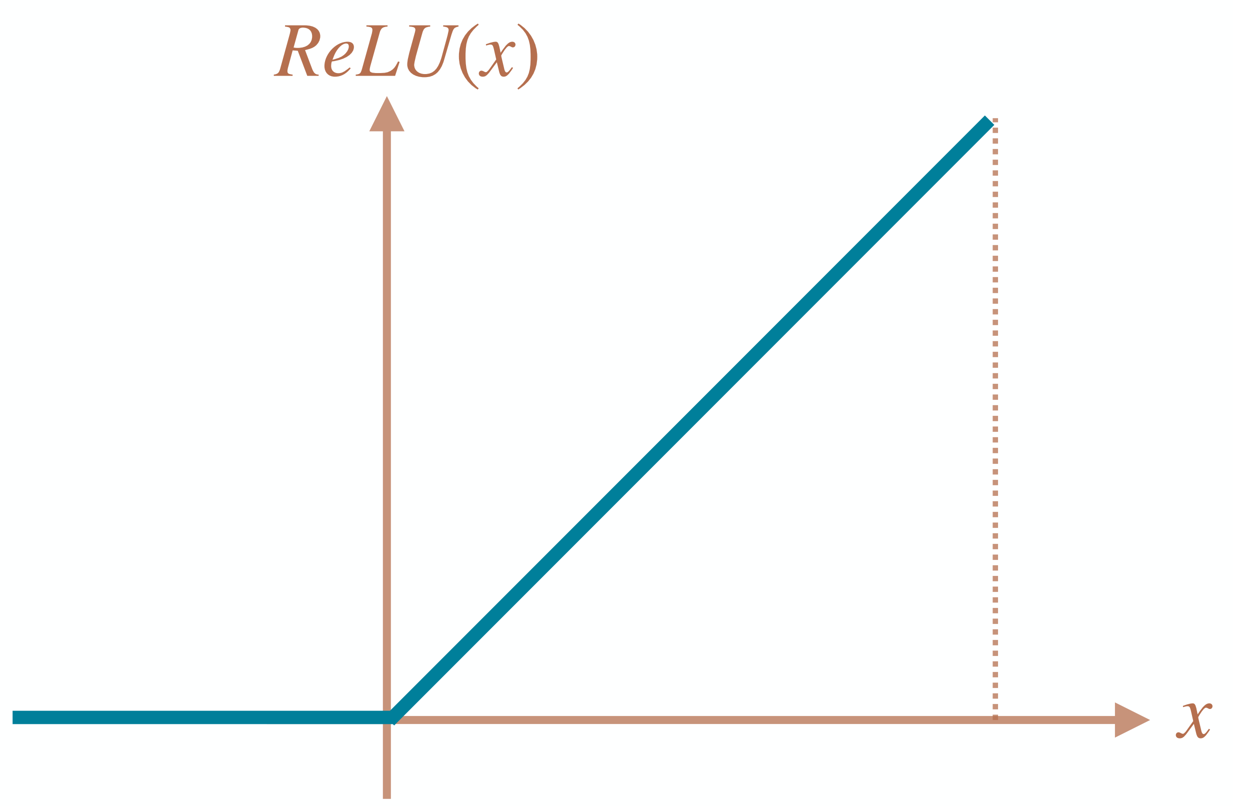
Partial Abstract Non-Interference

Partial Completeness

Neural Network Surrogates

Less Computing Power and Less Computing Time





Neural Network Surrogate

Runway Overrun Warning

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```

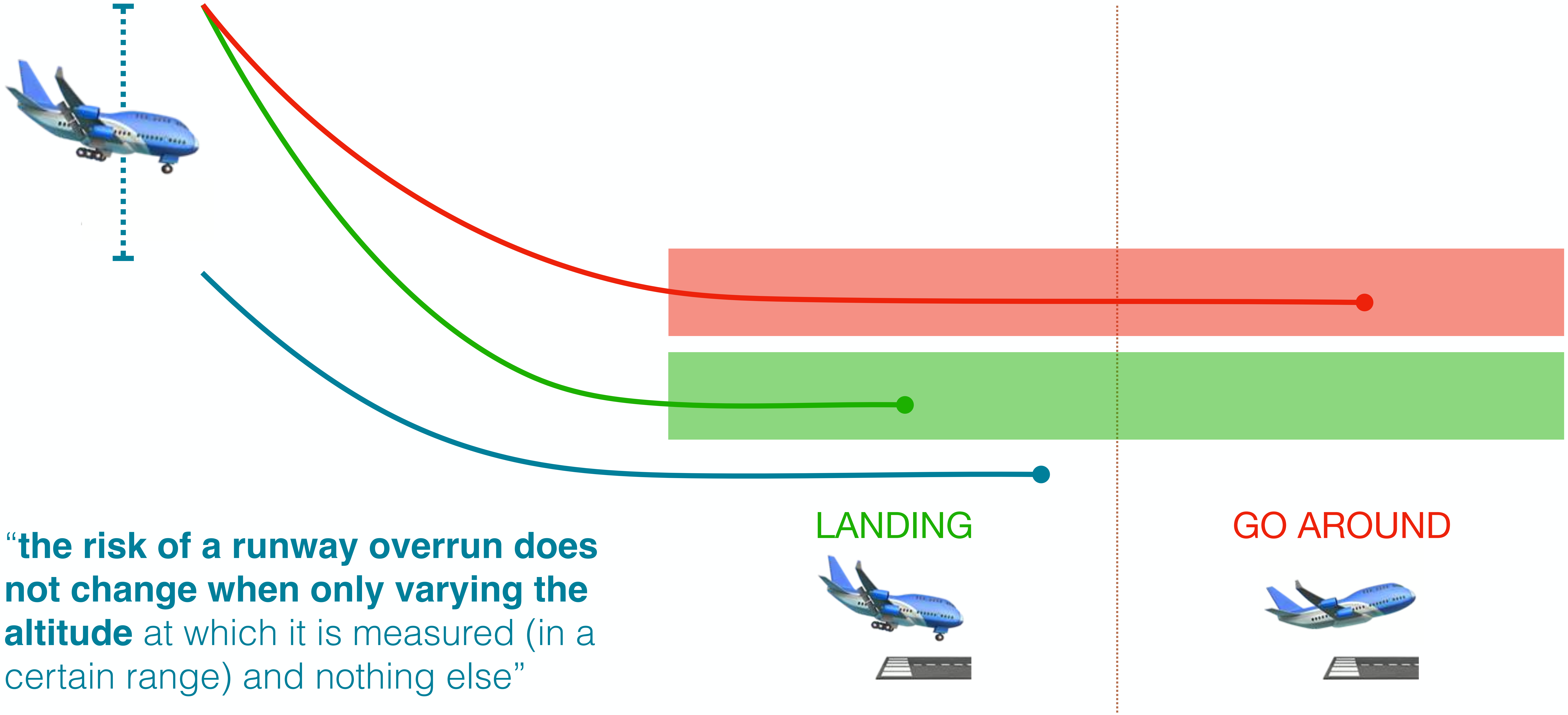
WEIGHT
TEMPERATURE
ALTITUDE
SPEED
WIND
SLOPE

```
x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```

RUNWAY LENGTH
LANDING
GO AROUND

Global Prediction Stability [collaboration with Airbus]

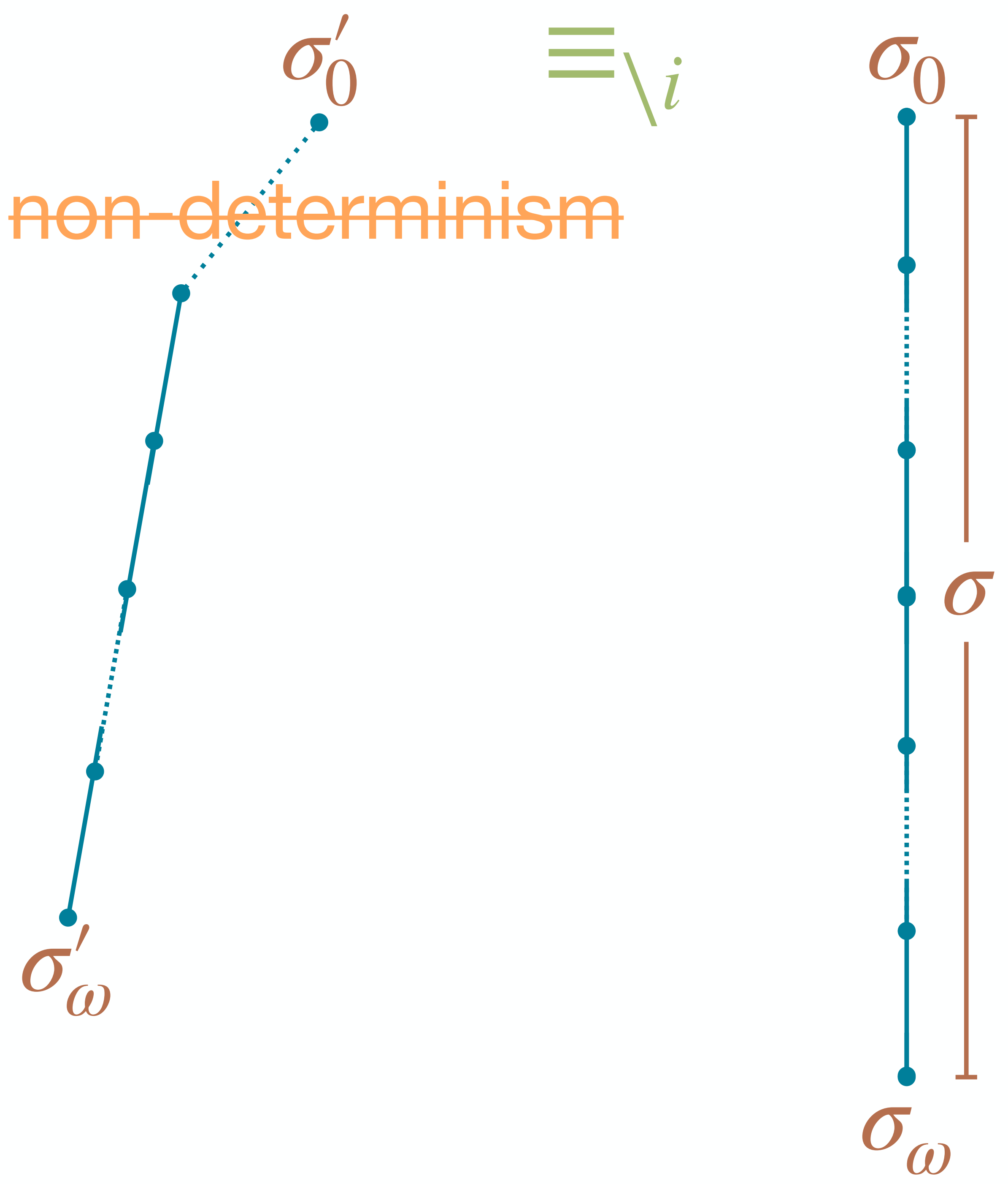
Prediction is Unaffected by Perturbations to Certain Inputs



“**the risk of a runway overrun does not change when only varying the altitude** at which it is measured (in a certain range) and nothing else”

Input Data Usage [ESOP 2018]

Neural Networks are Deterministic (and Always Terminating)



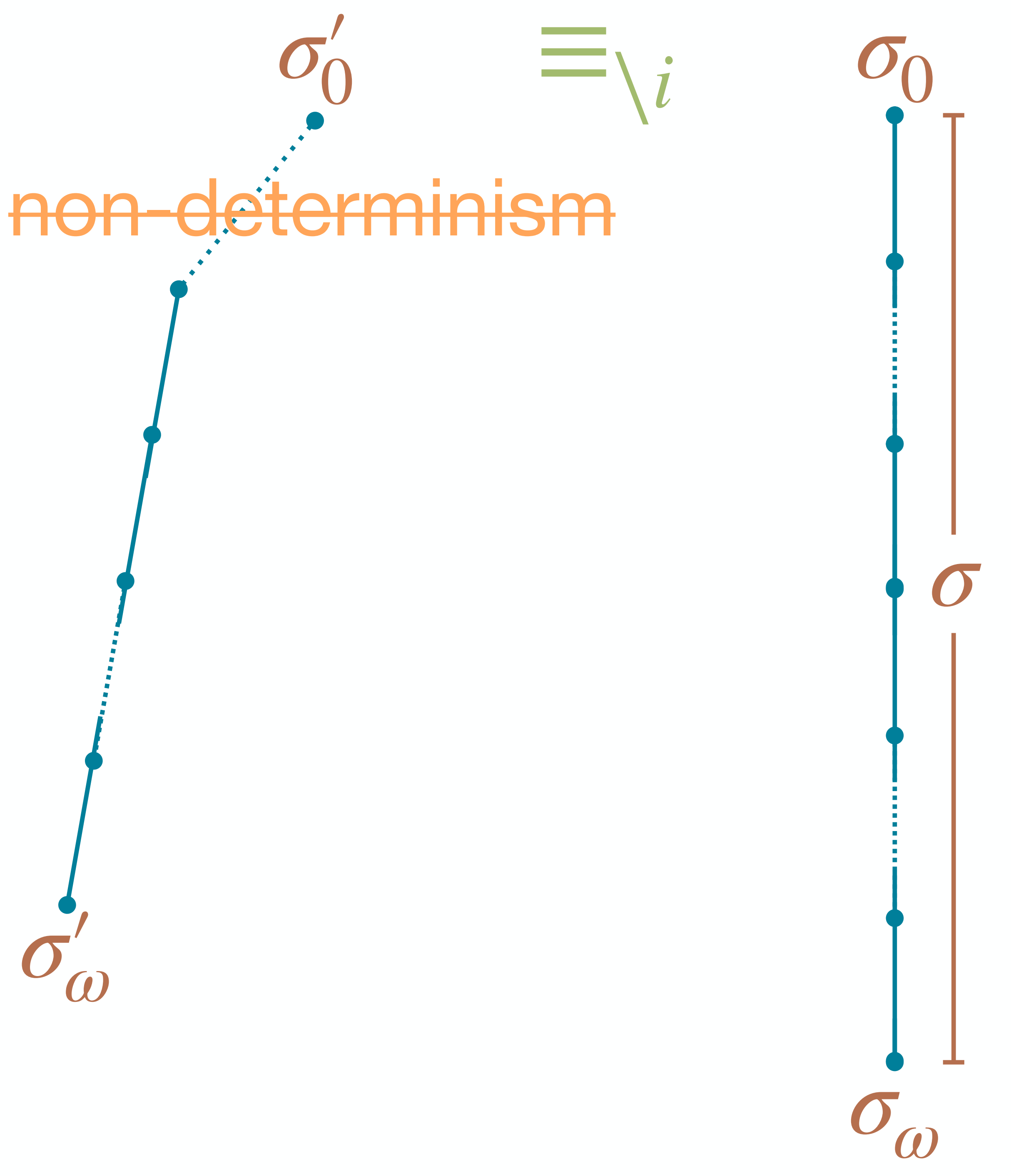
$$\text{USED}_i \stackrel{\text{def}}{=} \exists \sigma \sigma' \colon B \wedge C$$

$$B \stackrel{\text{def}}{=} \sigma_0 \equiv_{\setminus i} \sigma_0'$$

$$C \stackrel{\text{def}}{=} \sigma_\omega \neq \sigma_\omega'$$

Global Prediction Stability

Prediction is Unaffected by Perturbations to Certain Inputs



$$\neg \text{USED}_i \stackrel{\text{def}}{=} \forall \sigma\sigma' \colon B \Rightarrow \neg C$$

$$B \stackrel{\text{def}}{=} \sigma_0 \equiv_{\setminus i} \sigma'_0$$

$$\neg C \stackrel{\text{def}}{=} \sigma_\omega = \sigma'_\omega$$

