Probabilistic Abstract Interpretation

An abstract-interpretation based framework for verification and static analysis of probabilistic programs

Michael Monerau

Courant Institute, NYU
École Normale Supérieure de Paris, France

Patrick Cousot

Courant Institute, NYU
École Normale Supérieure de Paris, France

Static analysis of probabilistic programs. What? Why?

INTRODUCTION

Goals

1. Verify properties of probabilistic programs
2. Predict probabilities, e.g.:
   – Branching probabilities
   – Outputs distributions
3. Seamlessly lift non-probabilistic analyses

Provide a formal basis for probabilistic static analysis & Design actual analyses

The mathematics behind probabilities

PROBABILITY THEORY

Probability theory
Measurable space

- $(\Omega, \mathcal{E}, \mu)$ is called a ***measurable space*** when :
  - Ω : set of all possible ***scenarios***
  - An ***event*** $E \in \mathcal{P}(\Omega)$ is a set of scenarios
  - $\mathcal{E} \in \mathcal{P}(\mathcal{P}(\Omega))$: set of ***observable events***
    - $\Omega \in \mathcal{E}$
    - Stable by complementation and countable union
  - $\mu : \mathcal{E} \longrightarrow [0,1]$: ***measure*** $[\ \mu(E) = Prob(E)\ ]$
    - $\mu(\emptyset) = 0$ and $\mu(\Omega) = 1$
    - $(A_i)_{i \in \mathbb{N}}$ countable family of *disjoint* events, then

$$\mu(\cup_i A_i) = \sum_i \mu(A_i)$$

- $\Omega = \{tail, heads\}^3$

- $\mathcal{E} = \mathcal{P}(\Omega)$
- $\forall \omega \in \Omega,$
  $\mu(\{\omega\}) = 1/8$

Probability theory
Event probability

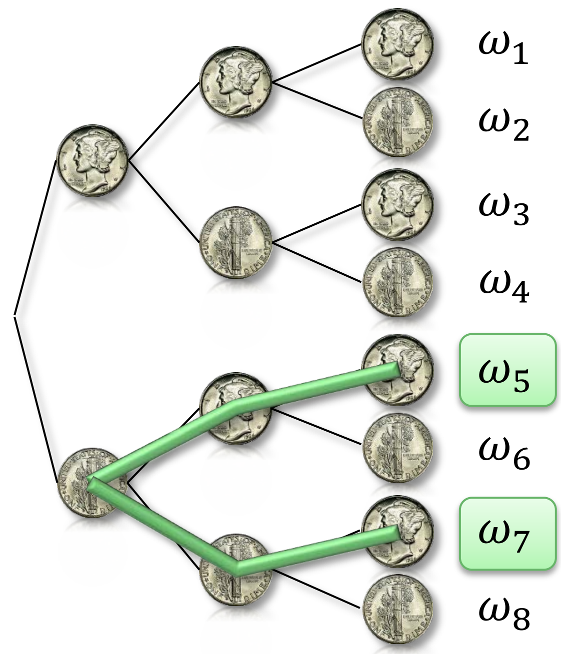
- Probability of an event $A \in \mathcal{E}$:

Characteristic function of A

$$P(A) = \mu(A) = \int_{\omega \in \Omega} \chi_A(\omega) d\mu(\omega)$$

EXAMPLE

- 3 throws of non-biased coins
  - $\Omega = \{tail, heads\}^3$
  - $\mathcal{E} = \mathcal{P}(\Omega)$
  - $\forall \omega \in \Omega, \mu(\{\omega\}) = 1/8$
- $\boxed{E = \left\langle \begin{matrix} coin_1 = tail \\ coin_3 = heads \end{matrix} \right\rangle}$



$\mu(\boxed{}) = 1/4$

Probability theory
Measurable function

$(E, \mathcal{E}, \cdot)$ and $(F, \mathcal{F}, \cdot)$ measurable spaces.
$X: E \rightarrowtail F$ is **measurable** iff

$$\forall B \in \mathcal{F}, \qquad X^{-1}(B) \in \mathcal{E}$$

Meaning:

- $\forall \omega \in \Omega$, an action $X(\omega)$ happens
- $B \in \mathcal{F}$: observable set of actions
- X **measurable** : if you can observe a set of actions, then you can observe the "parent" scenarios

Probability theory
Distribution

$\mathrm{X}: (\mathrm{E}, \mathcal{E}, \mu) \rightarrowtail (\mathrm{F}, \mathcal{F}, \cdot)$ measurable. The **distribution** $\boldsymbol{X}(\boldsymbol{\mu})$ of X is a measure on F :

$$\forall B \in \mathcal{F}, \qquad \mathrm{X}(\mu)(B) = \mu\left(\mathrm{X}^{-1}(B)\right)$$

Meaning:

Probability (*actions B*)

