Compositional Solution Space Quantification for Probabilistic Software Analysis

Mateus Borges, Marcelo d'Amorim (UFPE)
Antonio Filieri (Stuttgart)
Corina Pasareanu (CMU SV and NASA Ames)
Willem Visser (Stellenbosch)

Uncertain Environments

Uncertain Environments

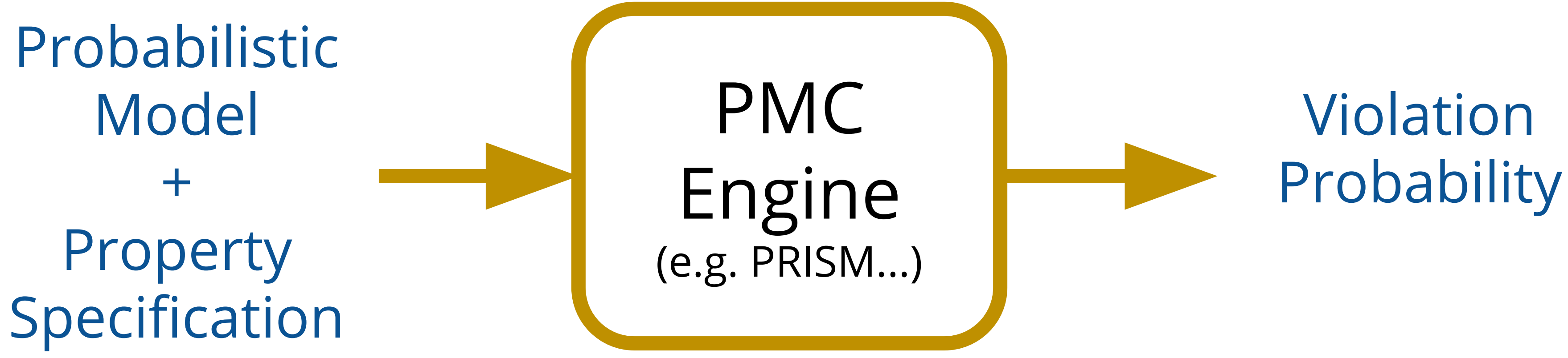
Quantitative Properties

Not restricted to boolean values

Establish non-functional requirements

➔ *Reliability, performance…*

Probabilistic Model Checking



Probabilistic Model Checking

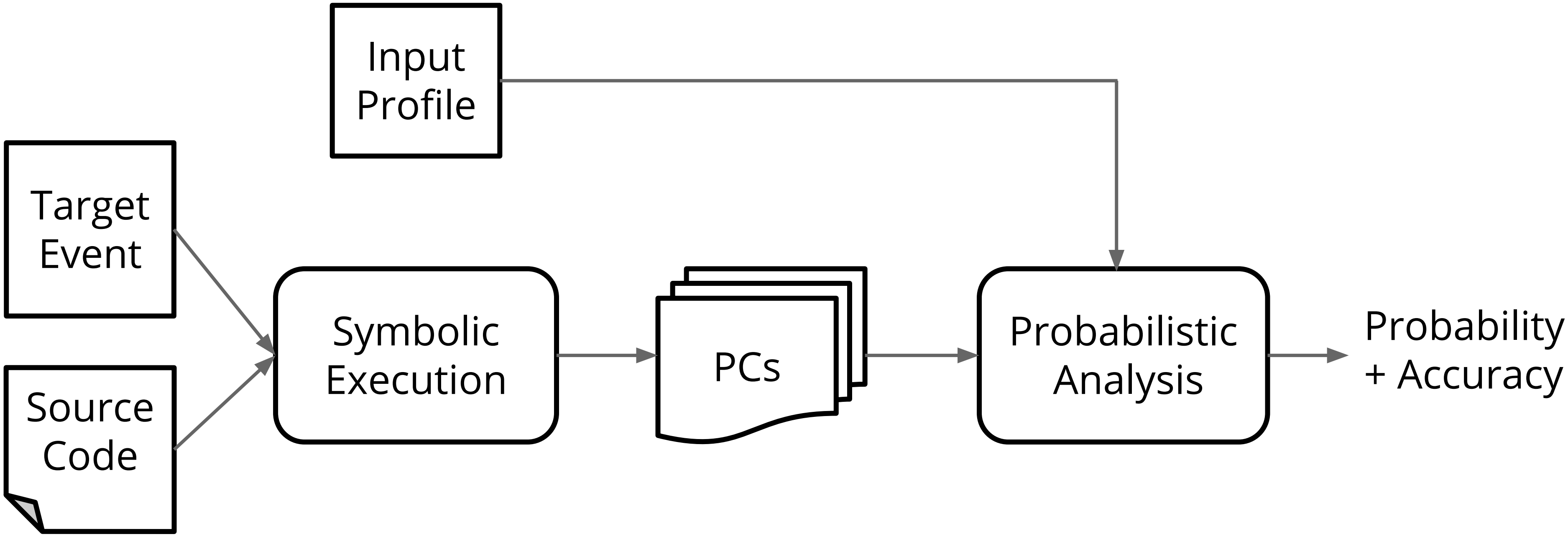
Problem: can be expensive!