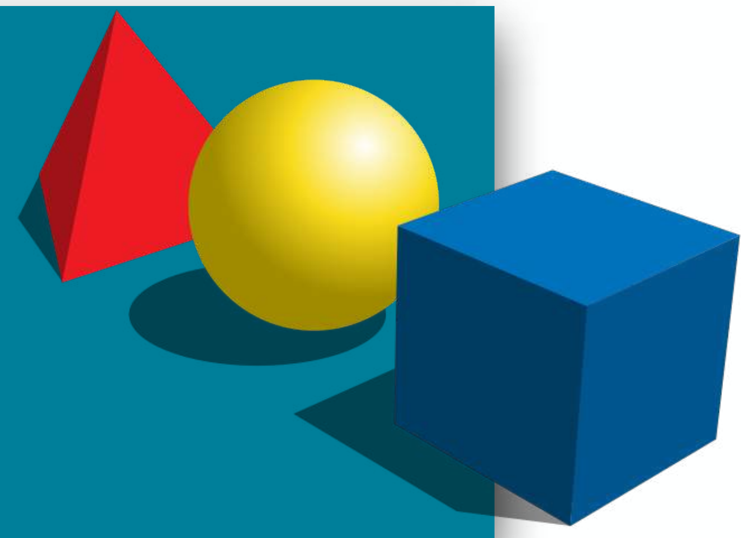
$$\mathcal{S}_i \stackrel{\text{def}}{=} \{ [\![P]\!] \mid \neg \text{USED}_i([\![P]\!]) \}$$

Global Prediction Stability Static Analysis

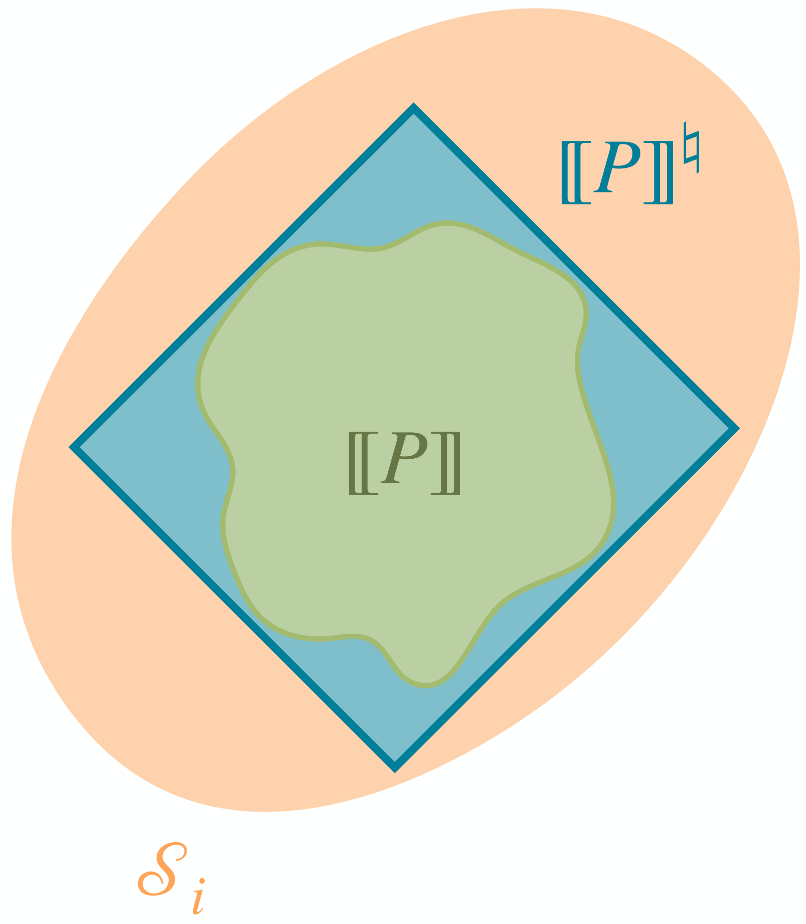
3-Step Recipe

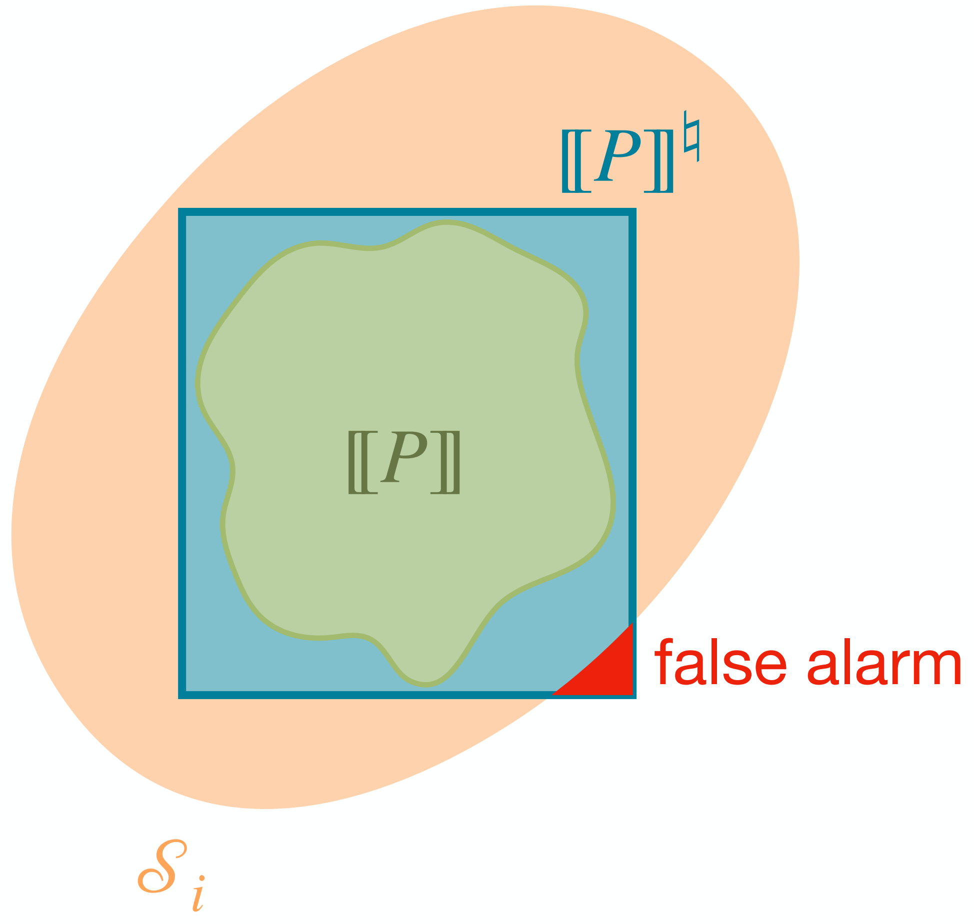
practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior





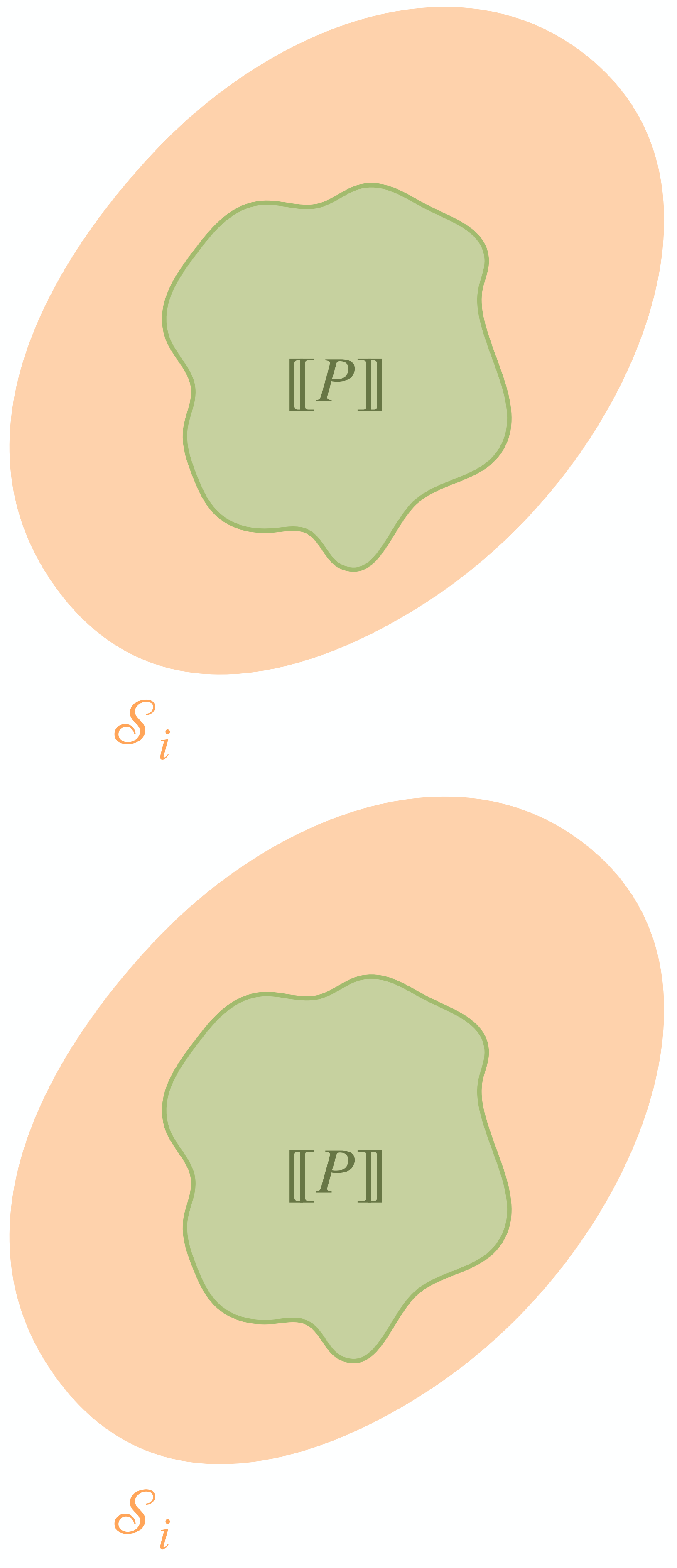
Global Prediction Stability Static Analysis

3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties

concrete semantics
mathematical models of the program behavior

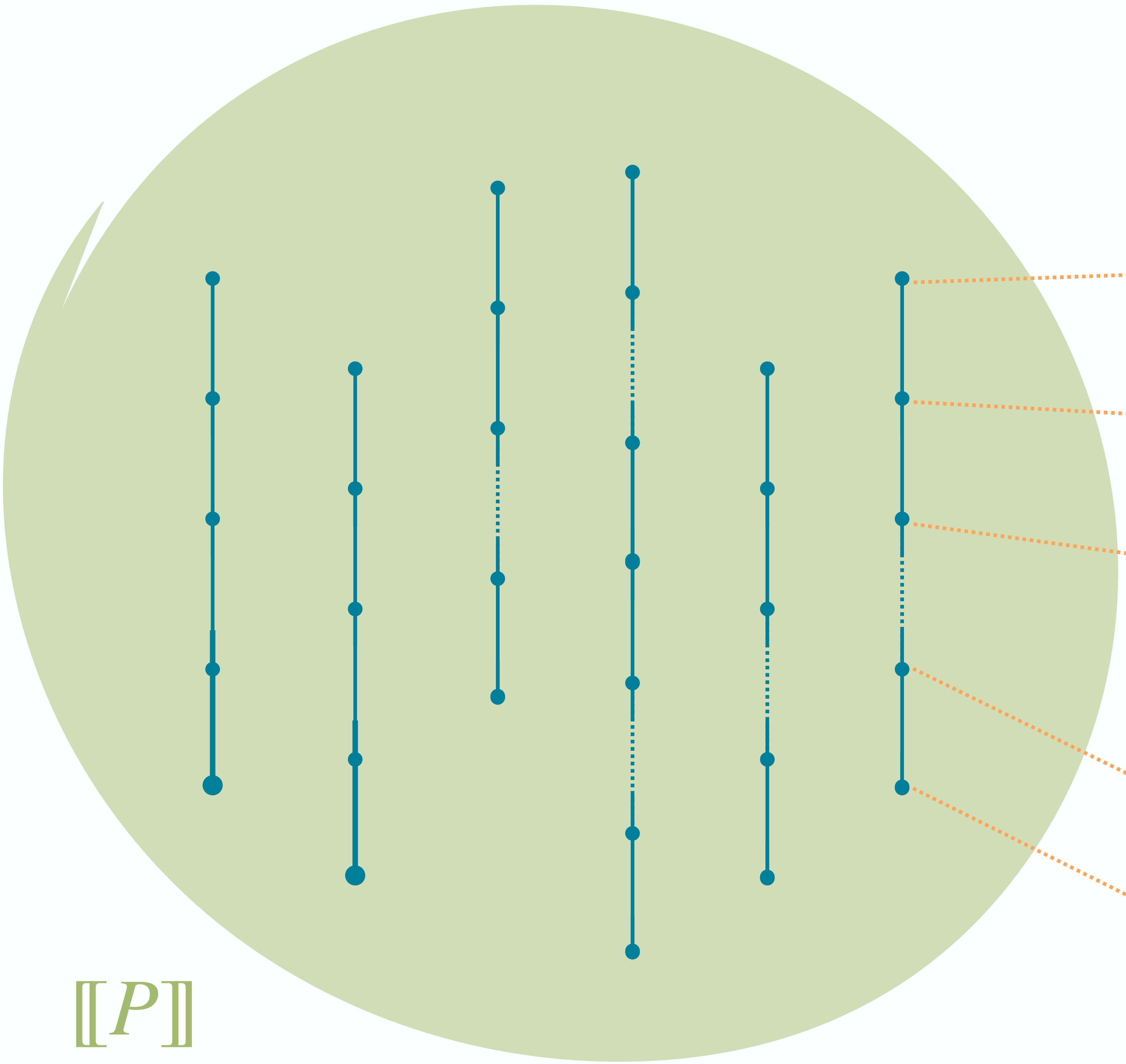


Concrete Semantics

Intuition

Concrete Semantics

(Maximal) Trace Semantics

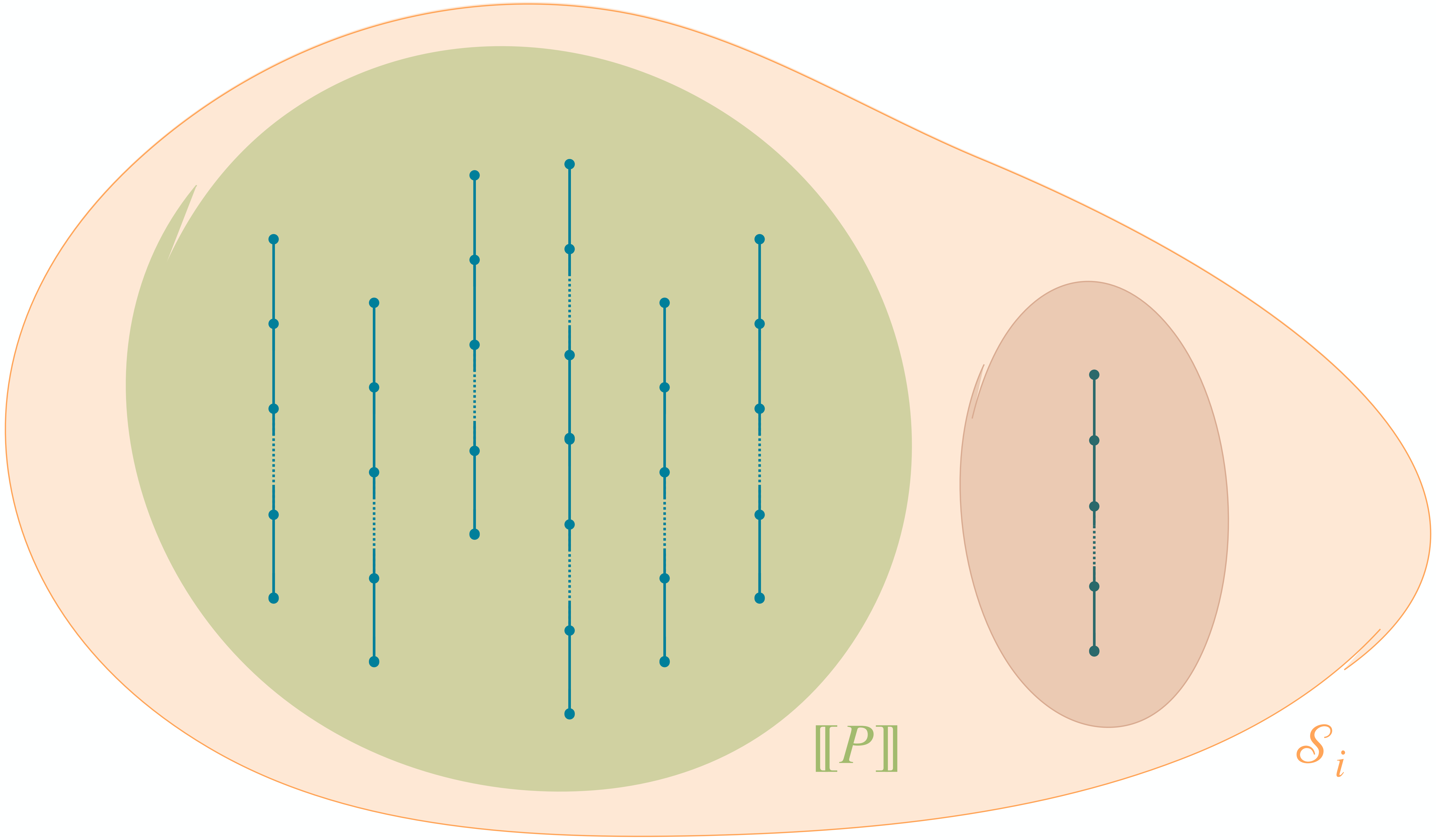


P

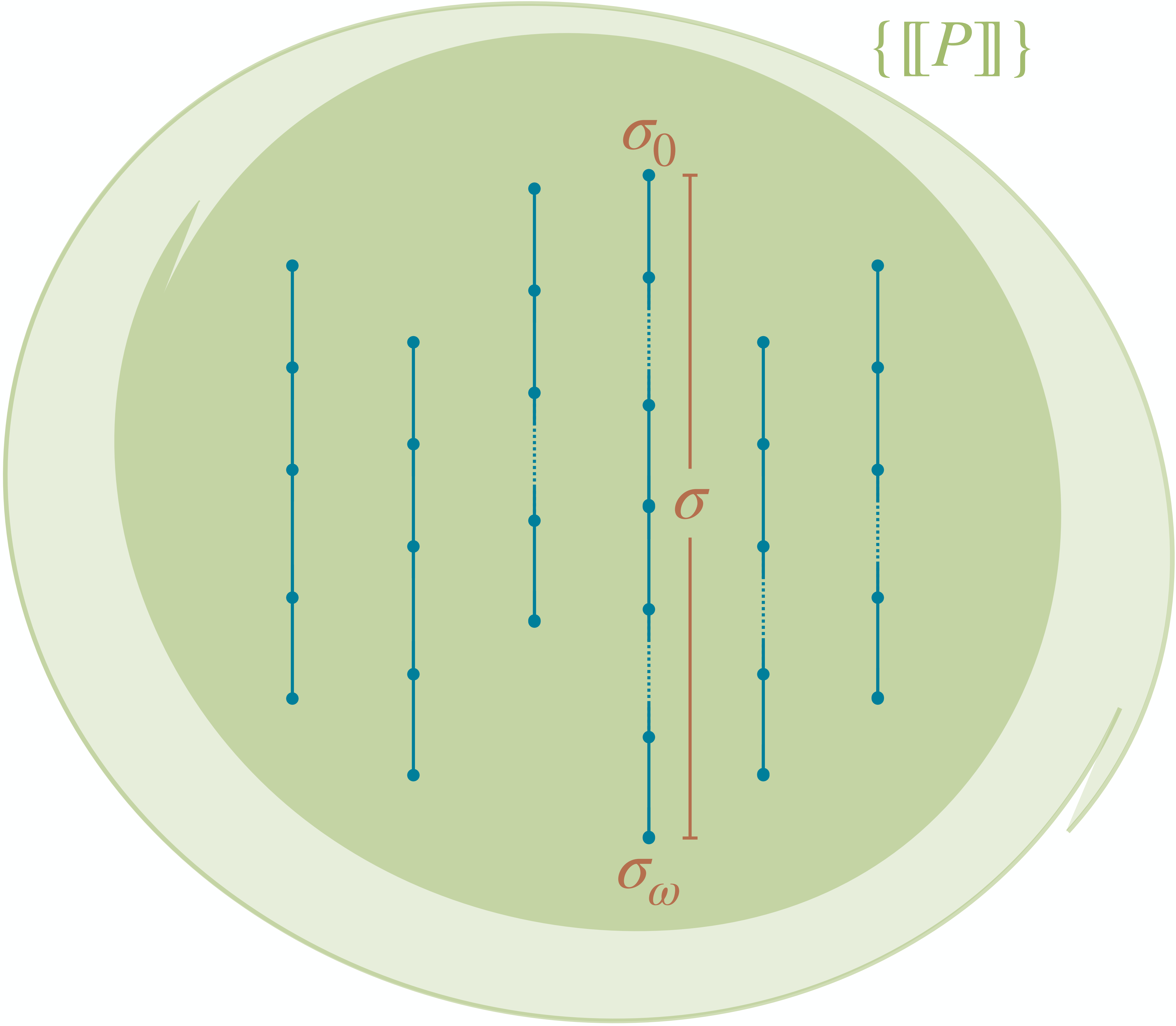
```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU(...)
x11 = ReLU(...)
x12 = ReLU(...)

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```

Global Prediction Stability Verification

$P \models \mathcal{S}_i \Leftrightarrow [\![P]\!] \in \mathcal{S}_i \Leftrightarrow \underbrace{\{[\![P]\!]\}}_{\text{Collecting Semantics}} \subseteq \mathcal{S}_i$



Hierarchy of Semantics [OOPSLA 2020]



$\{[P]\}^{\natural}_{\bullet}$ Parallel Semantics

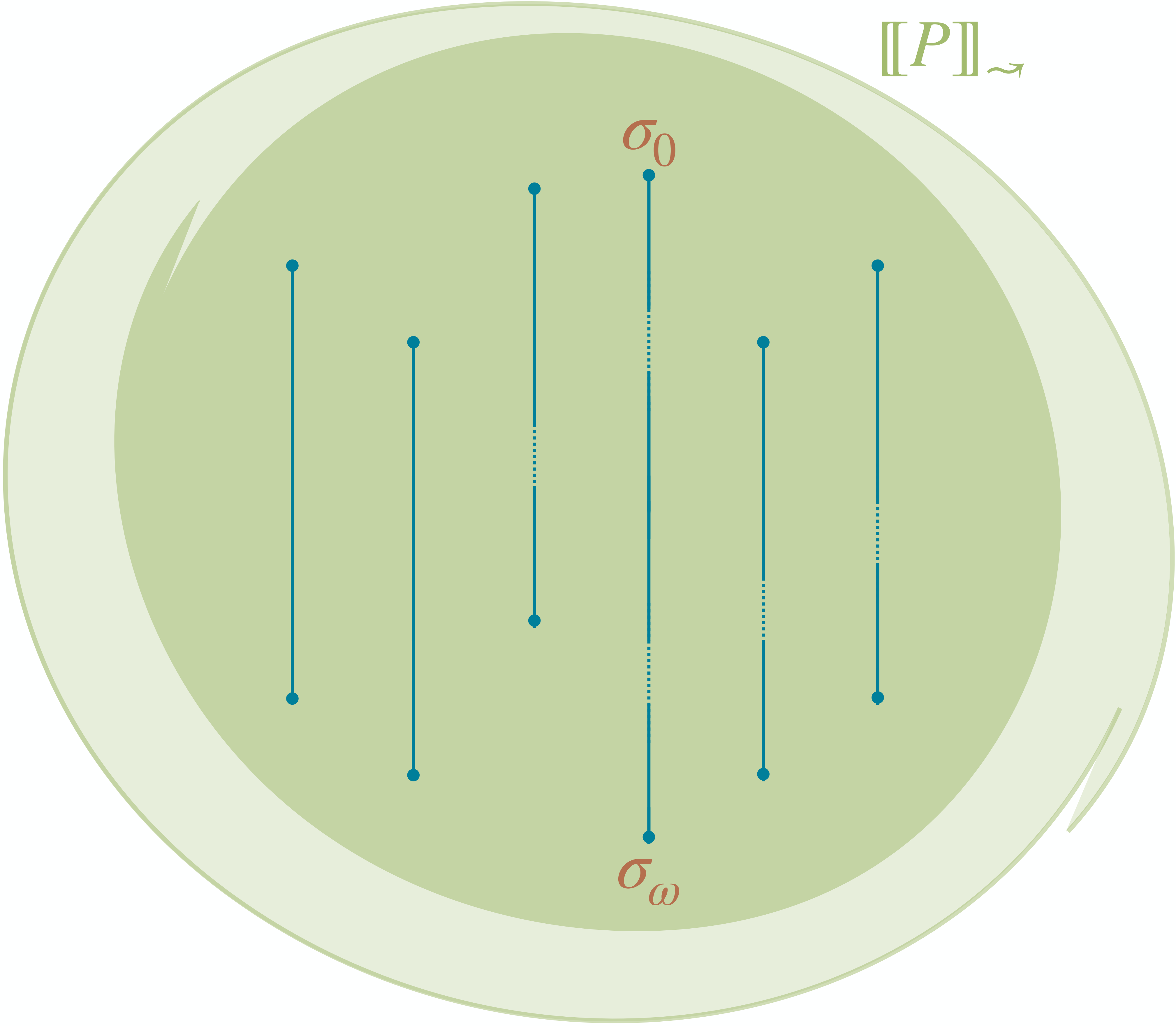
$\alpha^{\natural}_{\bullet}$

$[\![P]\!]_{\rightsquigarrow}$ Dependency Semantics

$\alpha_{\rightsquigarrow}$

$\{[\![P]\!]\}$ Collecting Semantics

Hierarchy of Semantics [OOPSLA 2020]