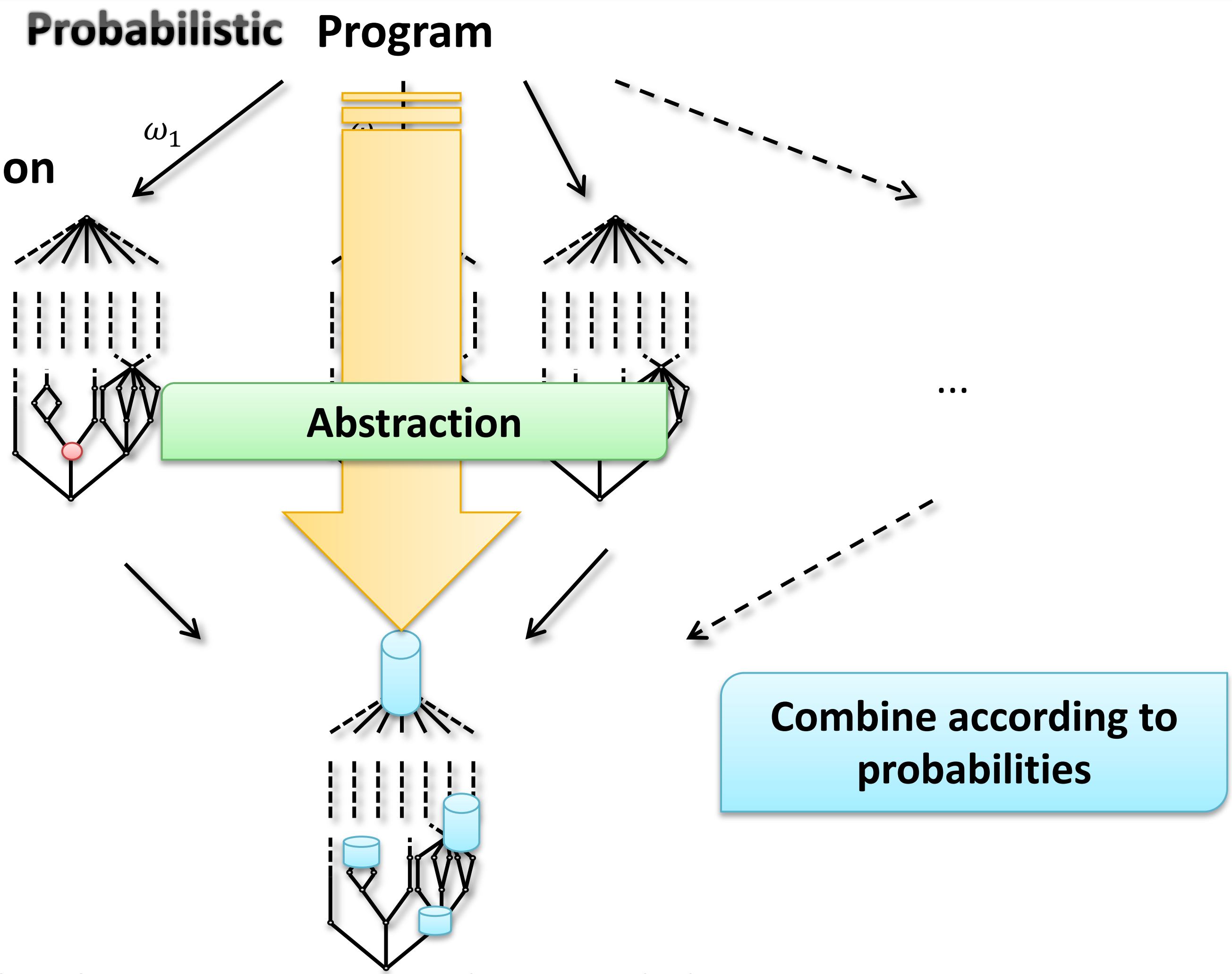
=

Probability (*“parent” scenarios*)

The Main Idea

Probabilistic

Abstract Interpretation



Our concrete probabilistic semantics

PROBABILISTIC CONCRETE SEMANTICS

Probabilistic Concrete Semantics NON-probabilistic case

- In non-probabilistic setting:
  - Semantic domain $\langle \mathcal{D}, \preccurlyeq \rangle$
  - Properties of programs are some $\Gamma \in \mathcal{P}(\mathcal{D})$
  - For a program P, $\exists F : \mathcal{P}(\mathcal{D}) \longrightarrow \mathcal{P}(\mathcal{D})$

$$S[\![P]\!] = lfp^{\subseteq} F$$

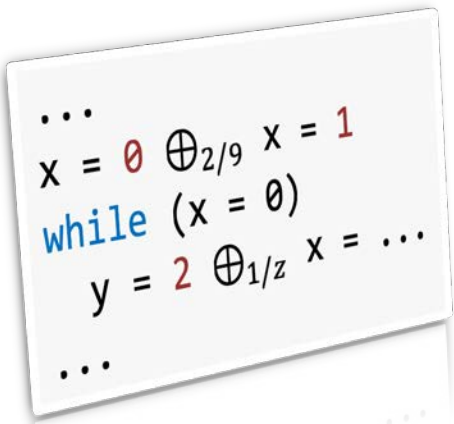
- Properties are abstracted by a Galois connection

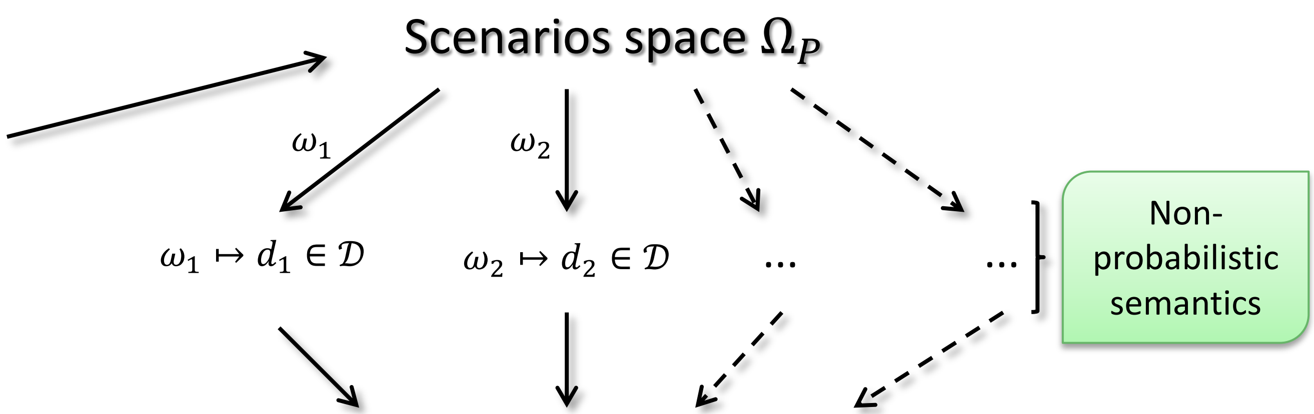
$$\langle \mathcal{P}(\mathcal{D}), \subseteq \rangle \leftrightarrows \langle \mathcal{A}, \sqsubseteq \rangle$$

Abstract F to $\bar{F} : \mathcal{A} \longrightarrow \mathcal{A}$ and find / over-approximate

$$\bar{S}[\![P]\!] = lfp^{\sqsubseteq} \bar{F}$$

Probabilistic Concrete Semantics
Probabilistic case





Program semantics $\boldsymbol{S_p[\![P]\!] : \Omega_P \rightarrowtail \mathcal{D}}$

For each scenario,
a *non-probabilistic* fixpoint semantics:

$$\forall \omega \in \Omega, \qquad S_p[\![P]\!](\omega) = lfp^{\sqsubseteq} F_\omega$$

$$\text{Let } F_\Omega : \left| \begin{array}{ccc} (\Omega_P \to \mathcal{D}) & \longrightarrow & (\Omega_P \to \mathcal{D}) \\ s & \longmapsto & [\omega \mapsto F_\omega(s(\omega))] \end{array} \right.$$