➔ You need to learn a new modelling language
➔ You need to model the system

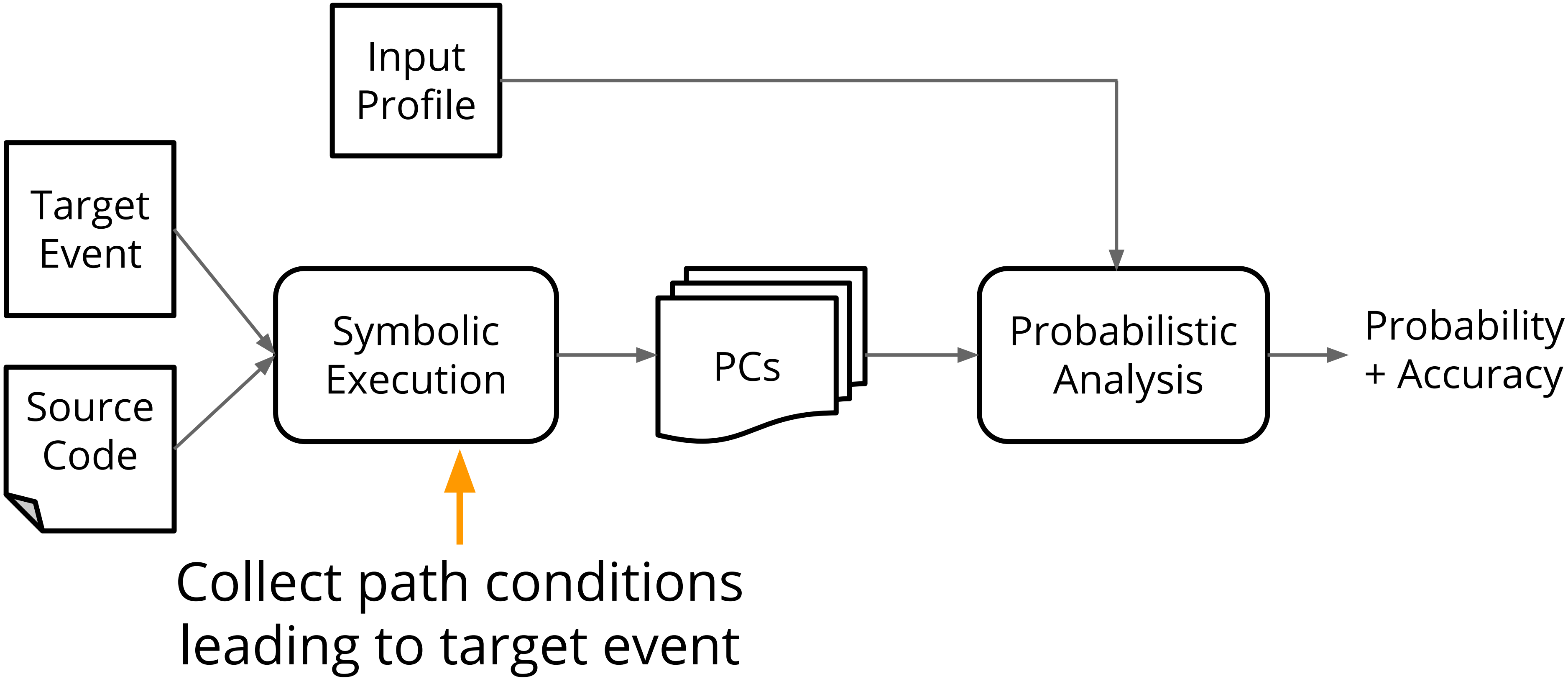
We would like to analyze *code*

Probabilistic Software Analysis

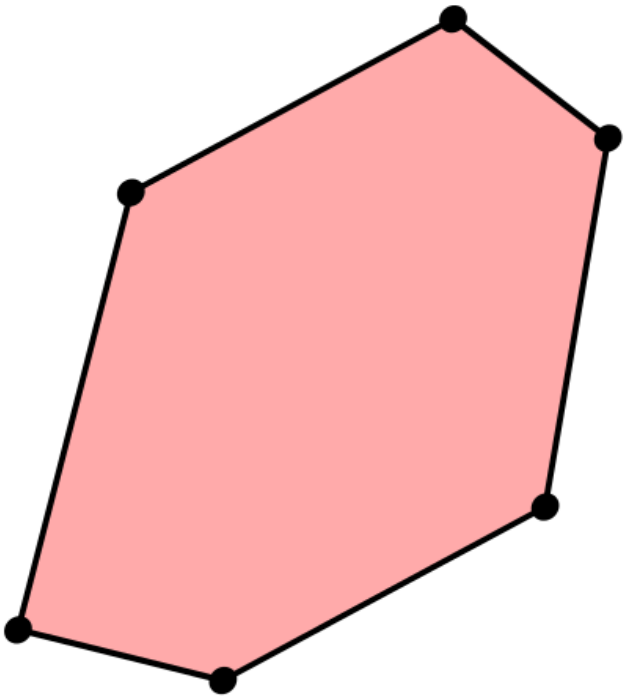
Probabilistic Software Analysis

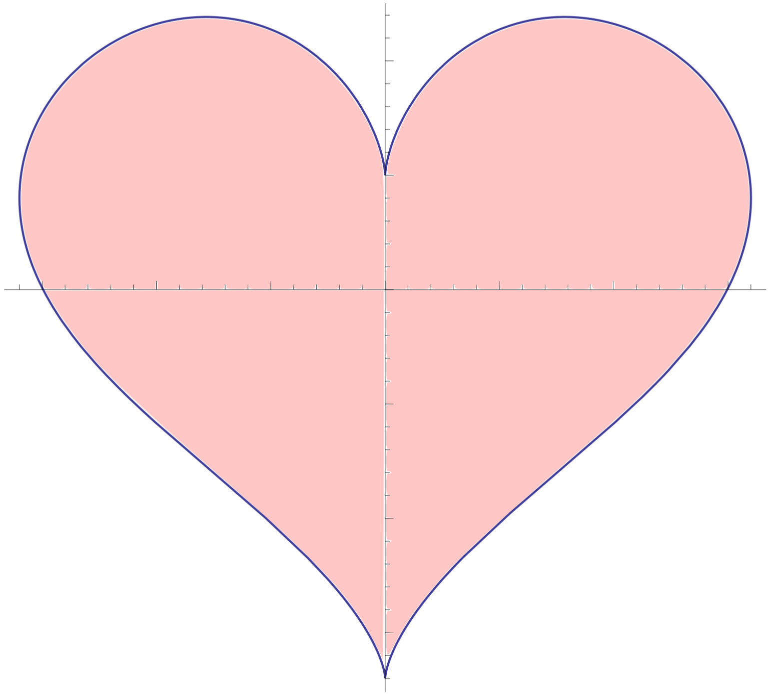


Probabilistic Software Analysis



Obstacle: Quantification





Integration Methods

Symbolic

➔ very expensive, restricted

Numerical

➔ expensive with multi-dimensional domains

Statistical

➔ approximate results