$\{[P]\}^{\bullet}$ Parallel Semantics

$\alpha^{\bullet}$

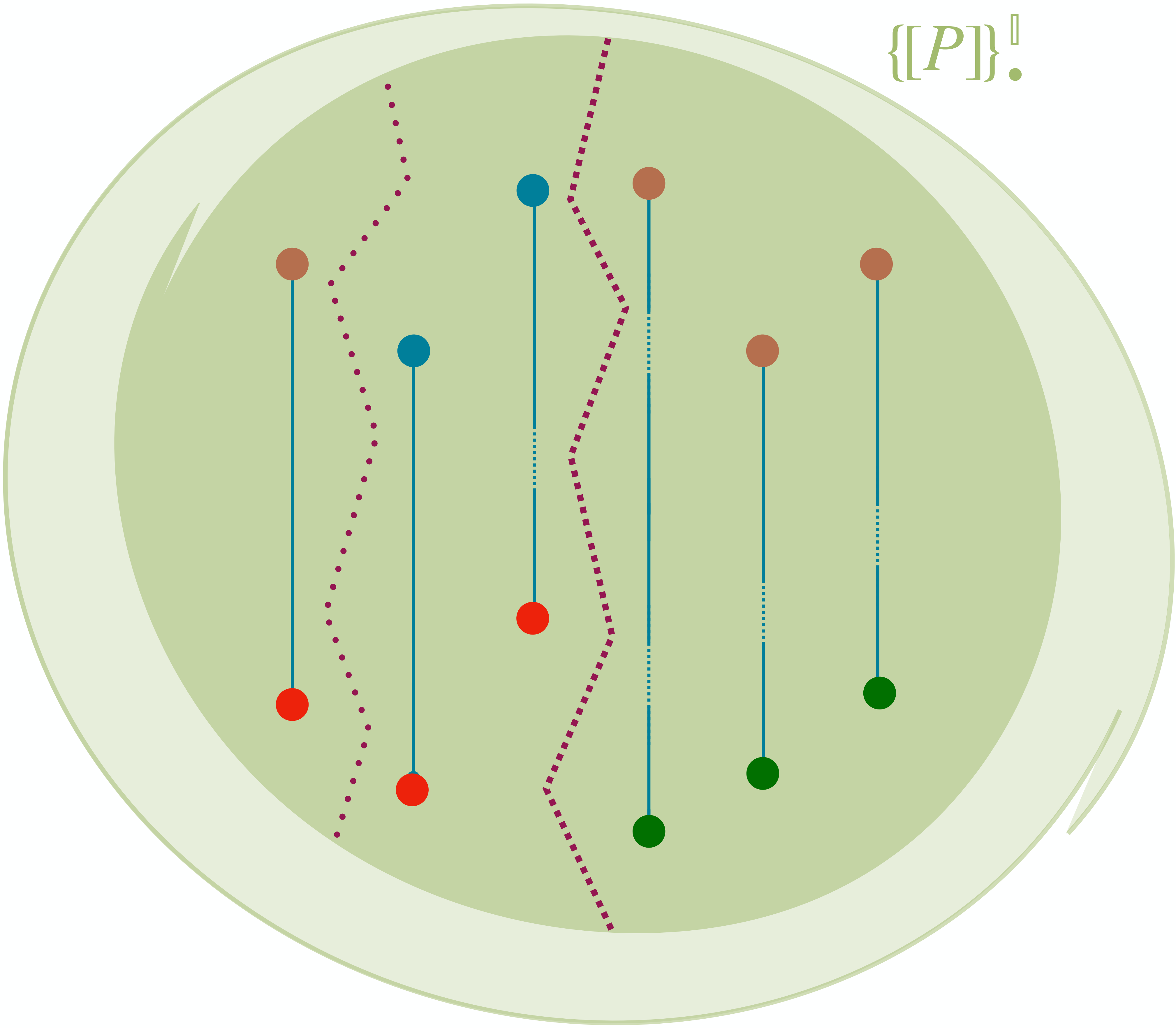
$[\![P]\!]_{\rightsquigarrow}$ Dependency Semantics

$\alpha_{\rightsquigarrow}$

$\{[\![P]\!]\}$ Collecting Semantics

$$\neg\text{USED}_i \overset{\text{def}}{=} \forall \sigma\sigma' \colon \sigma_0 \equiv_{\setminus i} \sigma'_0 \Rightarrow \sigma_\omega = \sigma'_\omega$$

Hierarchy of Semantics [OOPSLA 2020]



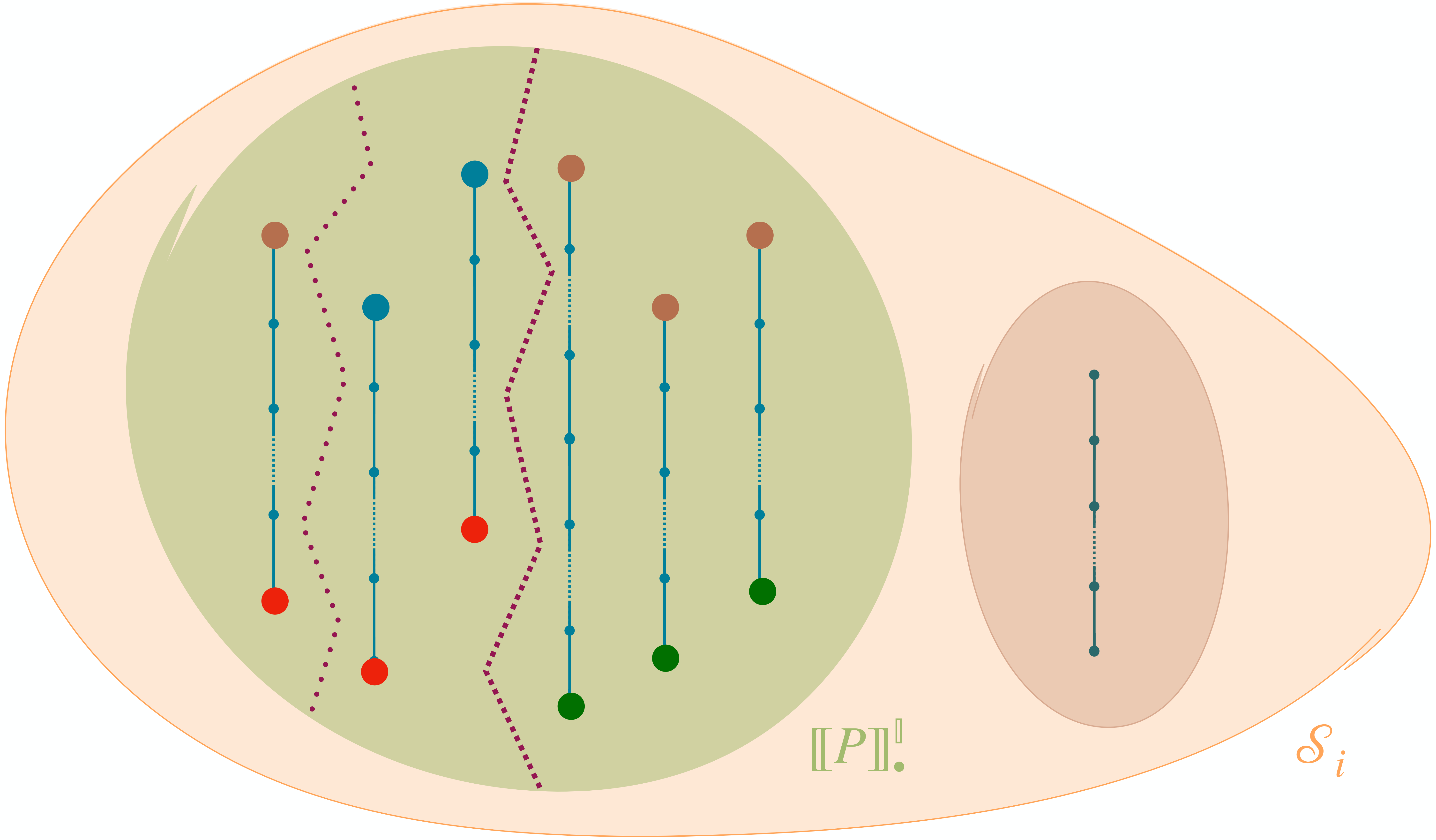
$\{[P]\}^{\natural}_{\bullet}$	Parallel Semantics
$\uparrow \alpha^{\natural}_{\bullet}$	
$[\![P]\!]_{\rightsquigarrow}$	Dependency Semantics
$\uparrow \alpha_{\rightsquigarrow}$	
$\{[\![P]\!]\}$	Collecting Semantics

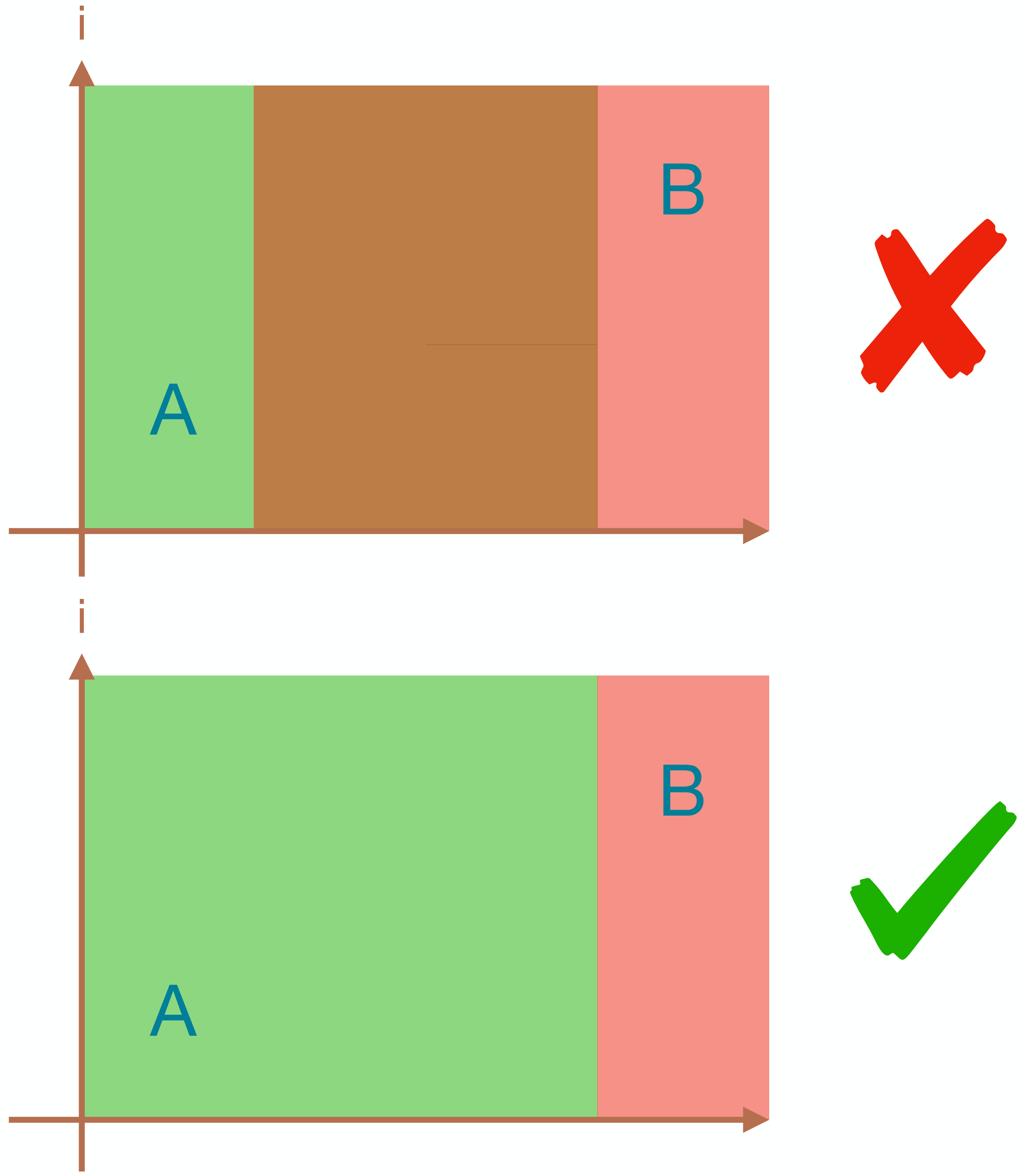
$$\neg\text{USED}_i \stackrel{\text{def}}{=} \forall \sigma\sigma' \colon \sigma_0 \equiv_{\setminus i} \sigma'_0 \Rightarrow \sigma_\omega = \sigma'_\omega$$

Global Prediction Stability Verification

$P \models \mathcal{S}_i \Leftrightarrow \forall I \in \mathbb{I}: \forall A, B \in \underbrace{[\![P]\!]^{\parallel}}: A^I_\omega \neq B^I_\omega \Rightarrow A^I_0 \not\equiv_{\setminus i} B^I_0$

Parallel Semantics

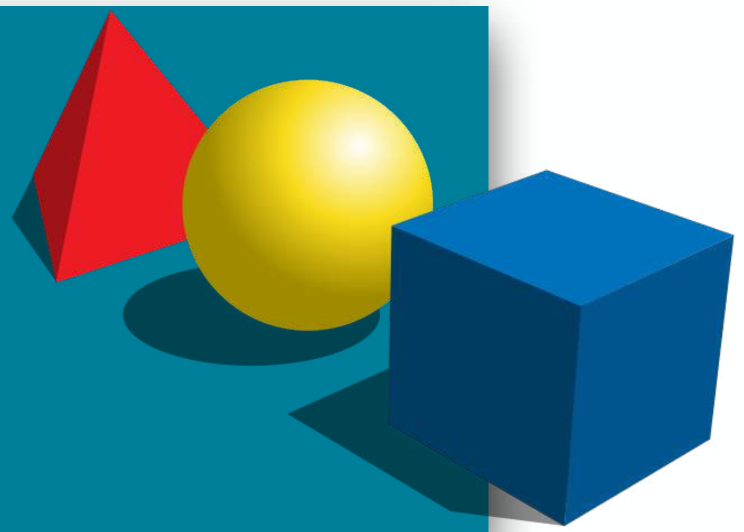




Global Prediction Stability Static Analysis

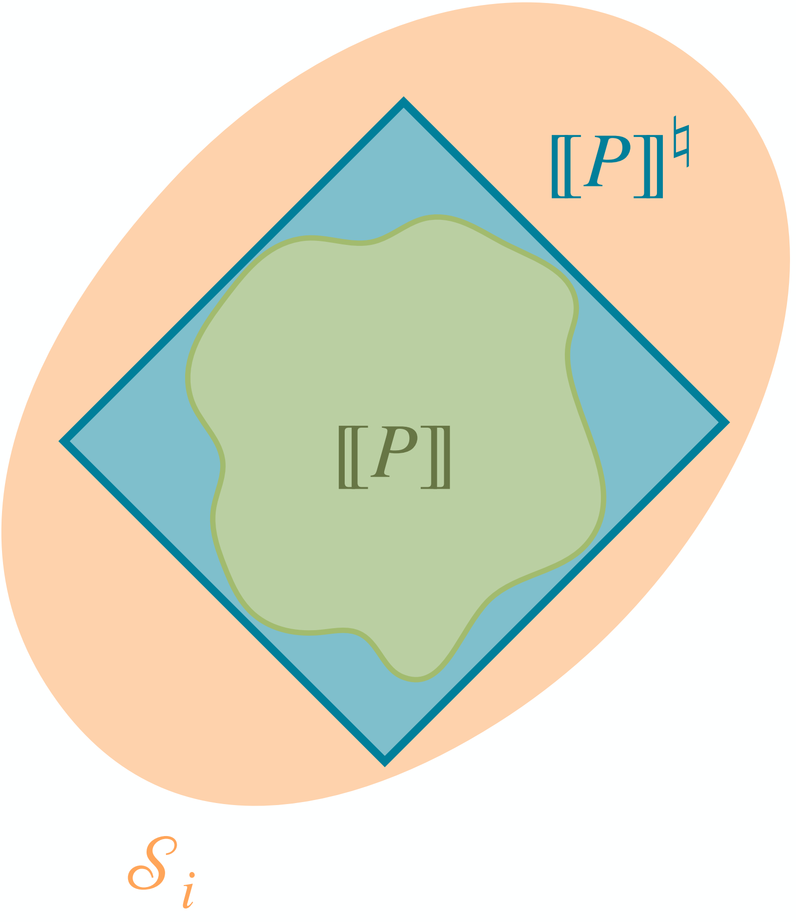
3-Step Recipe

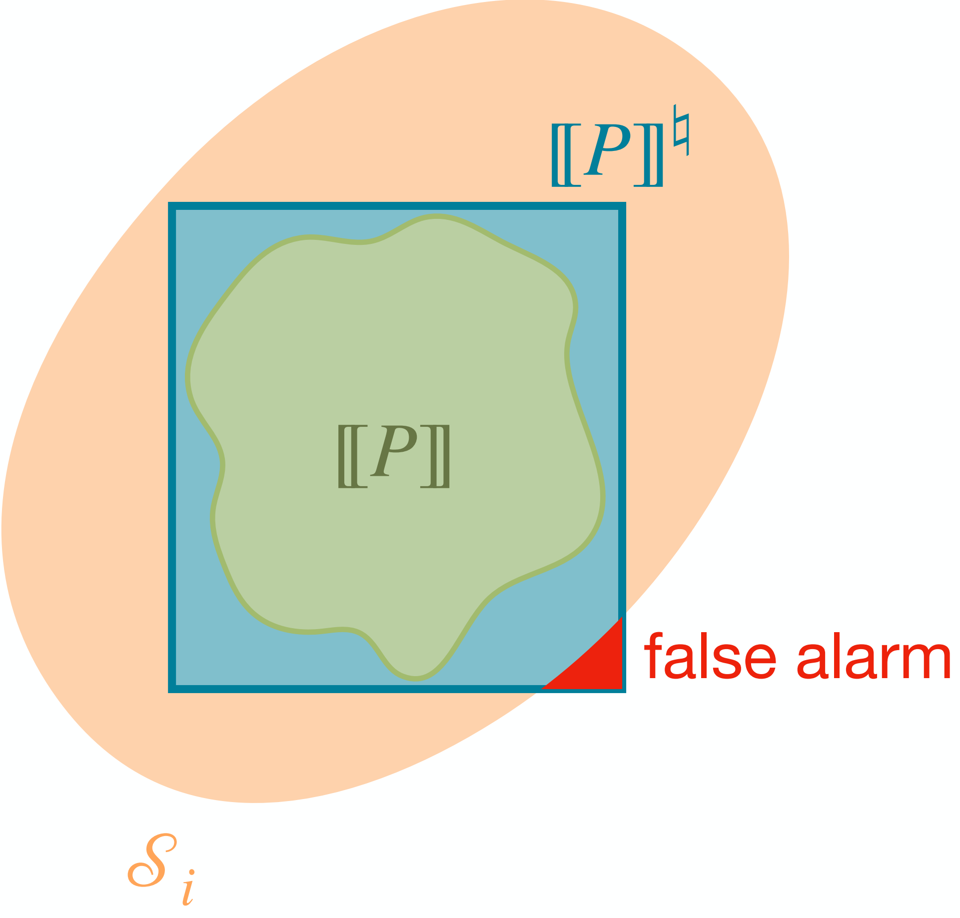
practical tools
targeting specific programs



abstract semantics, abstract domains
algorithmic approaches to decide program properties