Probabilistic semantics :

$$\boldsymbol{S_p[\![P]\!] = lfp^{\dot{\sqsubseteq}} F_\Omega}$$

Probabilistic Concrete Semantics
Adding probabilities

The semantics has probability information

Given $\langle \boldsymbol{\Omega_P}, \boldsymbol{\mathcal{E}}, \boldsymbol{\mu} \rangle$

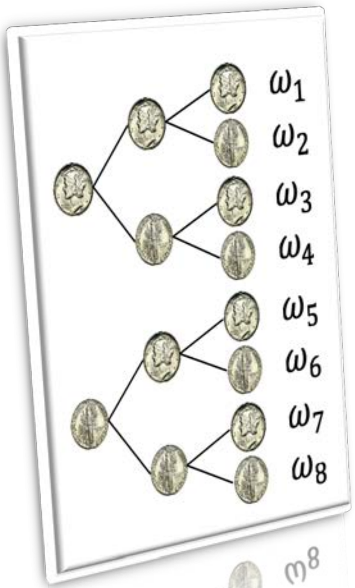
$\boldsymbol{S_p[\![P]\!] : \Omega_P \rightarrowtail \mathcal{D}}$

- Observable events
  $\mathcal{E} \subseteq \mathcal{P}(\Omega_P)$
- $\mu : \mathcal{E} \longrightarrow [0,1]$
  Probability of an event

- Observable properties
  $\mathcal{F} \subseteq \mathcal{P}(\mathcal{D})$
- $S_p[\![P]\!](\mu) : \mathcal{F} \longrightarrow [0,1]$
  Probability of a property

$S_p[\![P]\!]$ cannot say anything on non-observable properties, *ie.* outside $\mathcal{F}$.

Probabilistic Concrete Semantics
Sanity Checker



Many semantics can describe the same situation. So we quotient by picking only one representation using a :

Sanity Checker $V : (\Omega_P \to \mathcal{D}) \longrightarrow \{True, False\}$

For instance in the 3 coins flips case :

- Semantics $S_p[\![P]\!] : \Omega_P \rightarrowtail \mathcal{D}$. But...

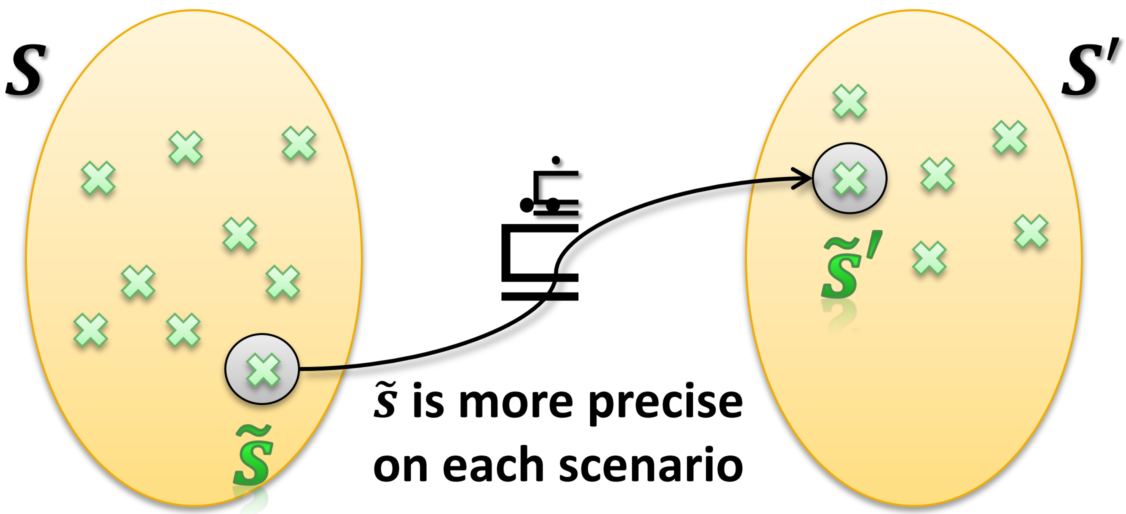
$$\forall \sigma \in \boldsymbol{\pi}(\Omega_P), let\ S_p^{\sigma}[\![P]\!] = S_p[\![P]\!] \circ \sigma$$

$S_p^{\sigma}[\![P]\!]$ is acceptable too

Concrete Domain: $\mathcal{PD}_p^V = \mathcal{P}(\{s : \Omega_P \rightarrowtail \mathcal{D} \mid V(s)\})$

Probabilistic Concrete Semantics
Order of logical implication

Concrete Domain : $\mathcal{PD}_p^V = \mathcal{P}(\{s\colon \Omega_P \rightarrowtail \mathcal{D} \mid V(s)\})$



$$\forall S, S' \in \mathcal{PD}_p^V, \qquad S \ \ddot{\sqsubseteq}\ S' \Leftrightarrow \forall s \in S, \exists s' \in S', s \mathrel{\dot{\sqsubseteq}} s'$$

“Abstraction is real, probably more real than nature” Josef Albers

ABSTRACTION

Abstraction
Which way to go?

- 3 abstractions of $\mathcal{PD}_p^V \subseteq \Omega_P \rightarrowtail \mathcal{D}$