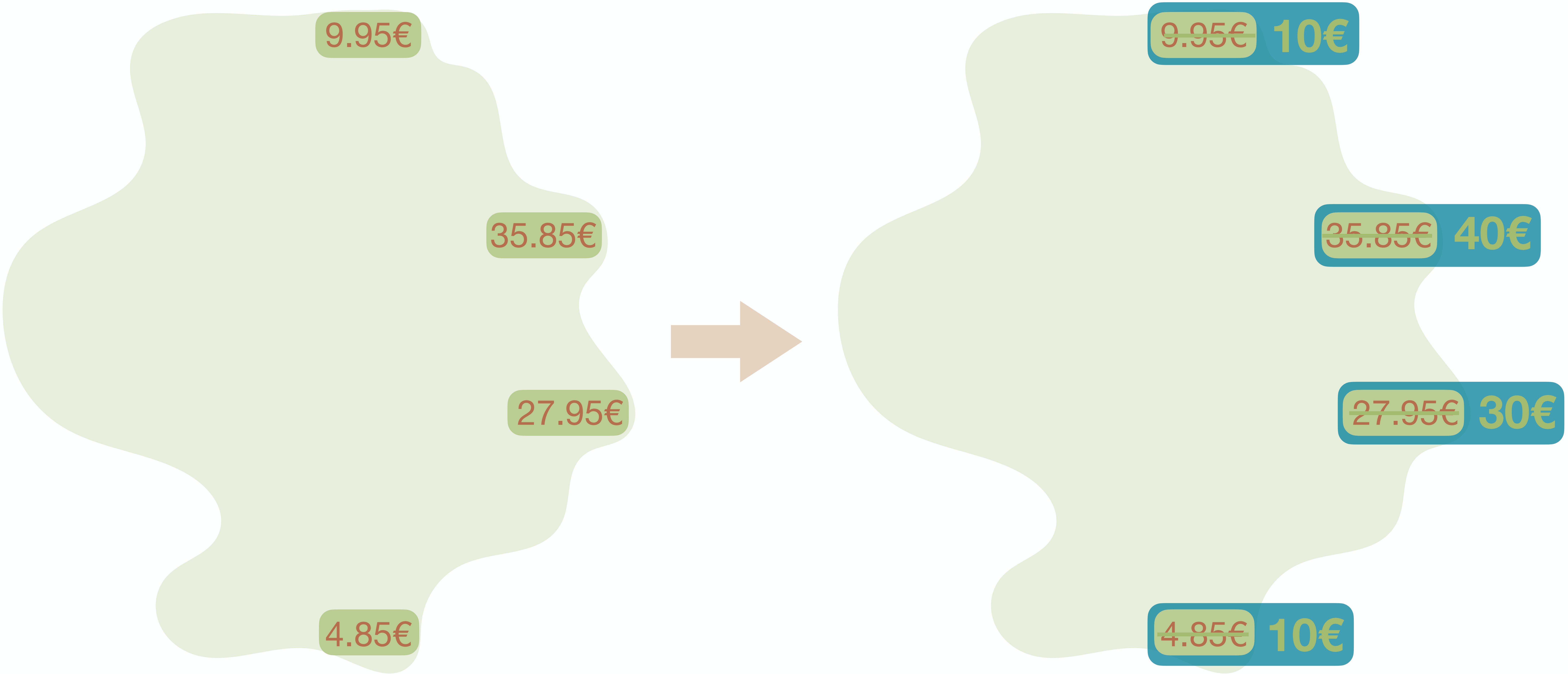
concrete semantics
mathematical models of the program behavior





Abstract Semantics

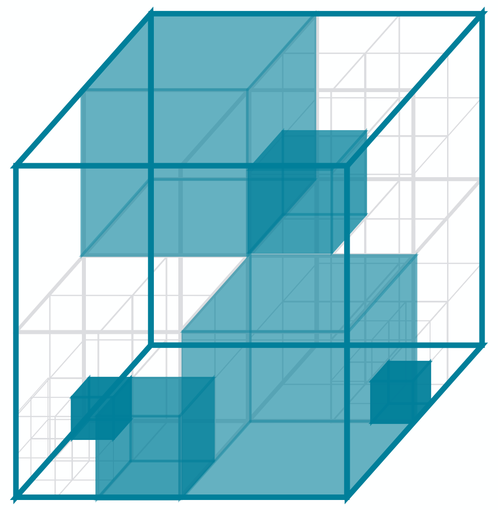
Intuition



Global Prediction Stability [OOPSLA 2020]

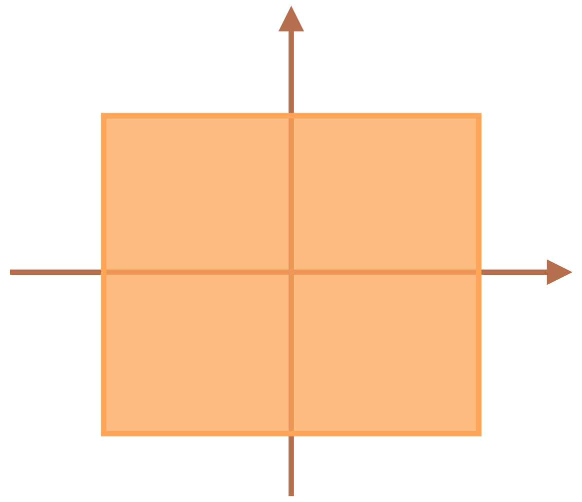
Static Forward Analysis

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



① **iteratively** partition the input space

② proceed **forwards in parallel** from all partitions

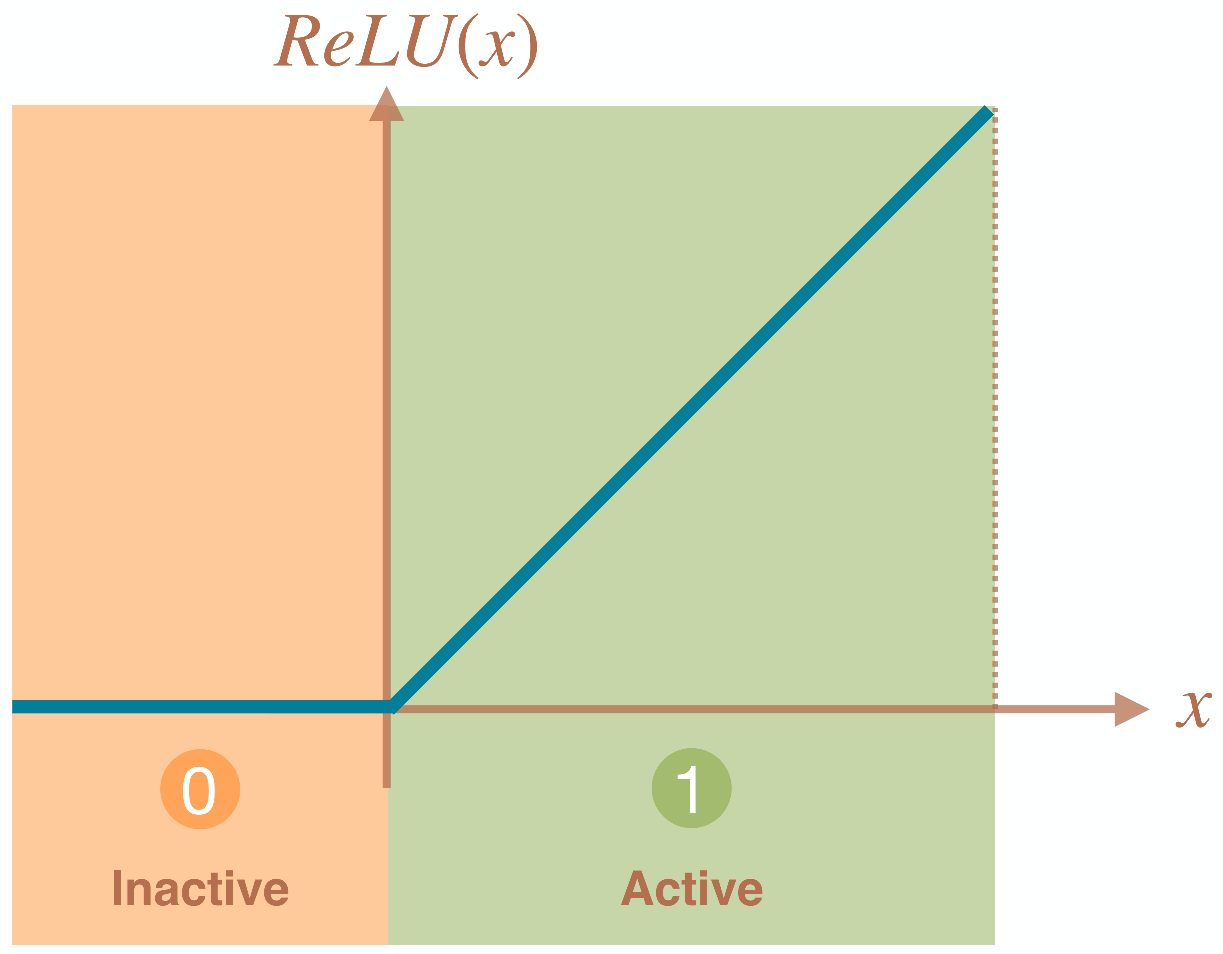


③ check output:
- **unique prediction** → ✔

④ group other partitions by **activation pattern**

Global Prediction Stability

Static Forward Analysis




```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.09[illegible]62)*x02 + (2.03[illegible]51)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.09[illegible]01)*x02 + (2.12[illegible]38)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.34[illegible]43)*x02 + (2.61[illegible]76)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.72[illegible]16)*x12 + (-3.4[illegible]653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.24[illegible]51)*x12 + (-3.8[illegible]811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.66[illegible]05)*x12 + (-4.2[illegible]086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.19[illegible]44)*x22 + (-2.6[illegible]086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.42[illegible]07)*x22 + (-3.6[illegible]113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.66[illegible]07)*x22 + (-4.2[illegible]974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.94[illegible]60)*x32 + (-4.0[illegible]463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0[illegible]95998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.[illegible]5429)*x32 + (5.[illegible]4773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```

Partition 1	Partition 2
x00: [0, 1]	x00: [0, 1]
x01: [-1, 0]	x01: [0, 1]
x02: T	x02: T
x03: [0.5, 1]	x03: [0.5, 1]
x04: [0, 1]	x04: [0, 1]
x05: [-1, 0]	x05: [-1, 0]

Activation patterns (both partitions): x10: 1, x11: 1, x12: ?; x20: 1, x21: 1, x22: 1; x30: 1, x31: 1, x32: 1; x40: 1, x41: 0, x42: ?

x50: …	x50: …
x51: …	x51: …

several partitions share the same activation pattern

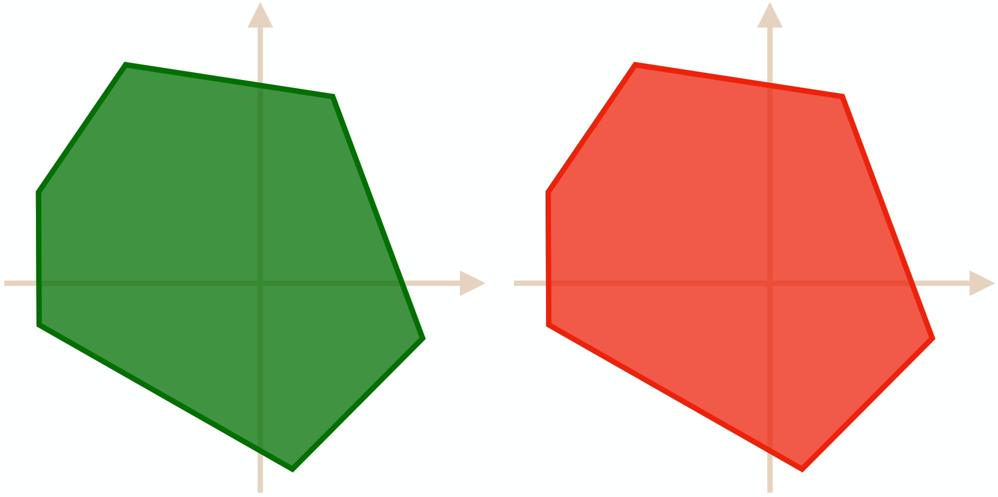
Global Prediction Stability [OOPSLA 2020]

Static Backward Analysis

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```

② proceed **backwards** in parallel **for each activation pattern**

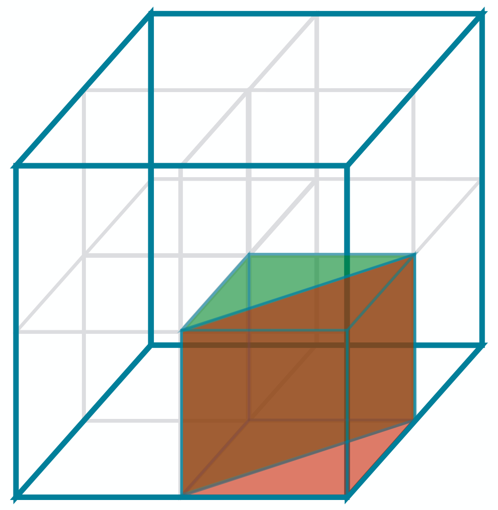
① start from an **abstraction** for each possible prediction outcome



Global Prediction Stability [OOPSLA 2020]

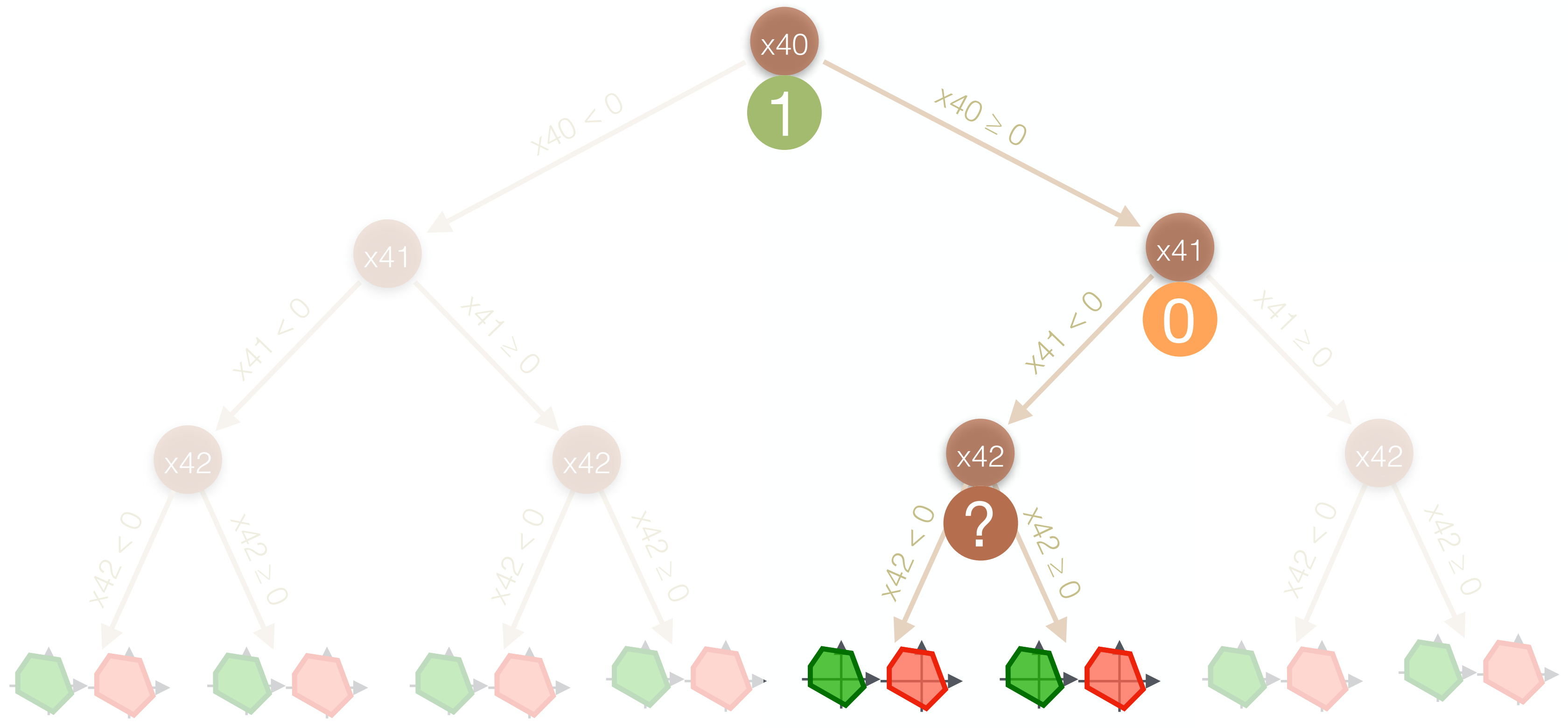
Static Backward Analysis

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```



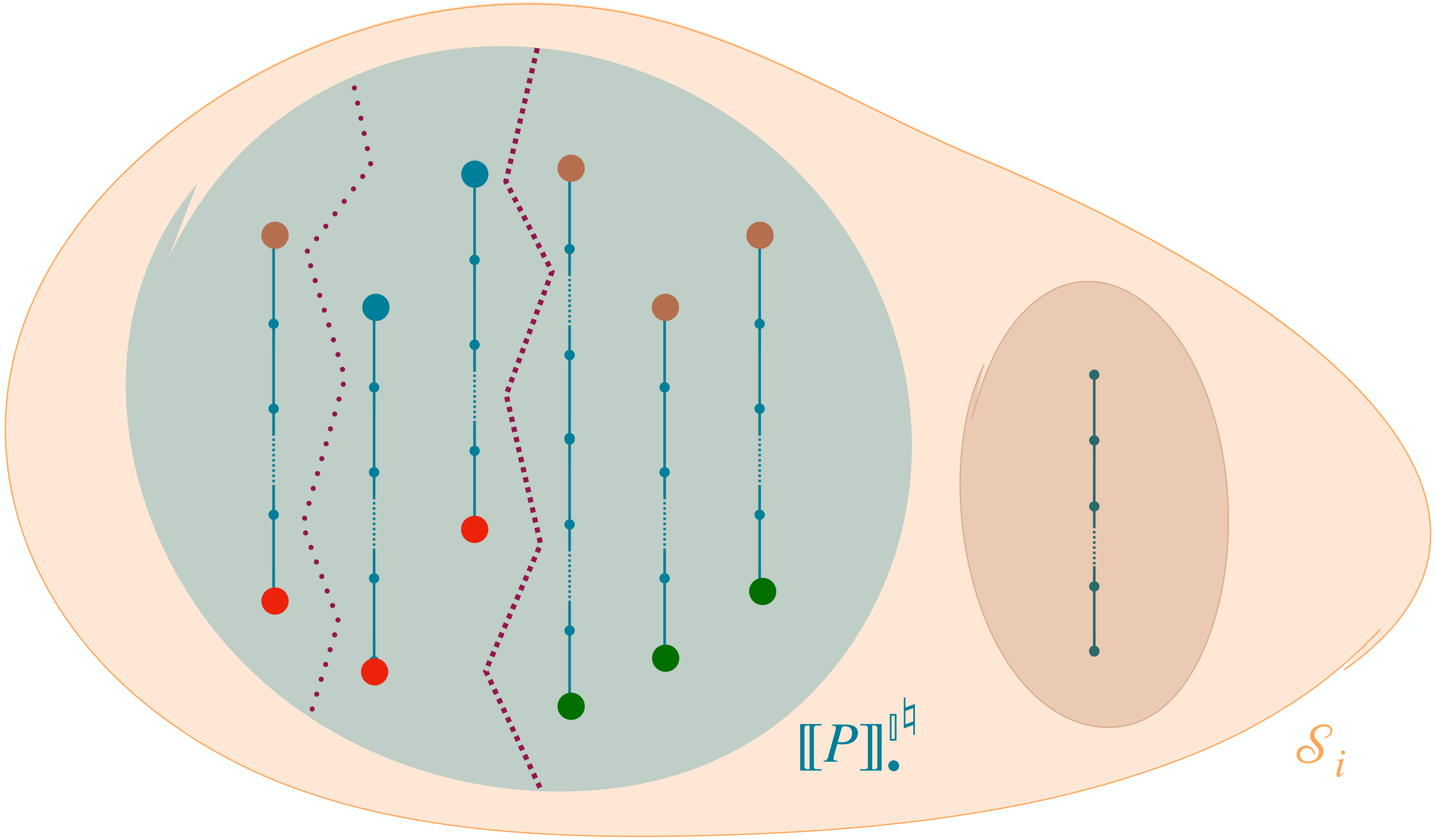
① check for **disjunction** in corresponding **input partitions**:
disjoint → ✔
otherwise → 🚨

```
x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

1 x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
0 x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
? x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



Global Prediction Stability Verification

$P \models \mathcal{S}_i \Leftarrow \forall I \in \mathbb{I}\colon \forall A, B \in \underbrace{[\![P]\!]^{\natural}_{\bullet}}_{\textbf{Abstract Semantics}}\colon A^I_\omega \neq B^I_\omega \Rightarrow A^I_0 \not\equiv_{\setminus i} B^I_0$



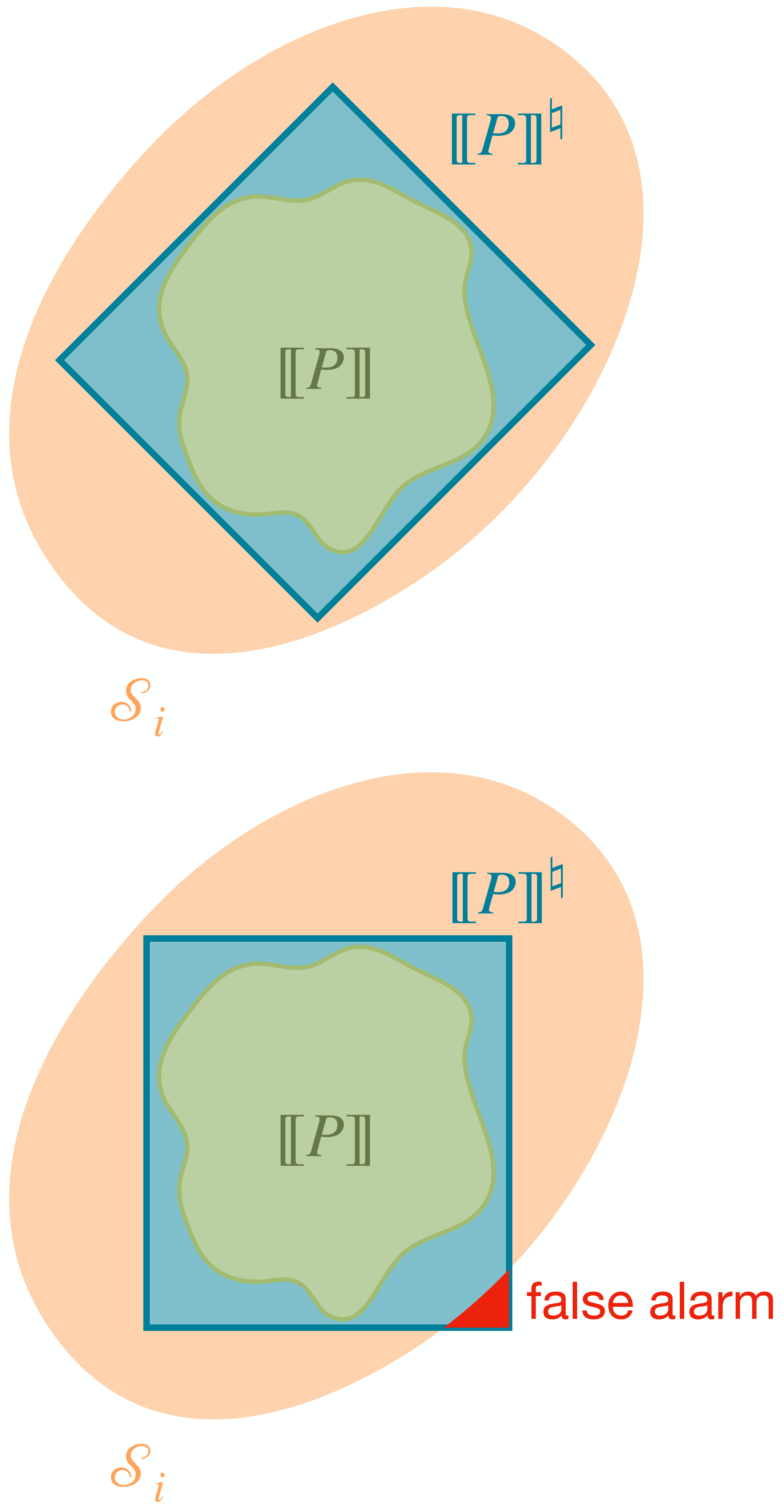
Global Prediction Stability Static Analysis

3-Step Recipe

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties

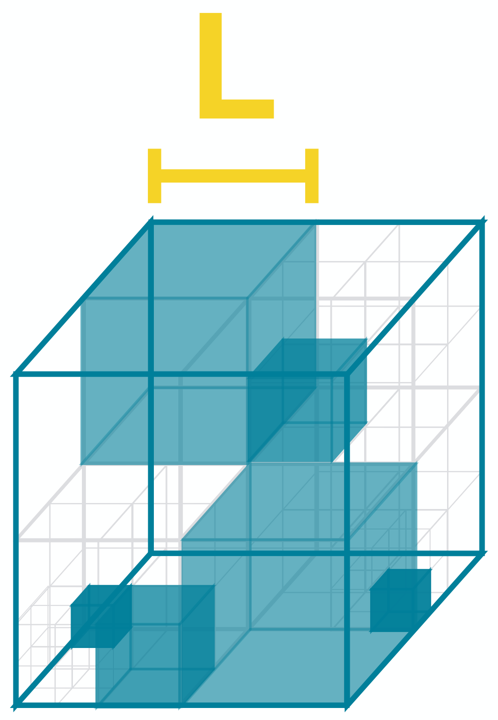
concrete semantics
mathematical models of the program behavior



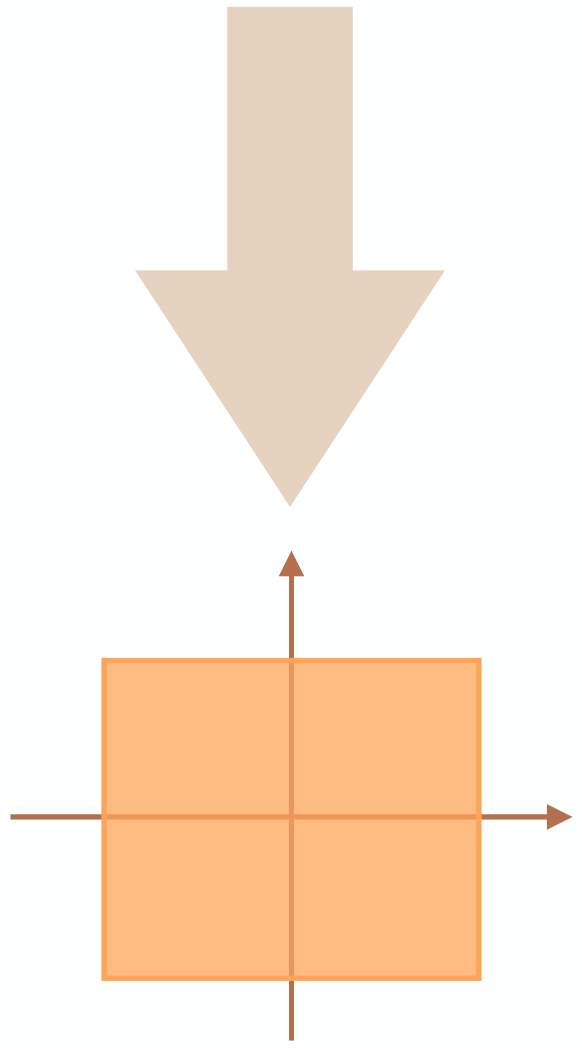
Global Prediction Stability [OOPSLA 2020]

Static Forward Analysis

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())

1 x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
1 x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
? x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

? x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
? x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
? x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

? x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
1 x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
0 x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

1 x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
0 x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
0 x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



① **iteratively** partition the input space



② proceed **forwards in parallel** from all partitions

③ check output:
- **unique prediction** → ✔

∪

④ group other partitions by **activation pattern**

Scalability-vs-Precision Tradeoff

Analyzed Input Space Percentage

L	U	Intervals
1	2	46,9 %
	6	46,9 %
0.5	2	76,9 %
	6	84,4 %

Execution Time

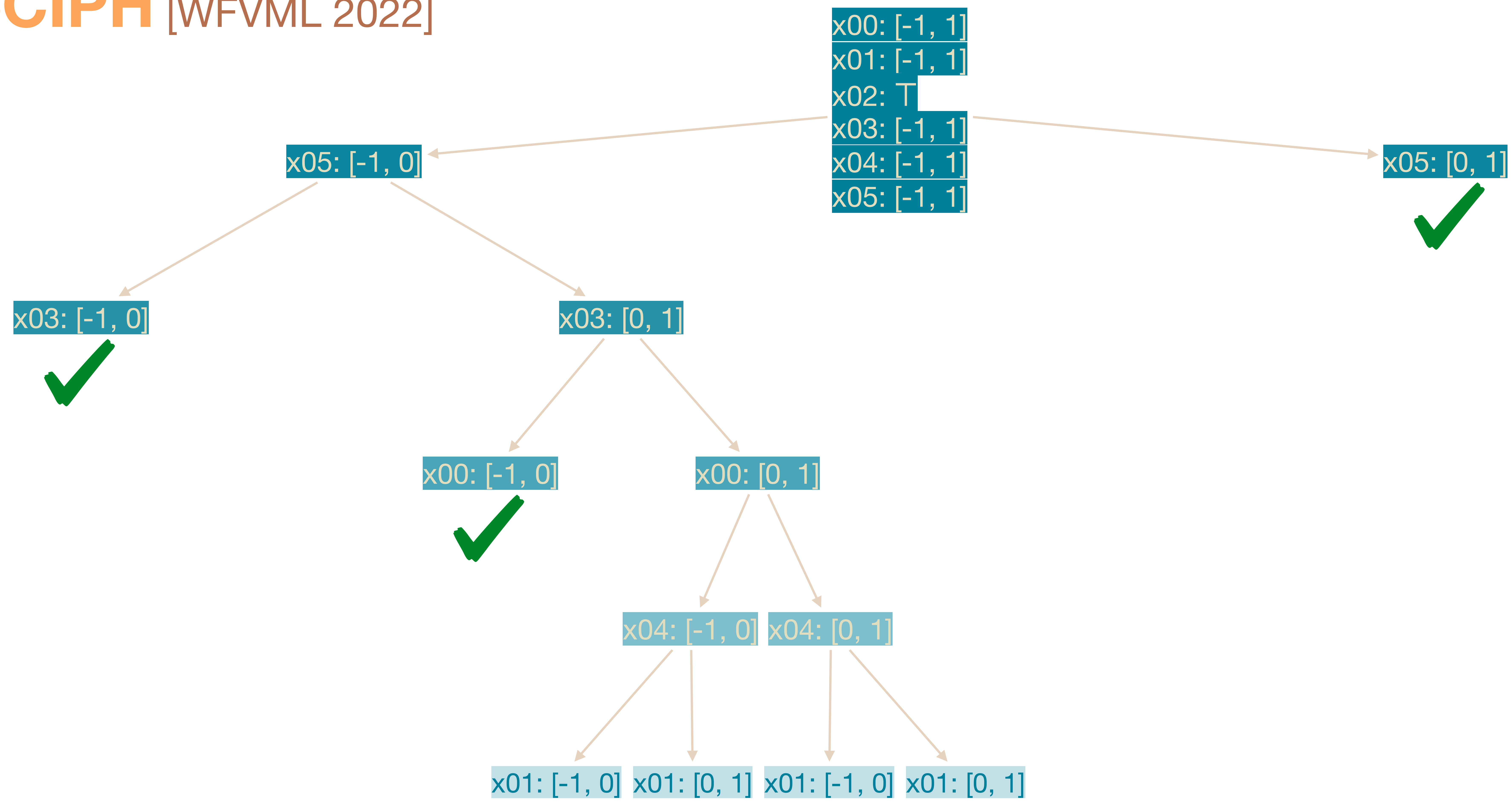
L	U	Intervals
1	2	0,08s
	6	0,16s
0.5	2	8,88s
	6	64,67s

Partitioning Strategy

Input Range

Partitioning Strategy

ReCIPH [WFVML 2022]



Scalability-vs-Precision Tradeoff

Analyzed Input Space Percentage

L	U	Intervals	Product [SAS 2021]	
			Input Range Partitioning	ReCIPH [WFVML 2022]
1	2	46,9 %	90,6 %	90,6 %
	6	46,9 %	90,6 %	90,6 %
0.5	2	76,9 %	100,0 %	100,0 %
	6	84,4 %	100,0 %	100,0 %

Execution Time

L	U	Intervals	Product [SAS 2021]	
			Input Range Partitioning	ReCIPH [WFVML 2022]
1	2	0,08s	0,26s	0,12s
	6	0,16s	0,35s	0,20s
0.5	2	8,88s	2,10s	1,61s
	6	64,67s	2,10s	1,62s

Scalability wrt Considered Input Space

Global Prediction Stability (100% of the Input Space)

	Symbolic	
ReLUs	Analyzed Input Space	Time
80	61.3 %	10h 25m 2s
320	24.2 %	9h 41m 36s
1280	0 %	> 13h

Local Prediction Stability (1% of the Input Space)

	Symbolic	
ReLUs	Analyzed Input Space	Time
80	1 %	3m 41s
320	1 %	21m 9s
1280	1 %	3h 31m 45s

Global Prediction Stability

[OOPSLA 2020, SAS 2021, WFVML 2022]

Liveness Non-Exploitability

Termination Resilience

Data Leakage

[TASE 2024, SCP 2025]

Input Data Usage

USED_i

[ESOP 2018]

Quantitative Data Usage

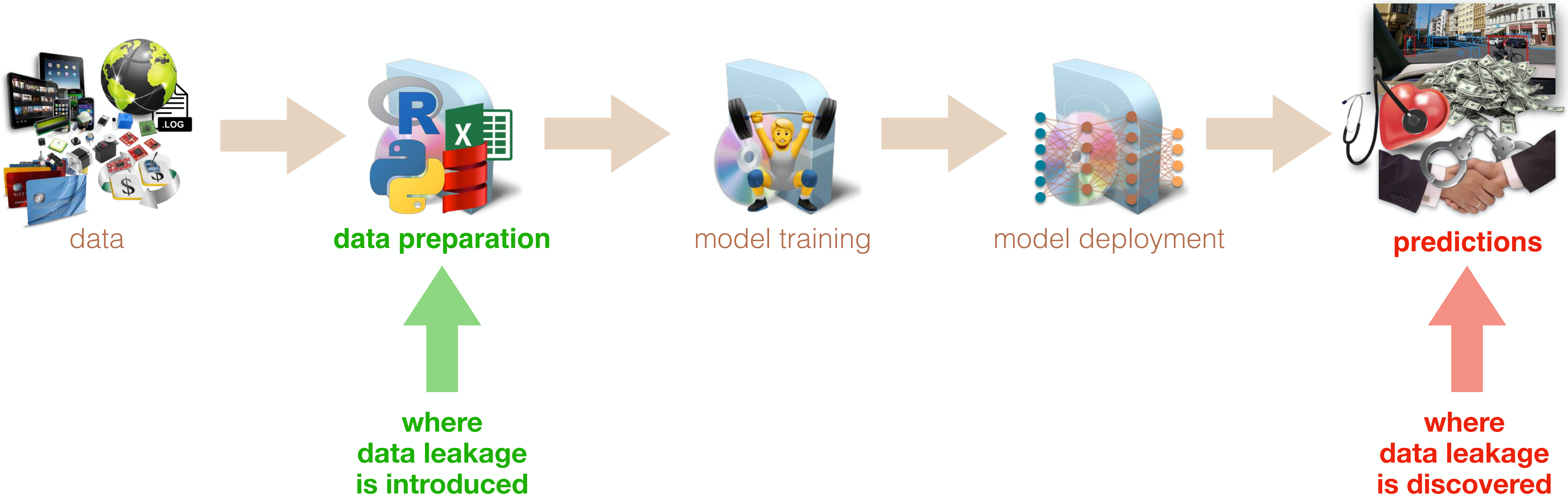
[NFM 2024, SAS 2024]

[SAS 2025]