$$\boldsymbol{S_p[\![P]\!] : \Omega_P \rightarrowtail \mathcal{D}}$$

1. Abstract Ω_P

2. Abstract $\mathcal{D}$

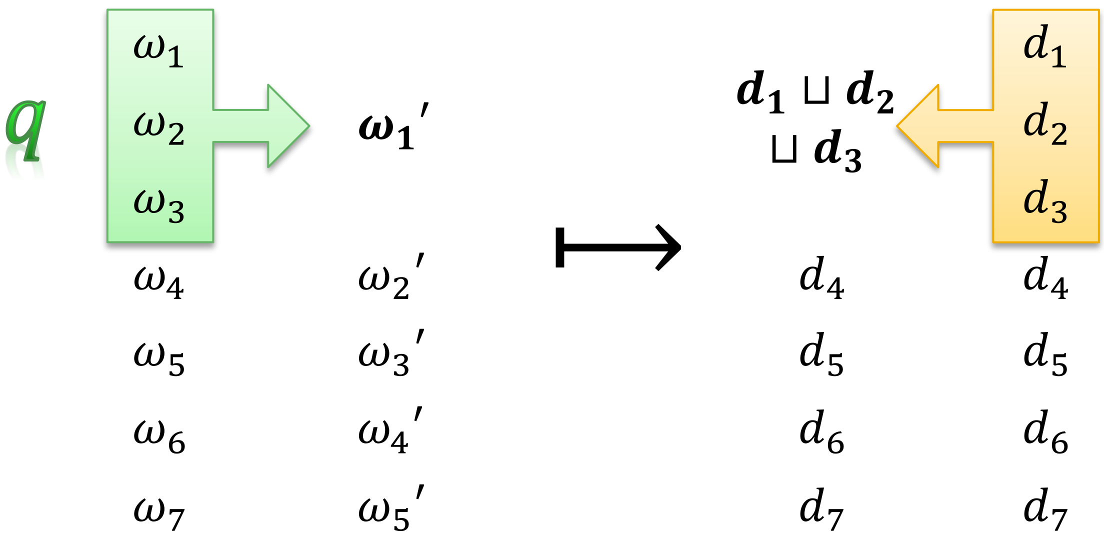
3. Abstract functions to distributions

Abstract away probability details

ABSTRACTION ON THE Ω_P SIDE

1. Abstracting Ω_P
Quotient

$$\begin{tikzcd}
S_p[\![P]\!] : \Omega_P \arrow[rr] \arrow[dr, "q"'] & & \mathcal{D} \arrow[d, "\alpha_\Omega"] \\
& \Omega_P{}' \arrow[r] & \mathcal{D}
\end{tikzcd}$$



Everything is lifted by q

- q is measurable
- $\mu \dashrightarrow \mu \circ q^{-1}$ (q-distribution)

1. Abstracting Ω_P
Expressing non-determinism by quotienting

```
[x = 0 □ x = 1]
if (z = 0)
  y = 2  1/4⊕3/4  y = 4
else
  y = 1  1/5⊕4/5  y = 3
```

- q “forgets” probabilistic choice for **x** :
  - $q : \{l, r\}^3 \rightarrowtail \{l, r\}^2$
  - $q(a, b, c) = (b, c)$
- Probabilistic properties depending on **x** are no longer observable, but those independent from **x** are still observable

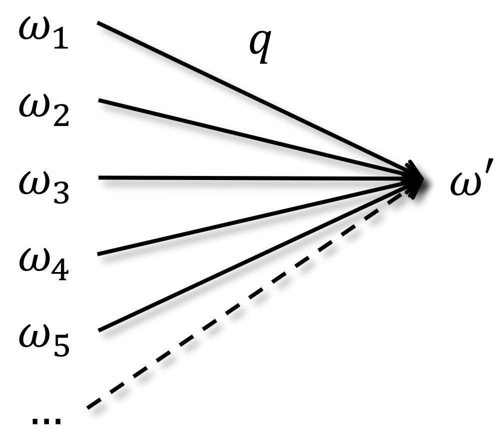
$\omega_1, \omega_2 \Rightarrow \boldsymbol{\omega_1}' \longmapsto \langle x \in [0,1], y = 2\rangle \Leftarrow \langle x = 0\,; y = 2\rangle, \langle x = 1\,; y = 2\rangle$

$\omega_3, \omega_4 \Rightarrow \boldsymbol{\omega_2}' \longmapsto \langle x \in [0,1], y = 4\rangle \Leftarrow \langle x = 0\,; y = 4\rangle, \langle x = 1\,; y = 4\rangle$

Non-determinism = abstraction of probabilistic choice

1. Abstracting Ω_P
Safe-abstraction

- If $\Omega'_P = \text{singleton} = \{\omega'\}$
  - Still sound (every scenario output has been joined)
  - No more probabilities



Brings back to the usual Abstract Interpretation setting

Lift an existing static analysis to the probabilistic setting

ABSTRACTION ON $\mathcal{D}$ SIDE

2. Abstracting $\mathcal{D}$
Lifting a classical analysis

- Hypothesis :

$$\langle \mathcal{P}(\mathcal{D}), \subseteq \rangle \leftrightarrows \langle \mathcal{A}, \sqsubseteq \rangle$$

- We have the semantics :

$$S_p[\![P]\!] : \Omega_P \rightarrowtail \mathcal{D}$$

And the semantic domain :

$$\mathcal{P}\mathcal{D}_p^V \approx \mathcal{P}(\Omega_P \rightarrowtail \mathcal{D}) \Longrightarrow \boldsymbol{\mathcal{P}(\Omega_P \longrightarrow \mathcal{P}(\mathcal{D}))}$$

How to make $\mathcal{P}(\mathcal{D})$ appear ?

2. Abstracting $\mathcal{D}$
Lifting a classical analysis

$$\begin{array}{rcl} S_p[\![P]\!] : \Omega_P & \longrightarrow & \mathcal{D} \\ \omega & \longmapsto & S_p[\![P]\!](\omega) \end{array}$$

$$\begin{array}{rcl} \alpha_{\mathcal{P}}(S_p[\![P]\!]) : \Omega_P & \longrightarrow & \mathcal{P}(\mathcal{D}) \\ \omega & \longmapsto & \{S_p[\![P]\!](\omega)\} \end{array}$$

$$\begin{tikzcd} \mathcal{P}(\Omega_P \rightarrowtail \mathcal{D}) \arrow[r, shift left, "\alpha_{\mathcal{P}}"] & \mathcal{P}(\Omega_P \rightarrowtail \mathcal{P}(\mathcal{D})) \arrow[l, shift left] \arrow[d, shift left, "\alpha_{\mathcal{A}}"'] \\ & \mathcal{P}(\Omega_P \rightarrowtail \mathcal{A}) \arrow[u, shift left] \end{tikzcd}$$

In the end, an abstraction saying :

« In scenario ω, abstract property $\overline{S[\![P]\!]}(\omega)$ is verified »

$$\begin{array}{rcl} \alpha_{\mathcal{A}}\,(\omega \mapsto \Gamma_\omega) : \Omega_P & \longrightarrow & \mathcal{A} \\ \omega & \longmapsto & \alpha(\Gamma_\omega) \end{array}$$