Partial Abstract Non-Interference

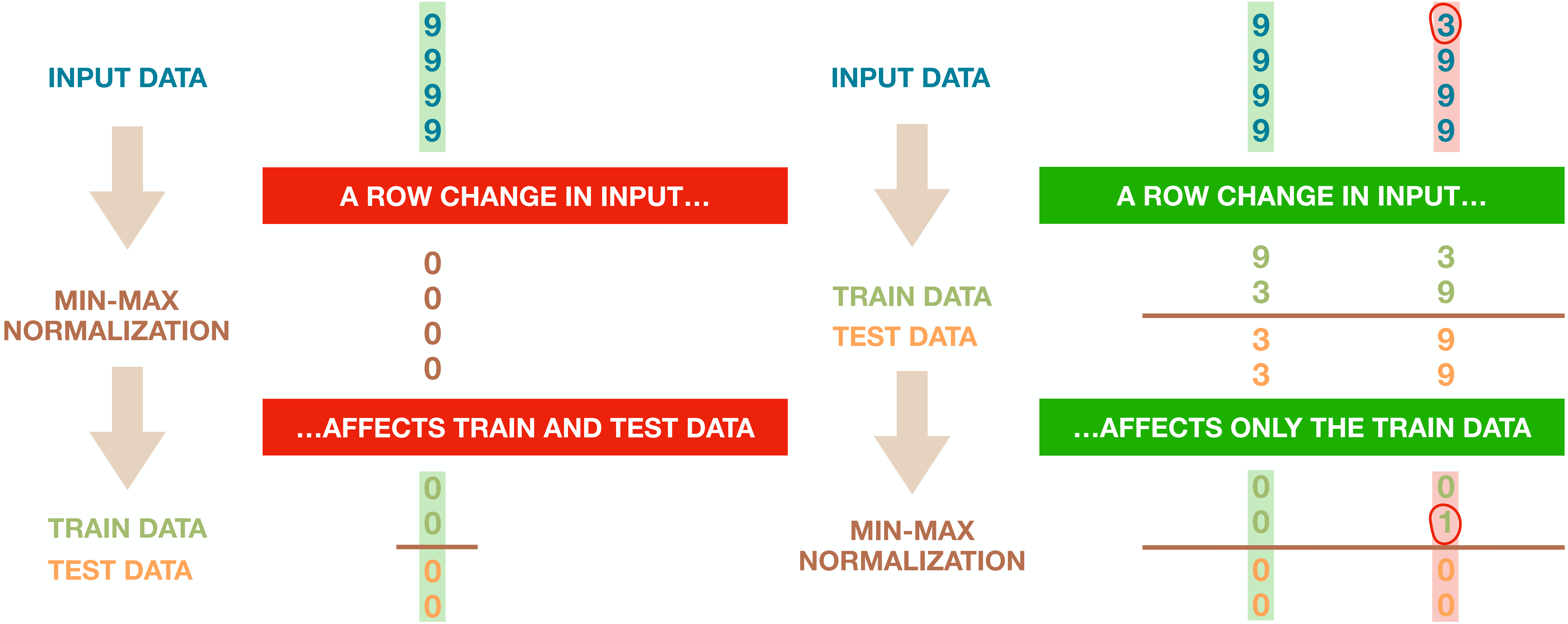
Partial Completeness

Machine Learning Pipeline

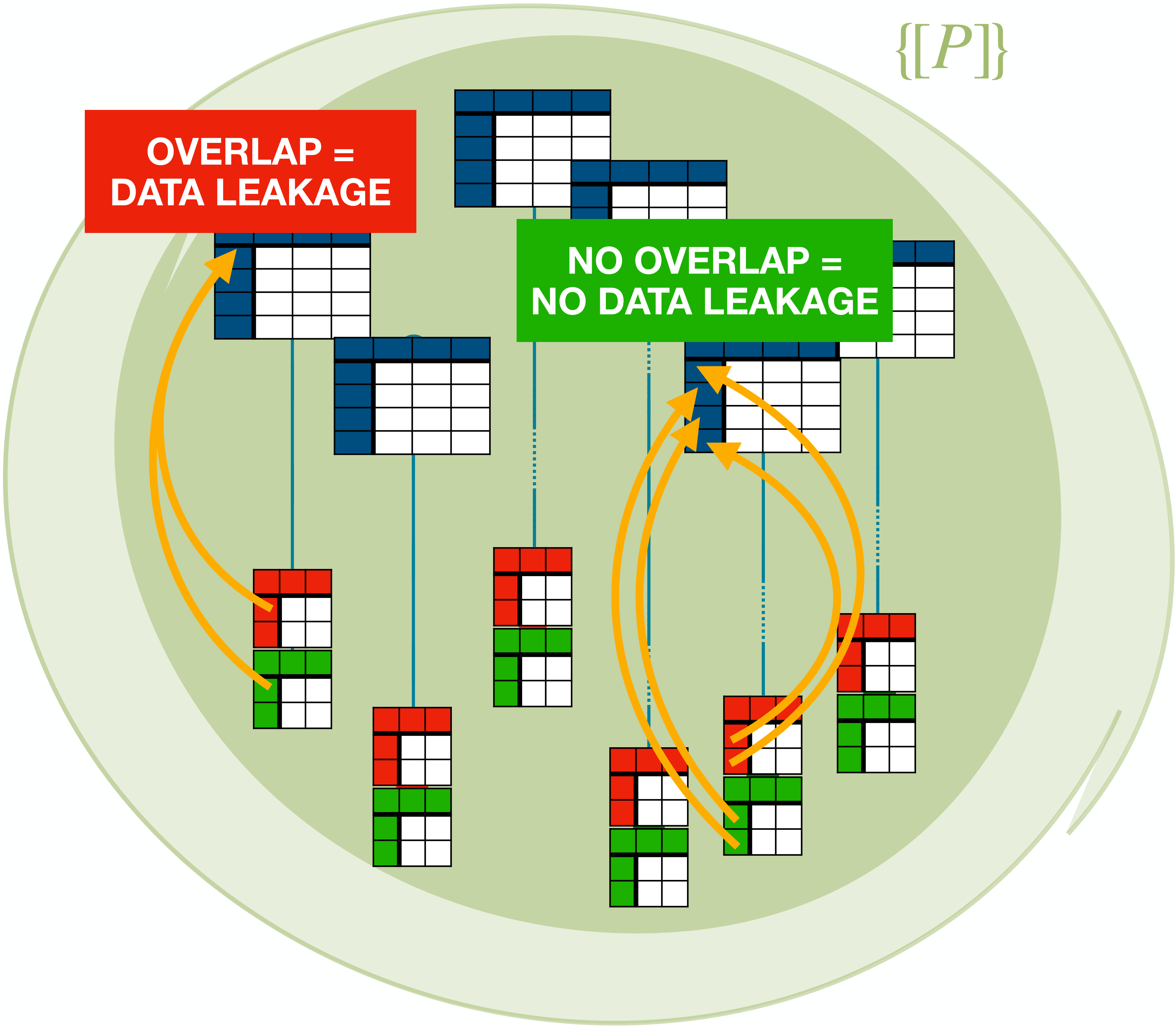


(Absence of) Data Leakage

Independence of Training and Testing Data



Hierarchy of Semantics [TASE 2024]



$\{[P]\}$ Data Leakage Semantics

$\uparrow \alpha^{!}$

$[\![P]\!]_{\rightsquigarrow}$ Dependency Semantics

$\uparrow \alpha_{\rightsquigarrow}$

$\{[\![P]\!]\}$ Collecting Semantics

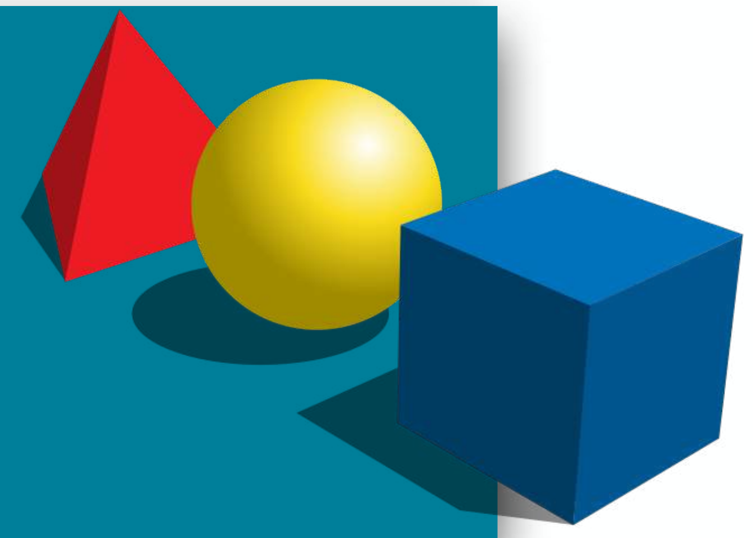
Data Leakage Static Analysis

4-Step Recipe

practical tools
targeting specific programs

practical tools
targeting specific programs

abstract semantics, abstract domains
algorithmic approaches to decide program properties



concrete semantics
mathematical models of the program behavior

Global Prediction Stability

[OOPSLA 2020, SAS 2021, WFVML 2022]

Liveness Non-Exploitability

↳ **Termination Resilience**

Data Leakage

[TASE 2024, SCP 2025]

Input Data Usage

USED_i

[ESOP 2018]

[SAS 2025]

Partial Abstract Non-Interference

↳ **Partial Completeness**

Quantitative Data Usage

[NFM 2024, SAS 2024]

Quantitative Data Usage [SAS 2024]

S2N-Bignum is Timing Side-Channel Free

Program	Input Variables Δ		Maybe Dangerous	Zero Impact
	Safe $\Delta\|_S$	Numerical $\Delta\|_N$		
Add	s_1, s_3, s_5	n_2, n_4, n_6	s_1	s_3, s_5, n_2, n_4, n_6
Amontifier	s_1	n_2, n_3, n_4	s_1	n_2, n_3, n_4
Amontmul	s_1	n_2, n_3, n_4, n_5	s_1	n_2, n_3, n_4, n_5
Amontredc	s_1, s_3, s_6	n_2, n_4, n_5	s_1, s_3, s_6	n_2, n_4, n_5
Amontsqr	s_1	n_2, n_3, n_4	s_1	n_2, n_3, n_4
Bitfield	s_1	n_2, n_3, n_4, n_5	s_1	n_2, n_3, n_4, n_5
Bitsize	s_1	n_2	s_1	n_2
Cdiv	s_1, s_3	n_2, n_4, n_5	s_1, s_3	n_2, n_4, n_5
Cdiv_exact	s_1, s_3	n_2, n_4, n_5	s_1	n_2, s_3, n_4, n_5
Cld	s_1	n_2	s_1	n_2
Clz	s_1	n_2	s_1	n_2
Cmadd	s_1, s_4	n_2, n_3, n_5	s_1, s_4	n_2, n_3, n_5
Cmnegadd	s_1, s_4	n_2, n_3, n_5	s_1, s_4	n_2, n_3, n_5
Cmod	s_1	n_2, n_3	s_1	n_2, n_3
Cmul	s_1, s_4	n_2, n_3, n_5	s_1, s_4	n_2, n_3, n_5
Coprime	s_1, s_3	n_2, n_4, n_5	s_1, s_3	n_2, n_4, n_5
Copy	s_1, s_3	n_2, n_4	s_1, s_3	n_2, n_4
Copy_row_from_table	s_3, s_4	n_1, n_2, n_5	s_3, s_4	n_1, n_2, n_5
Copy_row_from_table_16_neon	s_3	n_1, n_2, n_4	s_3	n_1, n_2, n_4
Copy_row_from_table_32_neon	s_3	n_1, n_2, n_4	s_3	n_1, n_2, n_4
Copy_row_from_table_8n_neon	s_3, s_4	n_1, n_2, n_5	s_3, s_4	n_1, n_2, n_5
Ctd	s_1	n_2	s_1	n_2
Ctz	s_1	n_2	s_1	n_2
Demont	s_1	n_2, n_3, n_4	s_1	n_2, n_3, n_4
Digit	s_1	n_2, n_3	s_1	n_2, n_3
Digitsize	s_1	n_2	s_1	n_2
Divmod10	s_1	n_2	s_1	n_2
Emontredc	s_1	n_2, n_3, n_4	s_1	n_2, n_3, n_4
Eq	s_1, s_3	n_2, n_4	s_1, s_3	n_2, n_4
Even	s_1	n_2		s_1, n_2
Ge	s_1, s_3	n_2, n_4	s_1, s_3	n_2, n_4
Gt	s_1, s_3	n_2, n_4	s_1, s_3	n_2, n_4
Iszero	s_1	n_2	s_1	n_2
Le	s_1, s_3	n_2, n_4	s_1, s_3	n_2, n_4
Lt	s_1, s_3	n_2, n_4	s_1, s_3	n_2, n_4
Madd	s_1, s_3, s_5	n_2, n_4, n_6	s_1, s_3, s_5	n_2, n_4, n_6

Global Prediction Stability

[OOPSLA 2020, SAS 2021, WFVML 2022]

Liveness Non-Exploitability

Termination Resilience

Data Leakage

[TASE 2024, SCP 2025]

Input Data Usage

USED_i

[ESOP 2018]

[SAS 2025]

Partial Abstract Non-Interference

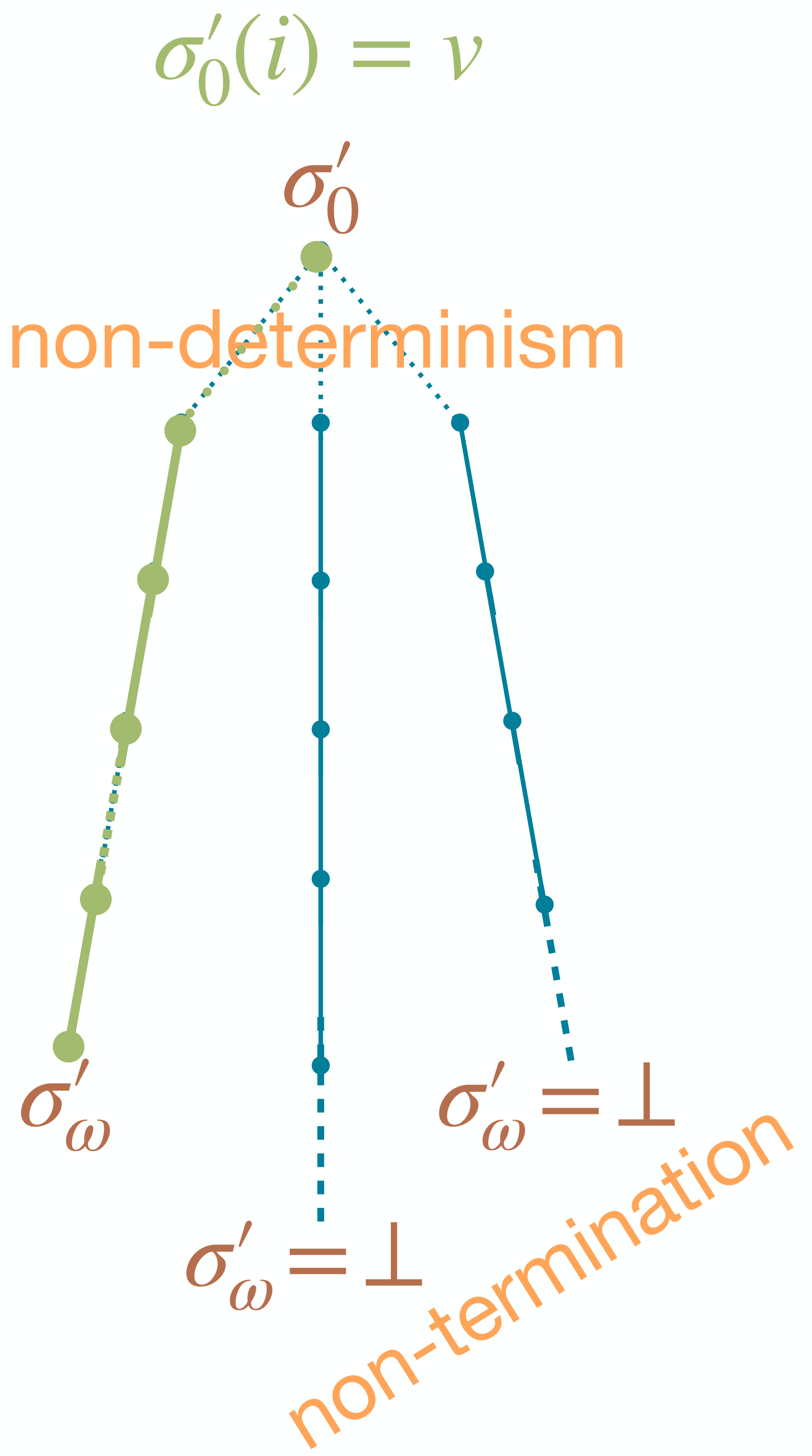
Partial Completeness

Quantitative Data Usage

[NFM 2024, SAS 2024]

Termination Resilience

Termination is Possible for All Untrusted Input Values



angelic non-determinism

$$\neg \mathsf{USED}_i \stackrel{\text{def}}{=} \forall v : \exists \sigma' : A_2 \wedge \neg C$$

$$A_2 \stackrel{\text{def}}{=} \sigma'_0(i) = v$$

$$\neg C \stackrel{\text{def}}{=} \sigma'_\omega \neq \bot$$

$$\mathcal{TR} \stackrel{\text{def}}{=} \{ [\![P]\!] \mid \forall i : \neg \mathsf{USED}_i([\![P]\!]) \}$$

Termination Resilience

Triage of Non-Termination Alarms

Property	Verified	Alarms
Termination	0	278
Termination Resilience	180	98

⟵ 35%

Global Prediction Stability

[OOPSLA 2020, SAS 2021, WFVML 2022]

Liveness Non-Exploitability

└···· Termination Resilience

Data Leakage

[TASE 2024, SCP 2025]

Input Data Usage

USED_i

[ESOP 2018]

Quantitative Data Usage

[NFM 2024, SAS 2024]

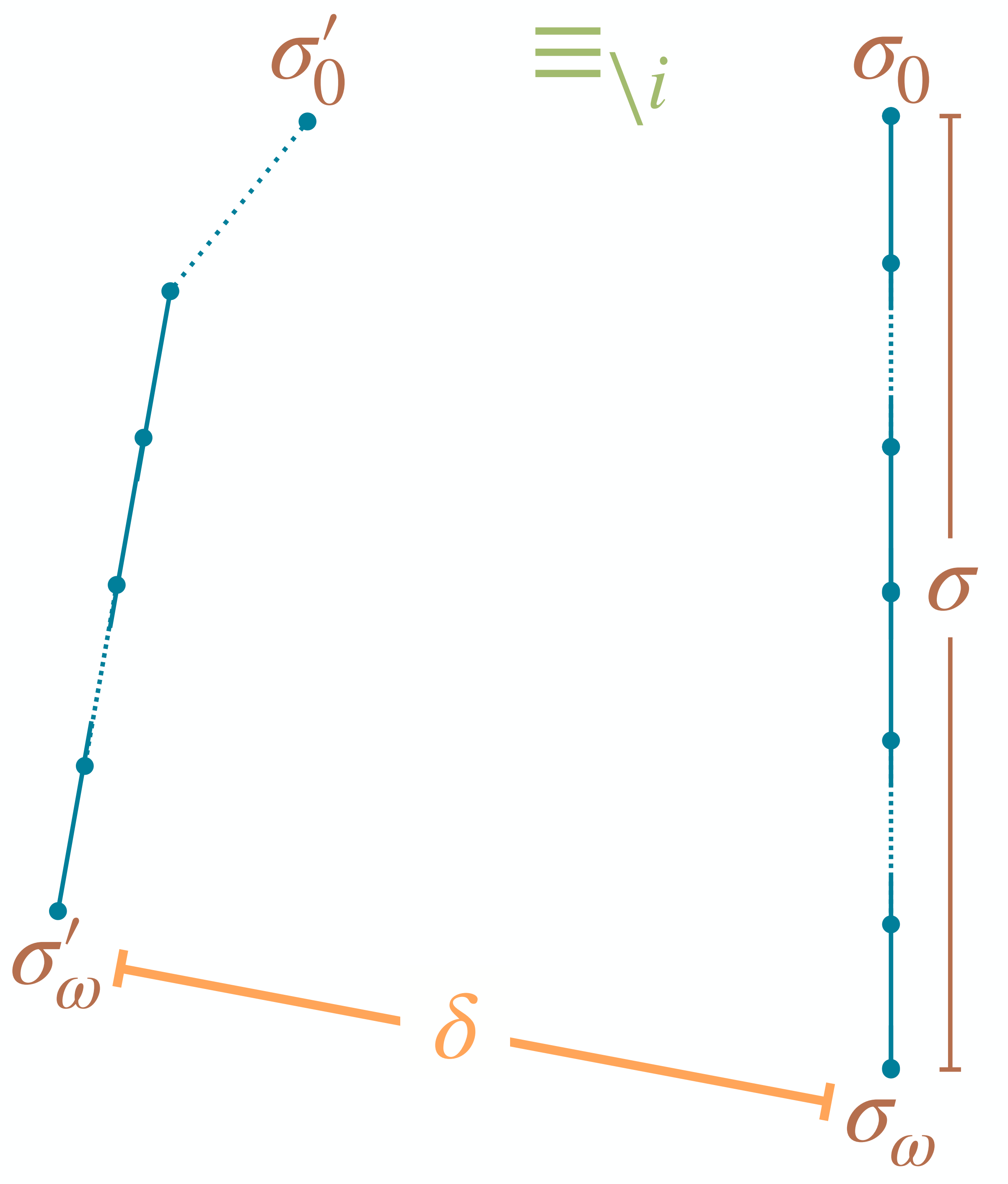
[SAS 2025]

Partial Abstract Non-Interference

└···· Partial Completeness

Partial Abstract Non-Interference

Outcome is Limitedly Affected by Perturbations to Certain Inputs



$$\neg \mathsf{USED}_i \stackrel{\text{def}}{=} \forall \sigma\sigma' \colon B \Rightarrow \neg C$$

$$B \stackrel{\text{def}}{=} \sigma_0 \equiv_{\setminus i} \sigma_0'$$

$$\neg C \stackrel{\text{def}}{=} \delta(\sigma_\omega, \sigma_\omega') \leq \epsilon$$

Partial Abstract Non-Interference [SAS 2025]

Bounded Behavior Variations for Inputs Sharing a Similar Property

$$\epsilon\text{-PartialANI} \stackrel{\text{def}}{=} \forall x y \colon B \Rightarrow \neg C$$

$$B \stackrel{\text{def}}{=} \delta_B(\eta(x), \eta(y)) = 0$$

$$\neg C \stackrel{\text{def}}{=} \delta_C(\rho(f(x)), \rho(f(y))) \leq \epsilon$$

Partial Abstract Non-Interference [SAS 2025]