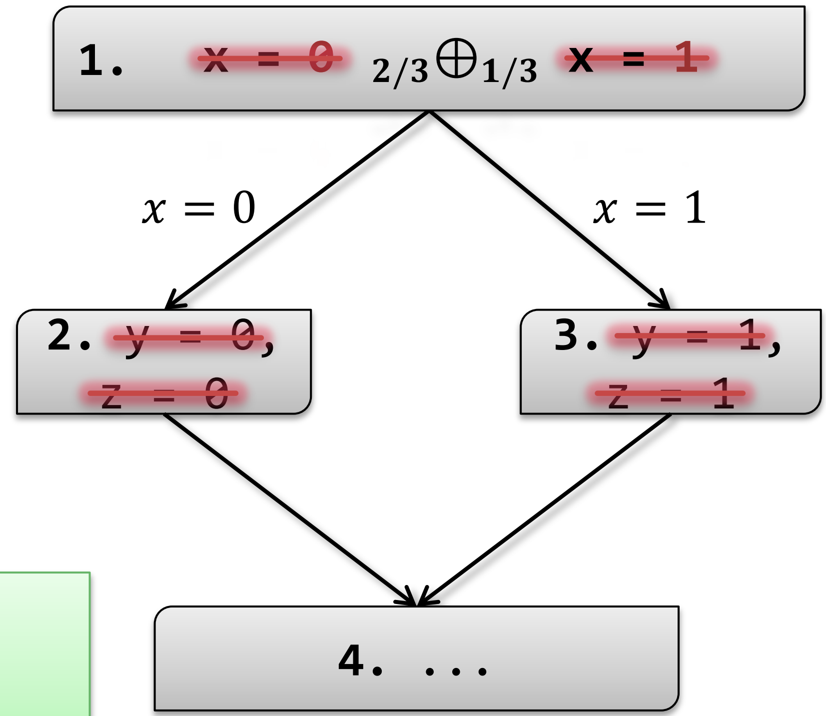
2. Abstracting $\mathcal{D}$
Example

Control flow estimation

Probabilistic semantics :

$\omega_1 \mapsto \langle 1, \quad \rangle \Big\langle 2, \quad \Big\rangle \Big\langle 4, \quad \Big\rangle$

$\omega_2 \mapsto \langle 1, \quad \rangle \Big\langle 3, \quad \Big\rangle \Big\langle 4, \quad \Big\rangle$



<u>Abstraction</u> :
Keep labels only
to infer just control flow probabilities

Abstract measurable functions into their distributions

DISTRIBUTION ABSTRACTION

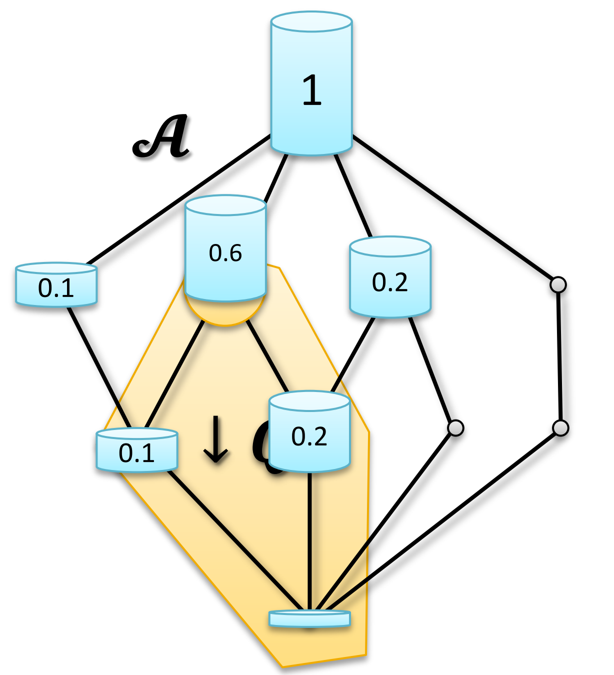
3. Distribution abstraction From functions to distributions

- Abstract semantics
  $\overline{S[\![P]\!]} : \Omega_P \rightarrowtail \langle \mathcal{A}, \mathcal{F} \rangle$

$$\Big\downarrow \boldsymbol{\alpha_{\mathcal{L}}}$$

- Semantics distribution :
  $\overline{S[\![P]\!]}(\mu) : \mathcal{F} \rightarrowtail [0,1]$

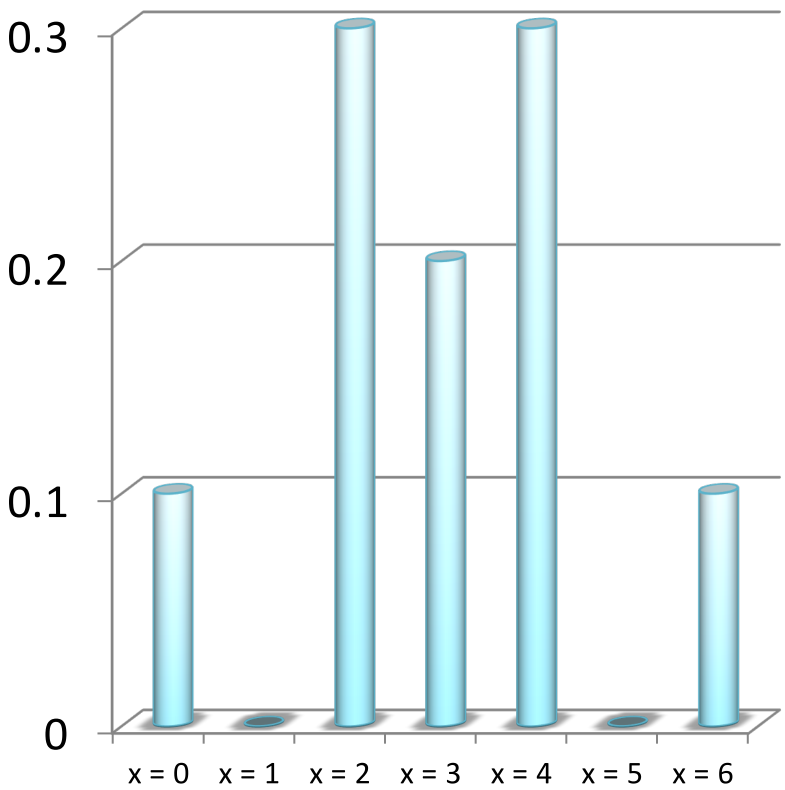
For $Q \in \mathcal{A}$,
$\downarrow Q = \{Q' \in \mathcal{A} \mid Q' \sqsubseteq Q\}$



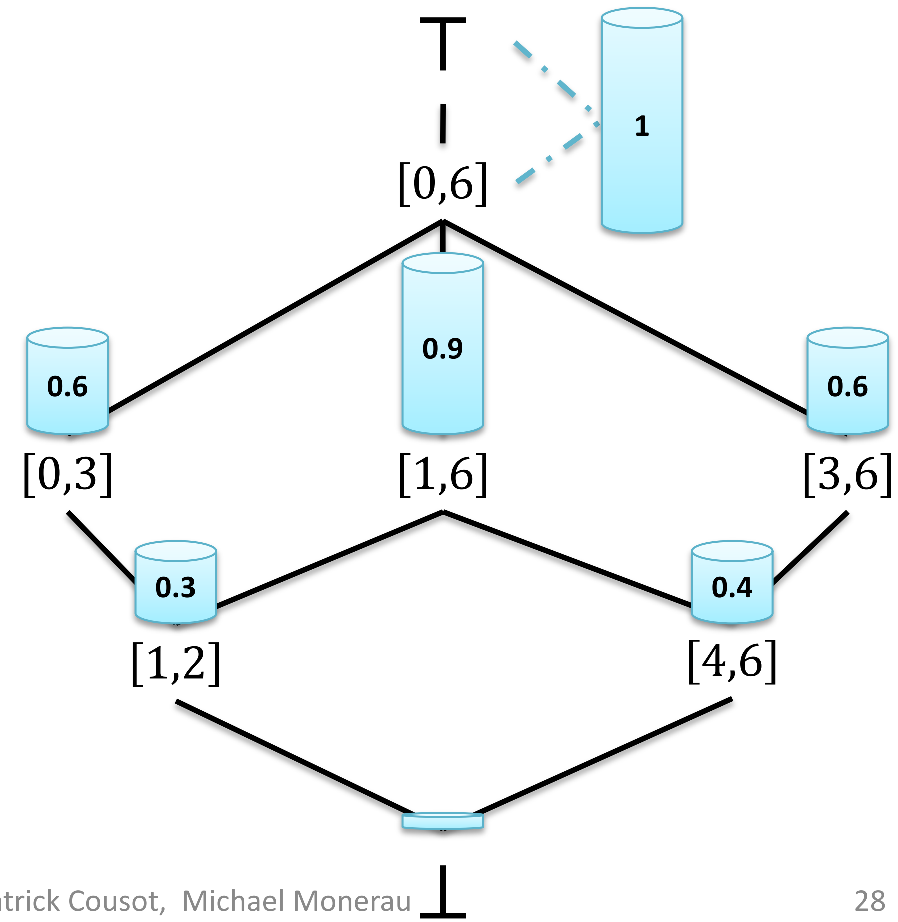
Information we want
$\boldsymbol{\overline{S[\![P]\!]}(\mu)(\downarrow Q)}$

3. Distribution abstraction
Example : putting weight on the lattice

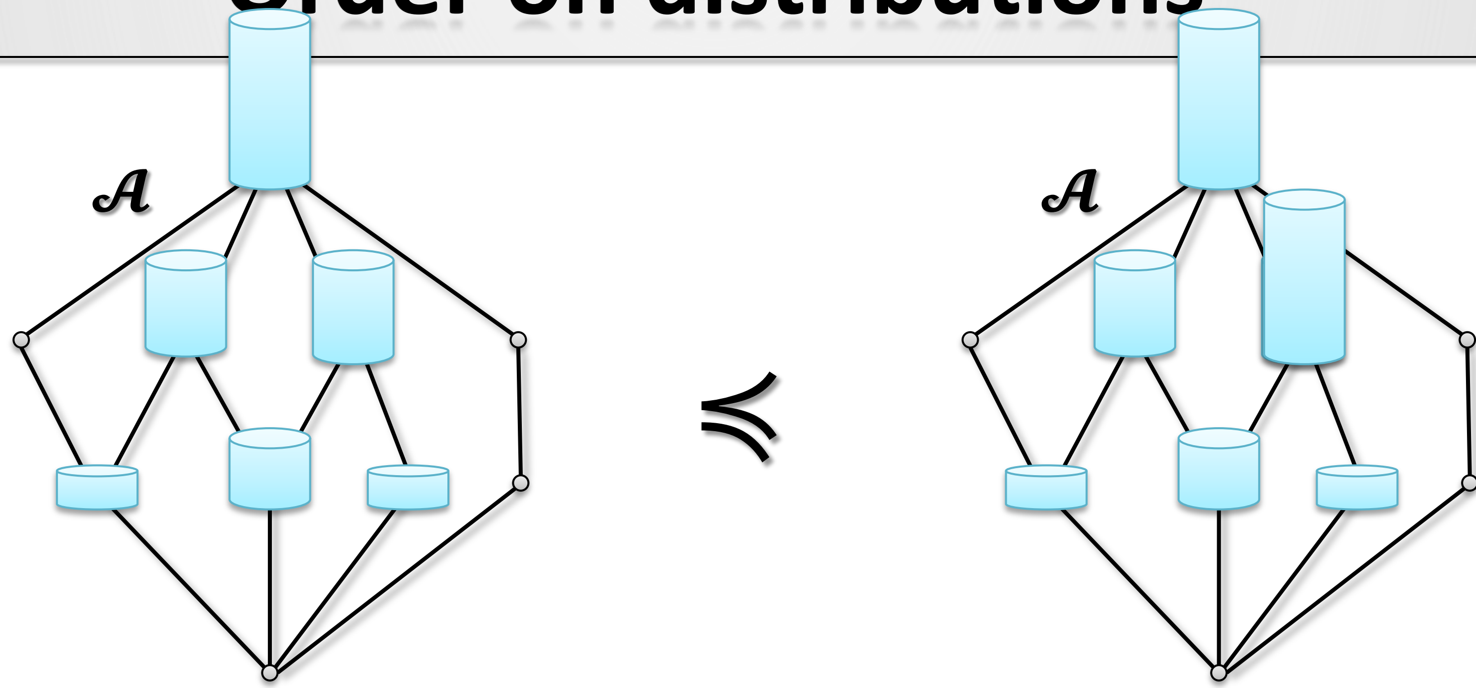
A distribution example



A corresponding lattice



3. Distribution abstraction
Order on distributions



Let l_1 and l_2 be two distributions,

$$l_1 \preccurlyeq l_2 \quad \Leftrightarrow \quad \forall Q \in \mathcal{A}, l_1(\downarrow Q) \geq l_2(\downarrow Q)$$