On the Relation With Partial Completeness

$$\epsilon\text{-PartialANI} \stackrel{\text{def}}{=} \forall xy\colon \delta_B(\eta(x), \eta(y)) = 0 \Rightarrow \delta_C(\rho(f(x)), \rho(f(y))) \leq \epsilon$$

$$\Downarrow$$

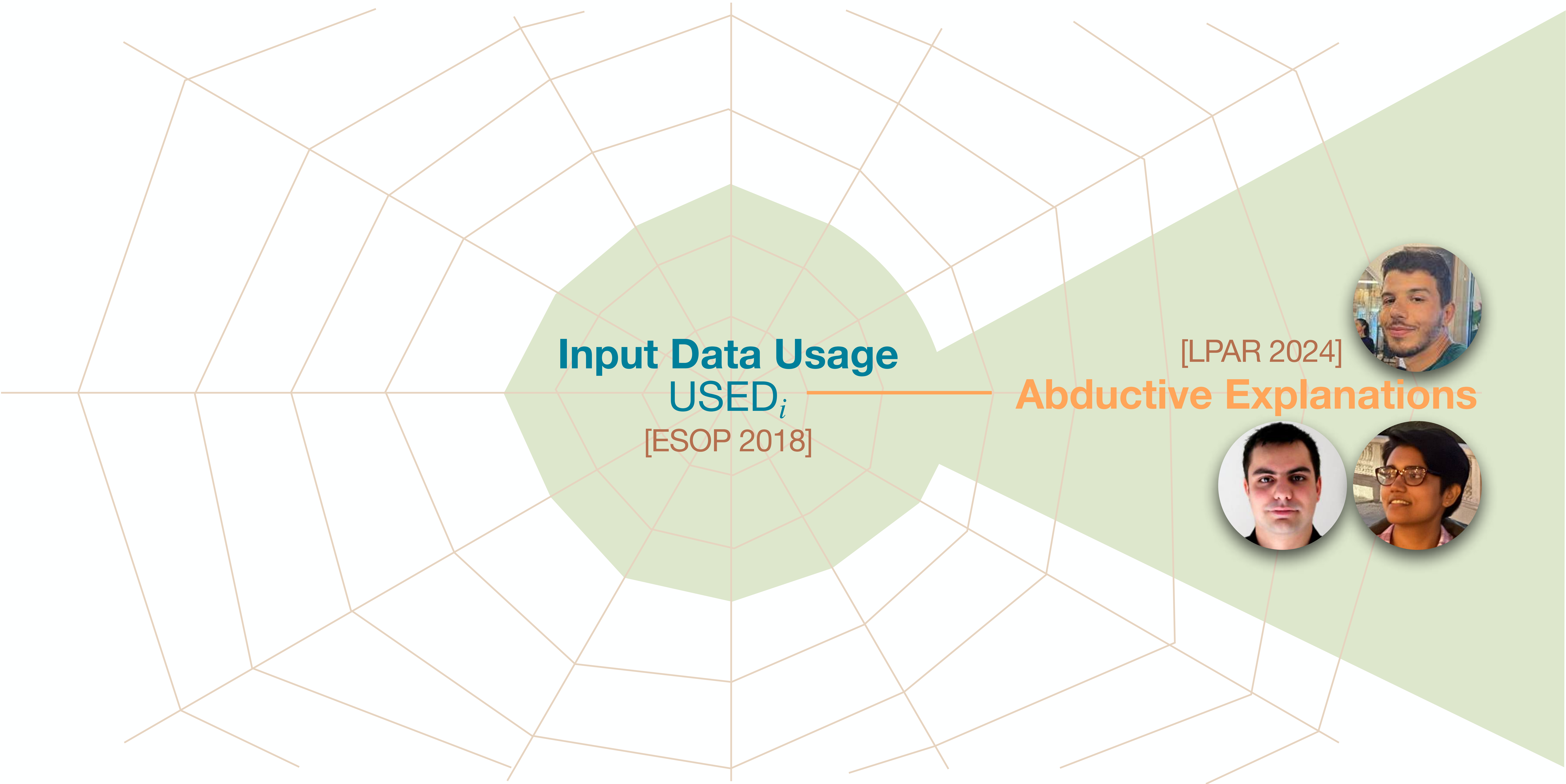
$$\epsilon\text{-PartialCompleteness} \stackrel{\text{def}}{=} \forall x\colon \delta_C(\rho(f(x)), \rho(f(\eta(x)))) \leq \epsilon$$

$$\Downarrow$$

$$2\epsilon\text{-PartialANI} \stackrel{\text{def}}{=} \forall xy\colon \delta_B(\eta(x), \eta(y)) = 0 \Rightarrow \delta_C(\rho(f(x)), \rho(f(y))) \leq 2\epsilon$$

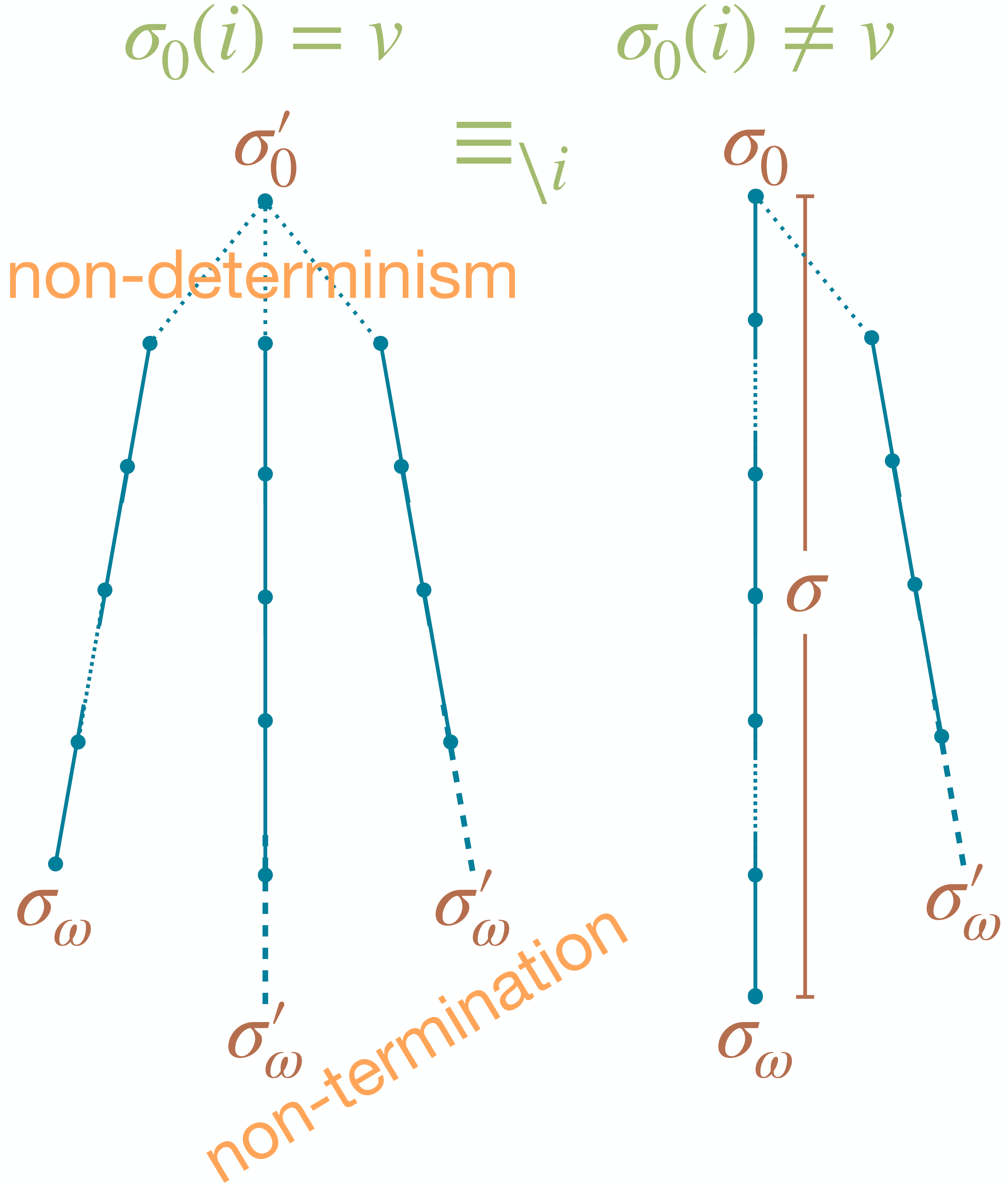
Verification

Explainability


Input Data Usage
USED$_i$
[ESOP 2018]
[LPAR 2024]
Abductive Explanations

Abductive Explanations (AXps)

Subset-Minimal Set of Inputs Sufficient for Determining Outcome



$$\neg\text{USED}_i \stackrel{\text{def}}{=} \forall \sigma v \colon A_1 \Rightarrow \exists \sigma' \colon A_2 \land B \land \neg C$$

$$A_1 \stackrel{\text{def}}{=} \sigma_0(i) \neq v$$

$$A_2 \stackrel{\text{def}}{=} \sigma'_0(i) = v$$

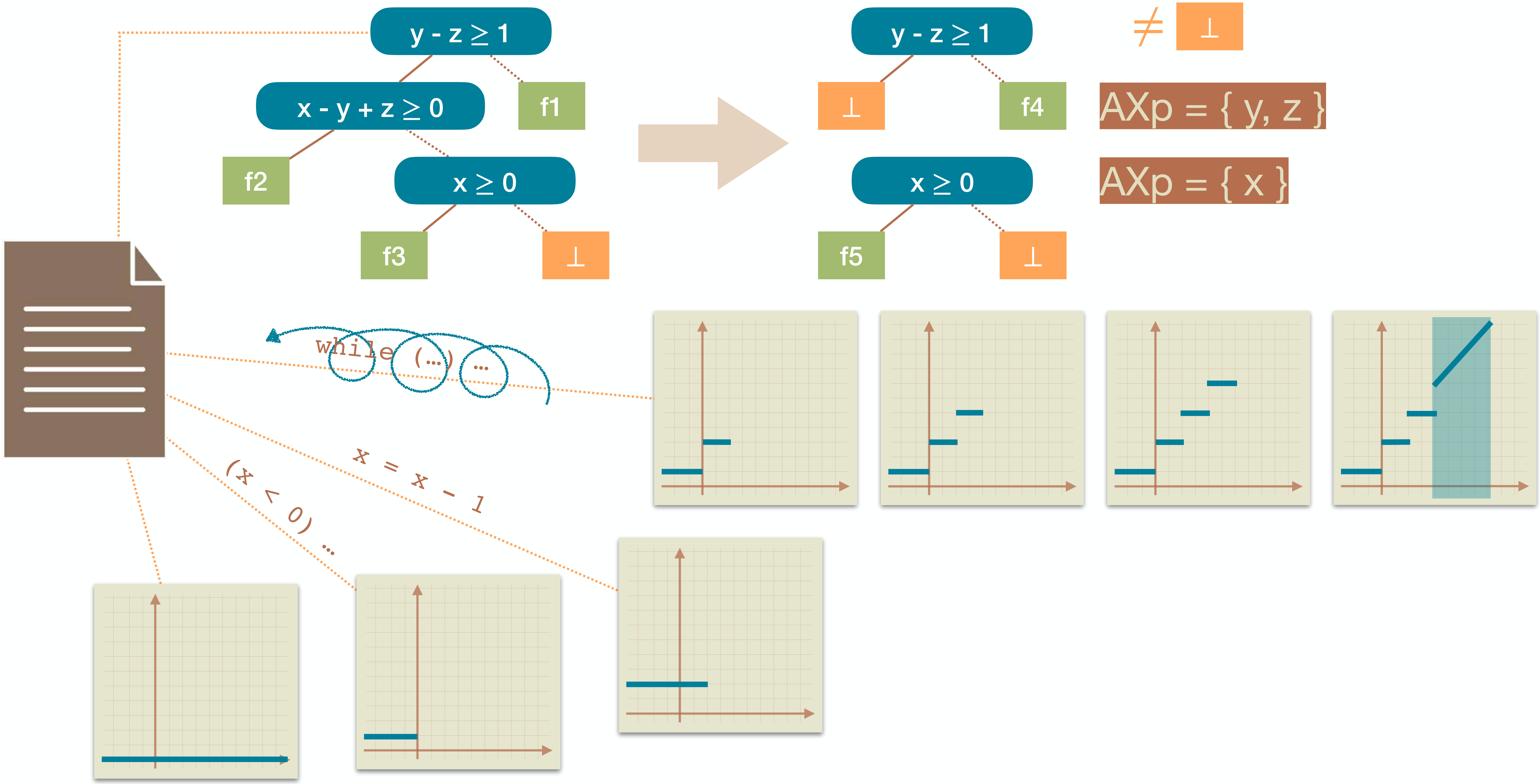
$$B \stackrel{\text{def}}{=} \sigma_0 \equiv_{\setminus i} \sigma'_0$$

$$\neg C \stackrel{\text{def}}{=} \sigma_\omega = \sigma'_\omega$$

$$\text{AXp} \stackrel{\text{def}}{=} \min_{\subseteq} \left\{ X \mid \forall i \in \mathbb{X} \setminus X \colon \neg\text{USED}_i([\![P]\!]) \right\}$$

AXps for Termination [LPAR 2024]

Drop (i.e., Havoc) Variables While AXp Condition Holds



AXps for Neural Network Predictions

Drop (i.e., Havoc) Inputs While AXp Condition Holds

= x51

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```

X:
x00: [-1, 1]
x01: [-1, 1]
x02: -1
x03: [-1, 1]
x04: [-1, 1]
x05: [-1, 1]

```
x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + ... 65)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + ... 32)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + ... 5)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + ...
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + ...
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + ...

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + ...
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + ...
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + ...

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + ...
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + ...
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + ...

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```

{ x00, x01, x02, x03, x04, x05 } → x51

Drop x00: { x01, x02, x03, x04, x05 } → x51

Drop x01: { x02, x03, x04, x05 } → x51

Drop x02: { x03, x04, x05 } → ✗

Drop x03: { x02, x04, x05 } → x51

Drop x04: { x02, x05 } → x51

Drop x05: { x02 } → x51

⇒ AXp = { x02 }

(Weak) AXps for Neural Network Predictions

Drop (i.e., Havoc) Inputs While AXp Condition Holds

= x51

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```

x: x00: 1, x01: 1, x02: -1, x03: 1, x04: 1, x05: -1

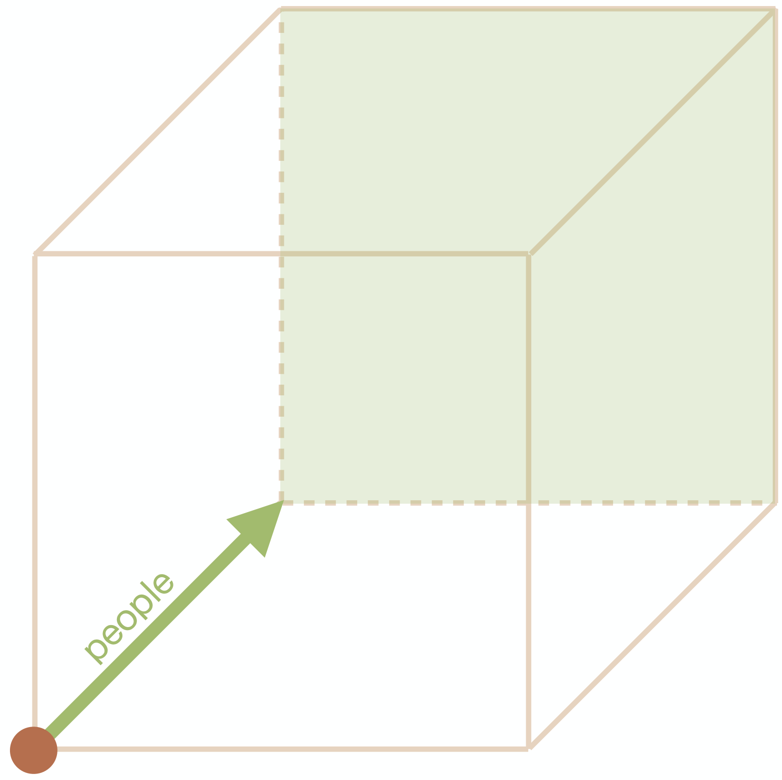
INTERVALS
wAXp = { x02, x03, x05 }

SYMBOLIC
wAXp = { x00, x02, x03 }
wAXp = { x02, x03, x05 }

DEEPPOLY
wAXp = { x02, x03 }
wAXp = { x02, x05 }
= PRODUCT

```
x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```

Research Agenda



- **relational explanations**

Abductive ReLU Explanations

Subset-Minimal Set of ReLUs Sufficient for Controlling Outcome

```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```

x:

x00: 0
x01: 0
x02: 0
x03: 0
x04: 0
x05: 0

```
1 x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
0 x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
0 x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

0 x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
0 x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
0 x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

0 x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
0 x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
0 x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

0 x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
0 x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
1 x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```

Abductive ReLU Explanations

Subset-Minimal Set of ReLUs Sufficient for Controlling Outcome

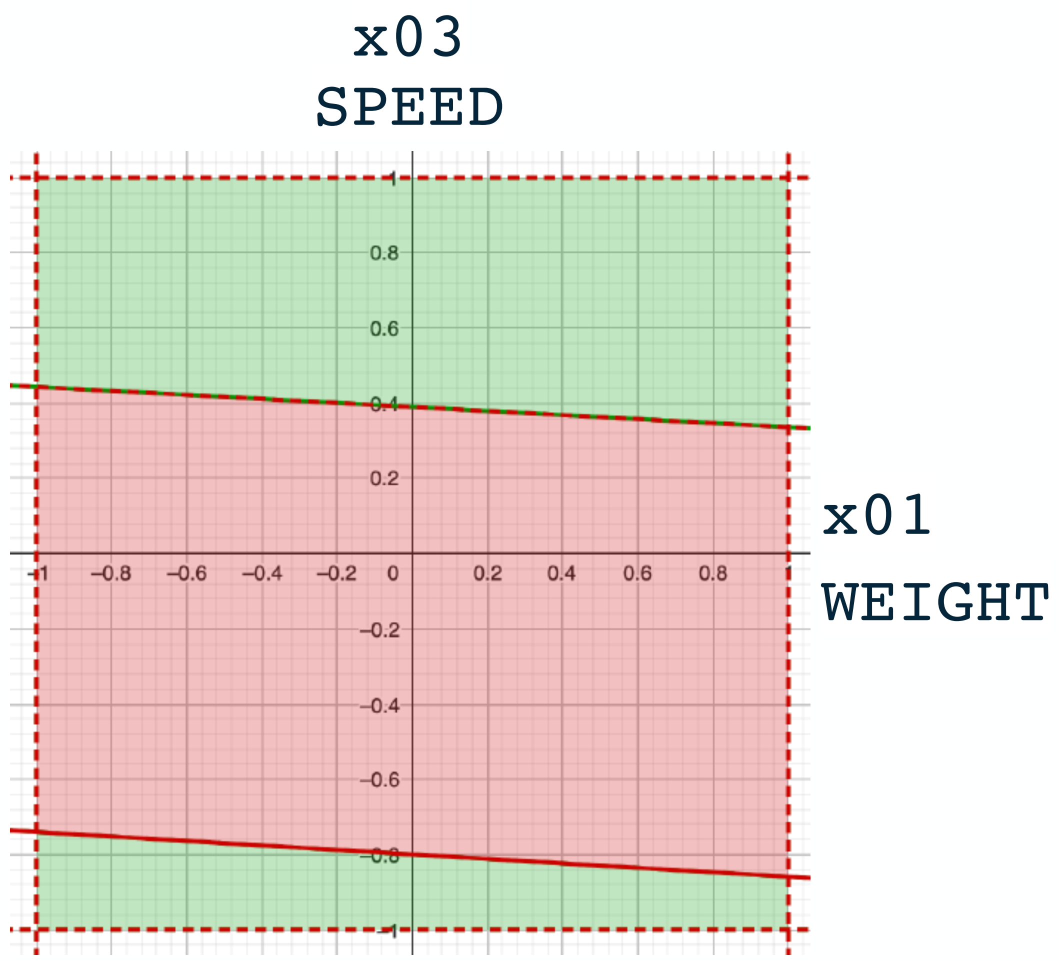
```
x00 = float(input())
x01 = float(input())
x02 = float(input())
x03 = float(input())
x04 = float(input())
x05 = float(input())
```