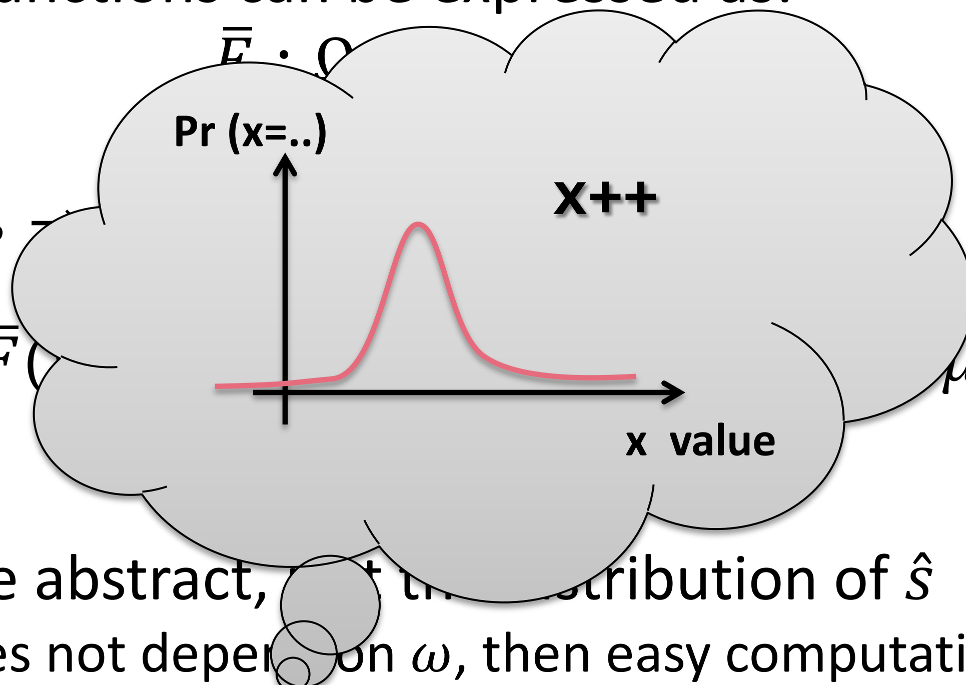
3. Distribution abstraction Transfer functions

- Transfer functions can be expressed as:

$\bar{F} : \Omega$

- For $\hat{s} : \Omega_P \rightarrow$

$P(\bar{F}($ ... $\mu(\omega)$



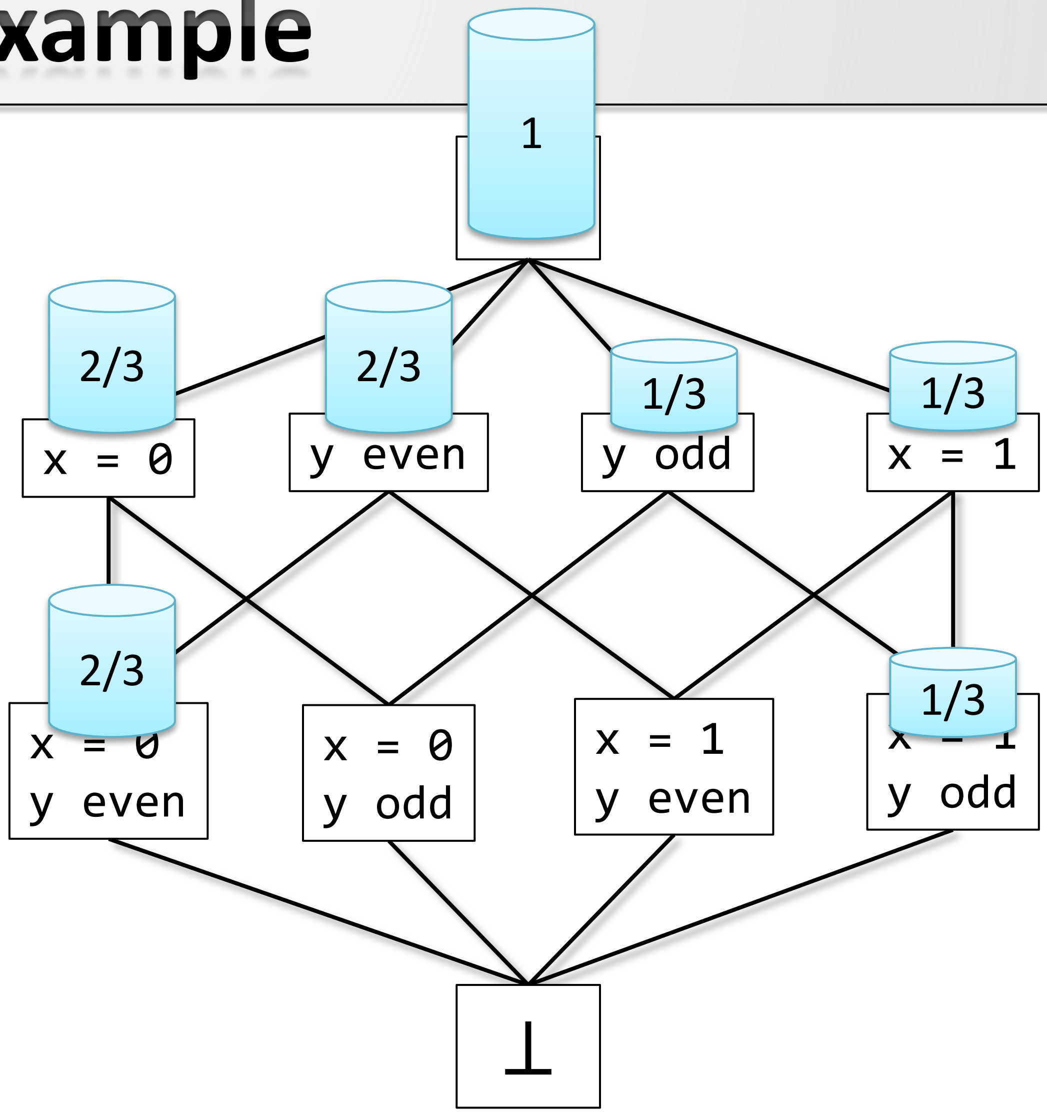
- But: in the abstract, ... tribution of $\hat{s}$
  - If $\bar{F}$ does not depen... on ω, then easy computation with just the $\hat{s}$ distribution
  - Otherwise, back to the concretisations (thus the precision of the sanity checker is important)
  - Too hard to compute? Over-approximate

3. Distribution abstraction Example

```
x = 0  2/3⊕1/3  x = 1
if (x = 0)
  y = 2  1/4⊕3/4  y = 4
else
  y = 1  1/5⊕4/5  y = 3
```

Our abstract domain :

The final distribution



Iteration in the abstract, composing the abstractions

Branching estimation

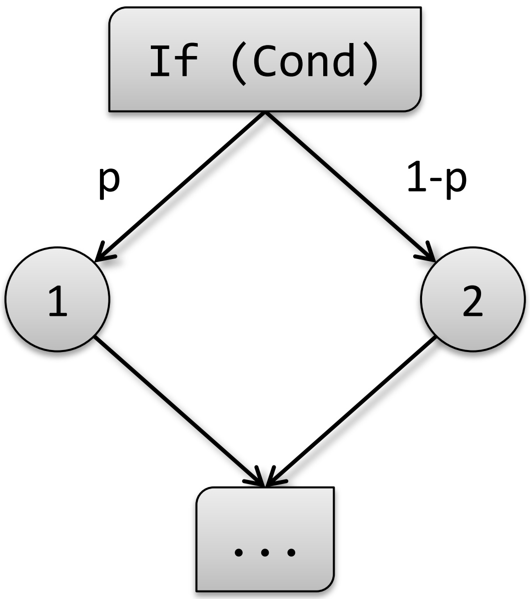
ON THE WAY TO MAKING THE ANALYSIS FULLY AUTOMATIC (INCLUDING INFINITE LATTICES)

Automatic analysis
The issue of branching

Goal : Finding abstract distributions
$\mathcal{P}(\mathcal{A}) \longrightarrow [0,1]$ automatically

- Transfer functions : OK
- Branching

$$P(\Gamma) = P(\Gamma \cap \ left) + P(\Gamma \cap \ right)$$
$$= \boxed{p}\,\underbrace{P(\Gamma \mid left)}_{\text{Computed in } 1} + \boxed{(1-p)}\,\underbrace{P(\Gamma \mid right)}_{\text{Computed in } 2}$$



Essential to estimate p

Automatic analysis Branching analysis

Branching with respect to a condition « Cond »

Let $\mathcal{F}$ denote the observable actions in $\mathcal{A}$, and p the probability of branching left

- Then, 2 cases :
  - Cond $=$ true is equivalent to a $C \in \mathcal{F}$
    - At the test location, the analysis discovered a distribution ν, then $\nu(C) \leq p$
    - If $\exists \check{C} \in \mathcal{F}$ which is the complement of Cond, then $1 - \nu(\check{C}) \geq p$
    - So complements should also be abstracted precisely
  - Otherwise, nothing can be said : $p \in [0,1]$

Automatic analysis
If-Else example

```
x = 0 2/3⊕1/3 x = 1
if (x = 0)
  y = 2 1/4⊕3/4 y = 4
else
  y = 1 1/5⊕4/5 y = 3
```

$$P(x = 0) = 2/3 \qquad P(x \neq 0) = 1/3$$

Tight bound on branching probability :
2/3 & 1/3

- At the end :

$$P(y\ even) = P(y\ even \cap x = 0) + P(y\ even \cap x \neq 0)$$

$$= \frac{2}{3} P(y\ even \mid x = 0) + \frac{1}{3} P(y\ even \mid x = 1)$$

$$= \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 0$$

$$= \frac{2}{3}$$

The abstract transfer function for If-Else on the distribution has been computed

Automatic analysis
While

```
while (Cond)
  body
```

- Same thing with Cond for branching
- But it may depend on the number of iterations too

Goal: Determine an over-approximating transfer function as precise as possible

- 2 main cases :
  - **Known influence of the body** on the distribution and on the branching : *mathematical formula* for the new distribution
  - **Unknown influence** : unroll until branching probability is small (or after *N* loops) and then over-approximate possible remaining loop iterations [widening]

Automatic analysis
While example

```
0. loop = 0
1. x = 0 1/3⊕2/3 x = 1
2. while (x = 0)
3.    x = 0 1/4⊕3/4 x = 1
4. loop++
```

Probabilities at location 2 :

$$P(x_2 = 0 \land loop_2 = 0) = 1/3$$
$$P(x_2 = 1 \land loop_2 = 0) = 2/3$$
$$P(x_2 = 0 \land loop_2 = 1) = 1/3 * 1/4$$
$$P(x_2 = 1 \land loop_2 = 1) = 1/3 * 3/4$$

- How to infer that ?

$$P(x_2 = b \land loop_2 = i) = P(x_4 = b \land loop_4 = i - 1)$$
$$= P(x_4 = b) \cdot P(loop_2 = i - 1)$$

Easy recurrence equation

On probabilistic static analysis

CONCLUSION

Probabilistic analysis: Related Work

- Works towards probabilistic Abstract Interpretation:
  - ≈ Abstraction of our Law-abstraction [Monniaux ‘00]
  - ≈ Mean behavior abstraction [Wiklicky ‘02]
- Probabilistic Model Checking [Kucera ‘10]
- Weakest precondition semantics [McIver ‘97]
- Strongest postcondition semantics [Hehner ‘04]

Conjecture:

Abstractions expressible in our framework

Future work

- More precise Law-style abstractions (relational abstractions)
- More precise techniques to predict branching
- Consider other abstractions for While loops to make their over-approximation more precise
- Implementation & Experimentation
- Non-Galois setting

Summing it up

- New probabilistic extension of Abstract Interpretation
- New way to express probabilistic semantics
- New ways to design probabilistic static analyses
- Lift classical static analyses to a probabilistic setting
- The precision of probabilistic and semantic abstractions are independent
- Very expressive, and precision can be adjusted by modular abstractions

A quick overview of Abstract Interpretation

ABSTRACT INTERPRETATION

Getting the idea on a simple example

OVERVIEW OF THE FRAMEWORK