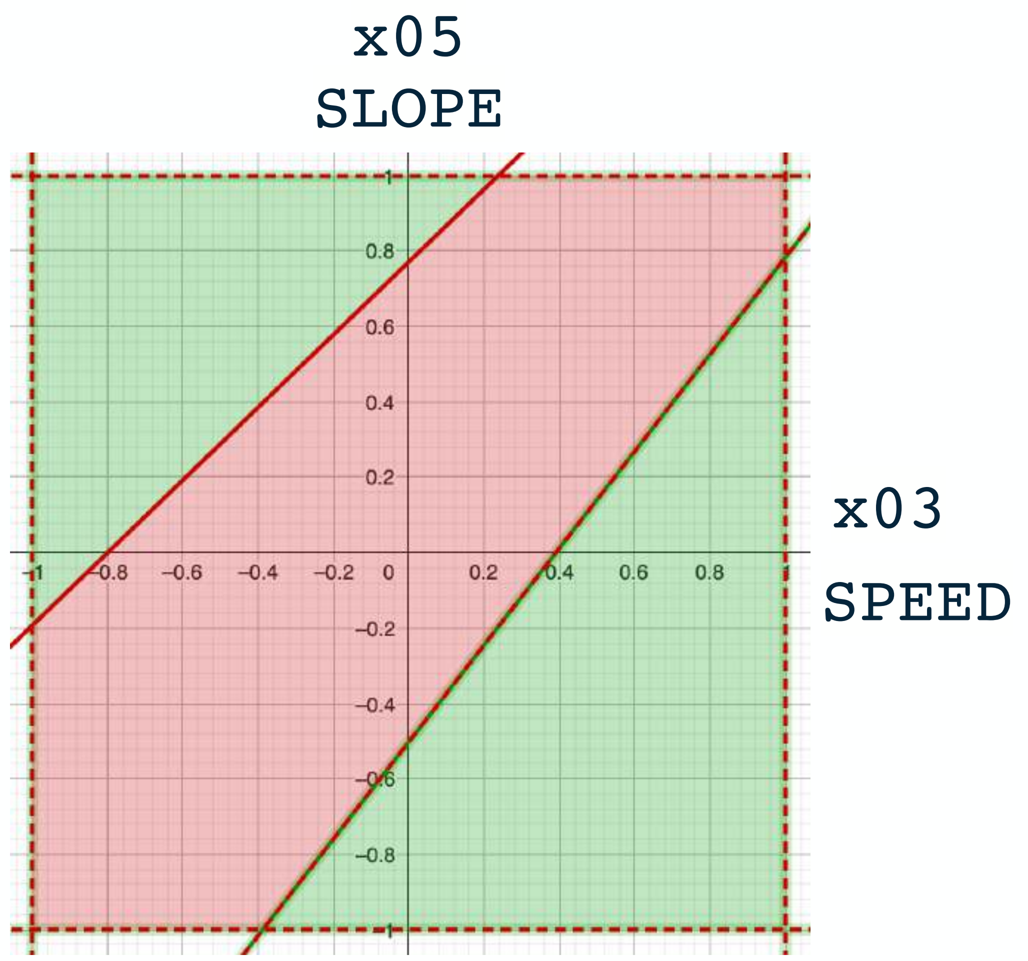
x:
x00: [-1, 1]
x01: [-1, 1]
x02: [-1, 1]
x03: [-1, 1]
x04: [-1, 1]
x05: [-1, 1]

ARXp = { x10, x11 }

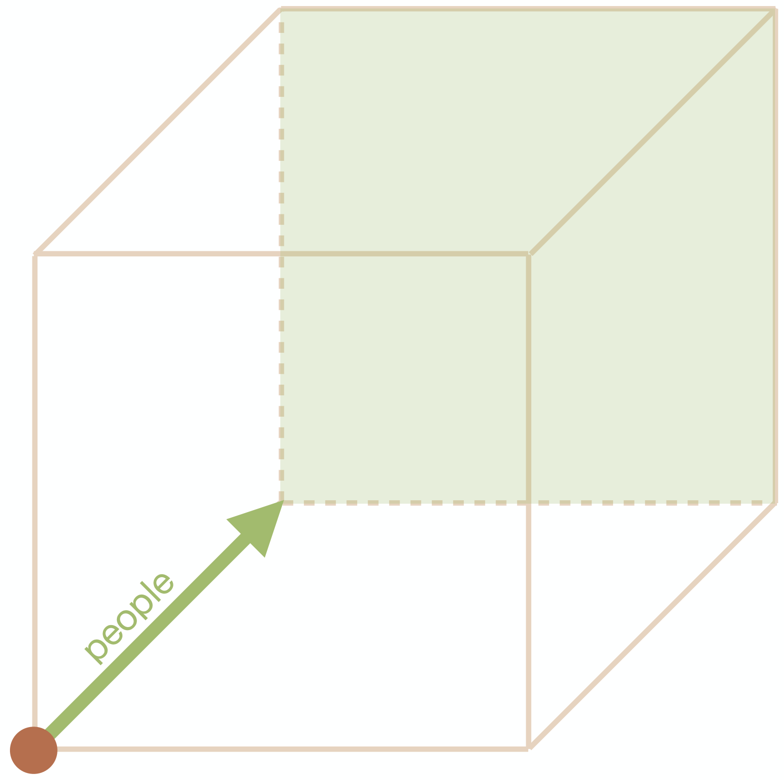
EXPLANATIONS IDENTIFY RELATIONSHIPS BETWEEN FEATURES

```
1 x10 = ReLU((0.120875)*x00 + (0.065404)*x01 + (0.097862)*x02 + (2.030051)*x03 + (0.101956)*x04 + (-2.103565)*x05 + (1.623834))
0 x11 = ReLU((0.113805)*x00 + (0.064486)*x01 + (0.090701)*x02 + (2.123338)*x03 + (0.076374)*x04 + (-1.651132)*x05 + (-0.828711))
? x12 = ReLU((0.755487)*x00 + (0.224640)*x01 + (0.344943)*x02 + (2.619876)*x03 + (0.346636)*x04 + (1.418635)*x05 + (-0.686885))

? x20 = ReLU((1.803209)*x10 + (1.222249)*x11 + (2.725716)*x12 + (-3.489653))
? x21 = ReLU((1.958950)*x10 + (2.388245)*x11 + (2.245851)*x12 + (-3.834811))
? x22 = ReLU((1.958103)*x10 + (2.273354)*x11 + (0.662405)*x12 + (-4.211086))

? x30 = ReLU((1.735994)*x20 + (0.666507)*x21 + (3.192344)*x22 + (-2.627086))
? x31 = ReLU((2.327110)*x20 + (2.685314)*x21 + (1.424807)*x22 + (-3.695113))
? x32 = ReLU((2.147212)*x20 + (2.285599)*x21 + (2.665507)*x22 + (-4.299974))

? x40 = ReLU((2.296390)*x30 + (1.980387)*x31 + (2.945360)*x32 + (-4.096463))
? x41 = ReLU((-0.552155)*x30 + (-0.828226)*x31 + (-0.495998)*x32)
? x42 = ReLU((-2.509773)*x30 + (1.199384)*x31 + (-0.245429)*x32 + (5.024773))

x50 = (-2.278012)*x40 + (0.180652)*x41 + (-16.663048)*x42 + (1864)
x51 = (2.278012)*x40 + (-0.180652)*x41 + (16.663048)*x42 + (-1864)
```



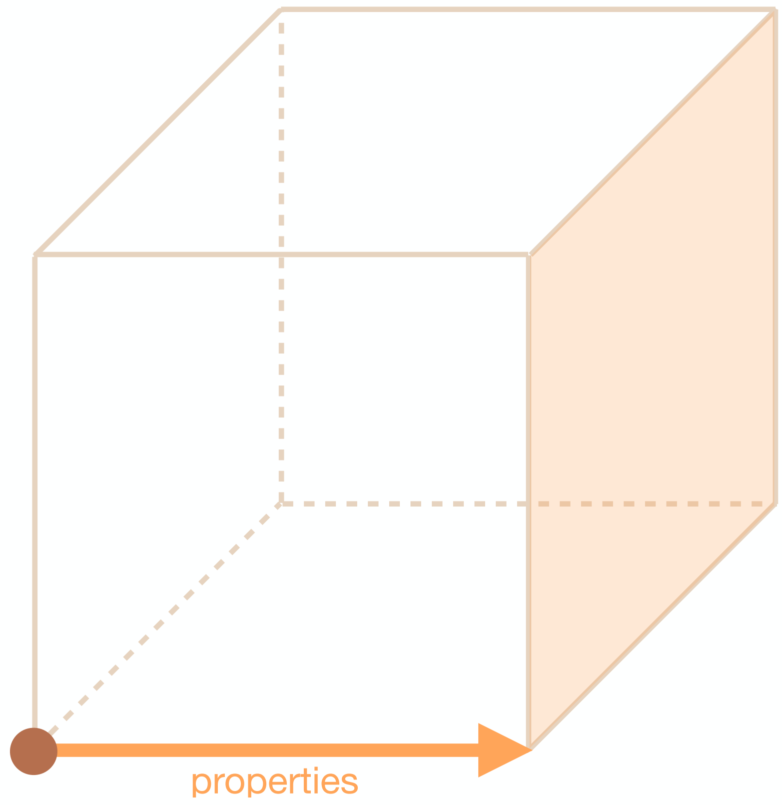


Research Agenda

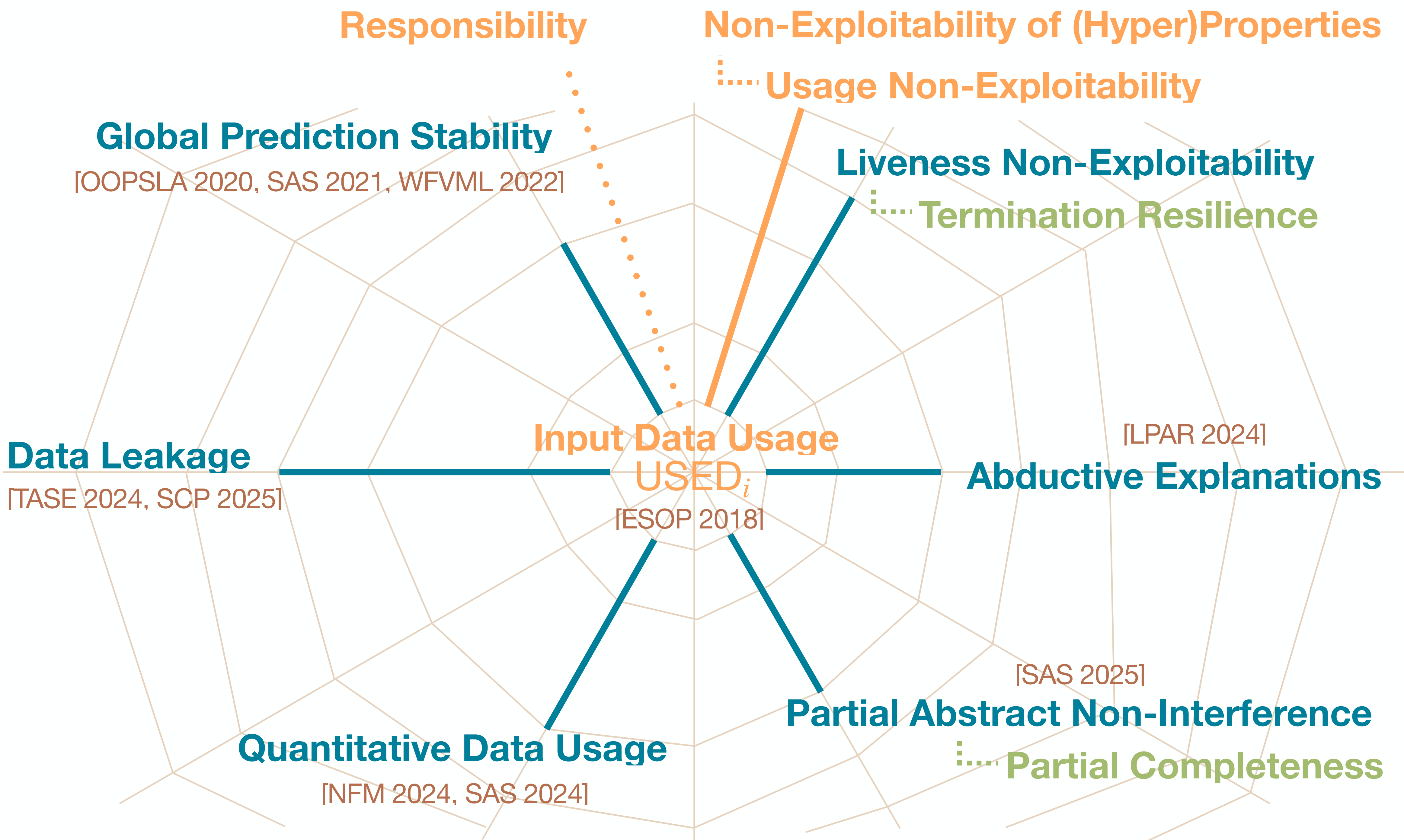


- **specification predicates** for machine learning models
- **incremental** and **compositional** verification approaches
- **relational explanations**

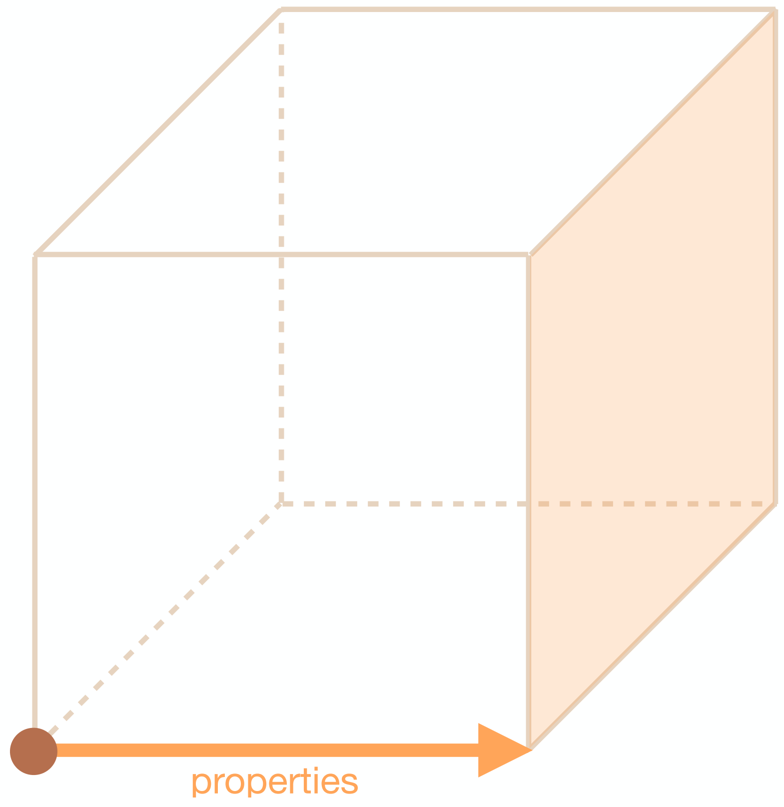
Research Agenda



- more (general) static analyses for **extensional properties** programs



Research Agenda

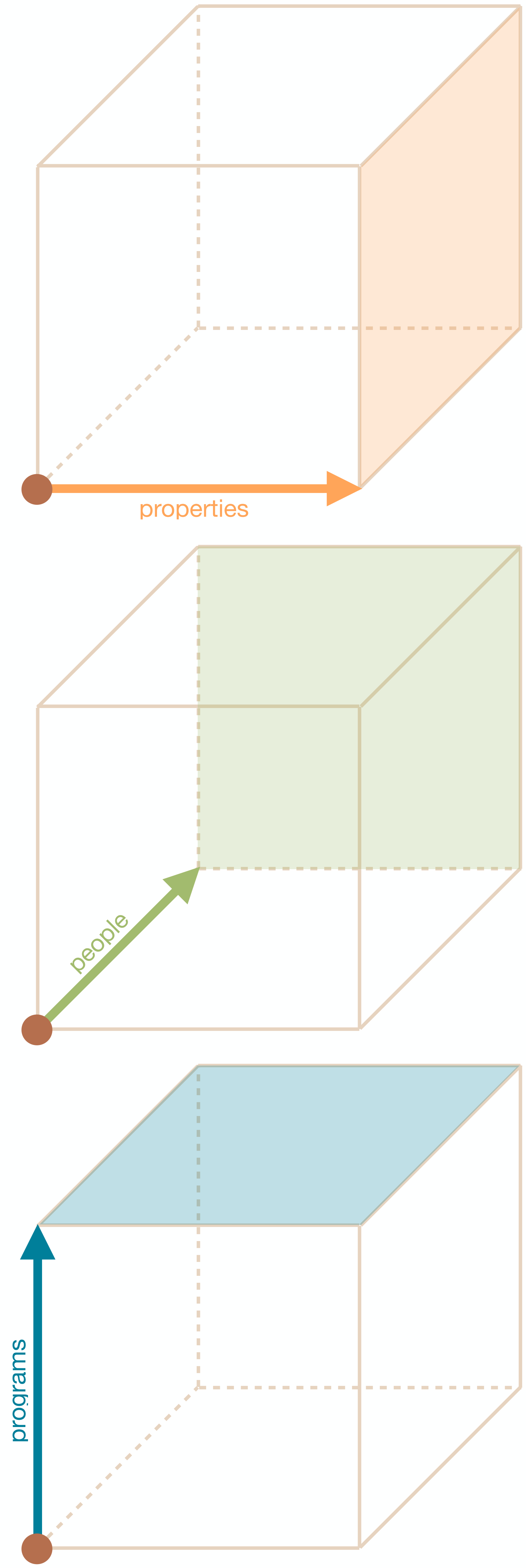


- more (general) static analyses for properties of the **observable behavior** of programs
- relate intensional and extensional program properties to **(partial) analysis completeness**

```
n = int(input())
i = 0

while (i < 10) {
   i = i + 1
}

if (i == n) {
   i = 1
}
```

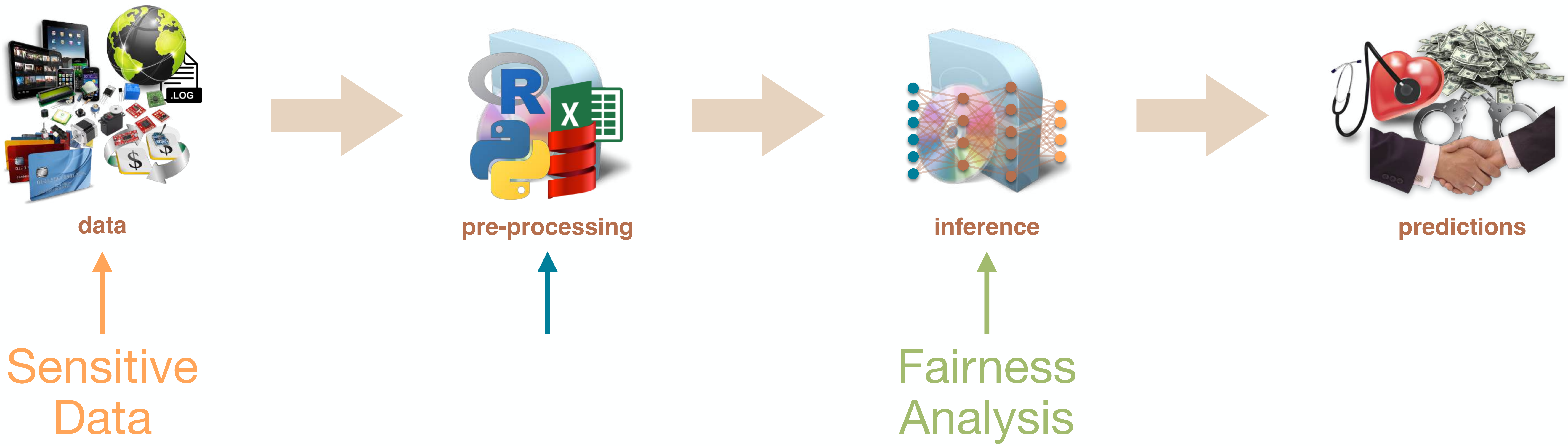
```
n = int(input())
i = 0

while (i < 10) {
   i = i + 1
}

if (i == n) {
   i = 1
} else {
   i = i
}
```

Research Agenda



- more (general) static analyses for properties of the **observable behavior** of programs
- relate intensional and extensional program properties to **(partial) analysis completeness**
- **specification predicates** for machine learning models
- **incremental** and **compositional** verification approaches
- **relational explanations**
- reason about **interactions** between *independently developed* and *evolving* software components
- deal with **trust boundaries** and untrusted software components

Machine Learning Pipeline




Global Prediction Stability
[OOPSLA 2020, SAS 2021, WFVML 2022]
Liveness Non-Exploitability
Termination Resilience
Data Leakage
[TASE 2024, SCP 2025]
Input Data Usage
$USED_i$
[ESOP 2018]
[LPAR 2024]
Abductive Explanations
[SAS 2025]
Partial Abstract Non-Interference
Partial Completeness
Quantitative Data Usage
[NFM 2024, SAS 2024]
THANKS!