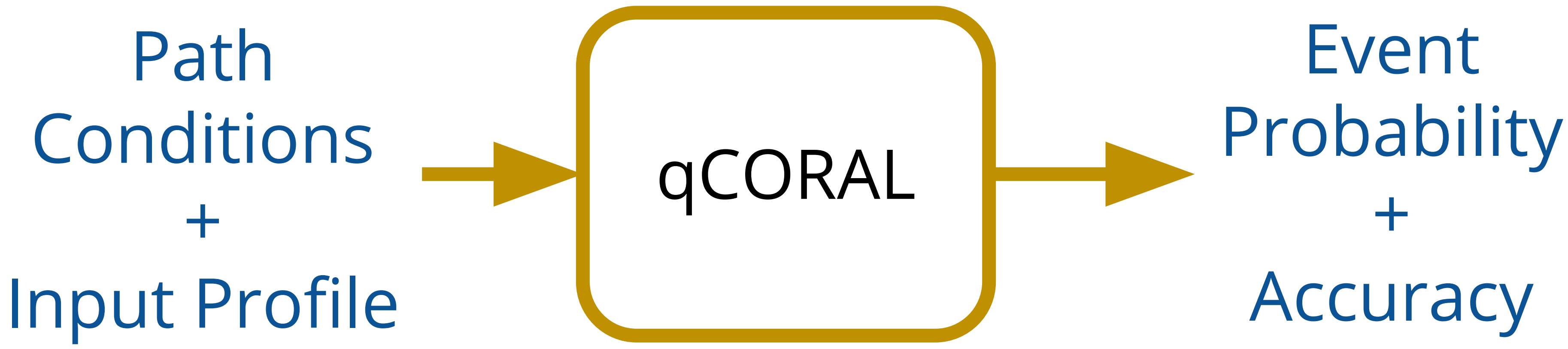
Challenge

Quantifying the solution space of complex mathematical functions

Example constraint from TSAFE module (Tactical Separation Assisted Flight Environment)

sqrt(pow(((x1 + (e1 * (cos(x4) – cos((x4 + (((1.0 * (((c1 * x5) * (e2/c2)) / x6)) * x2) / e1)))))) – (((e2/c2)) * (1.0 – cos((c1 * x5))))), 2.0)) > 999.0 & (c1 * x5) > 0.0 & x3 > 0.0 & x6 > 0.0 & c1 = 0.017... & c2 = 68443.0 & e1 = ((pow(x2,2.0) / tan((c1*x3)))/c2) & e2 = pow(x6,2.0) / tan (c1*x3)

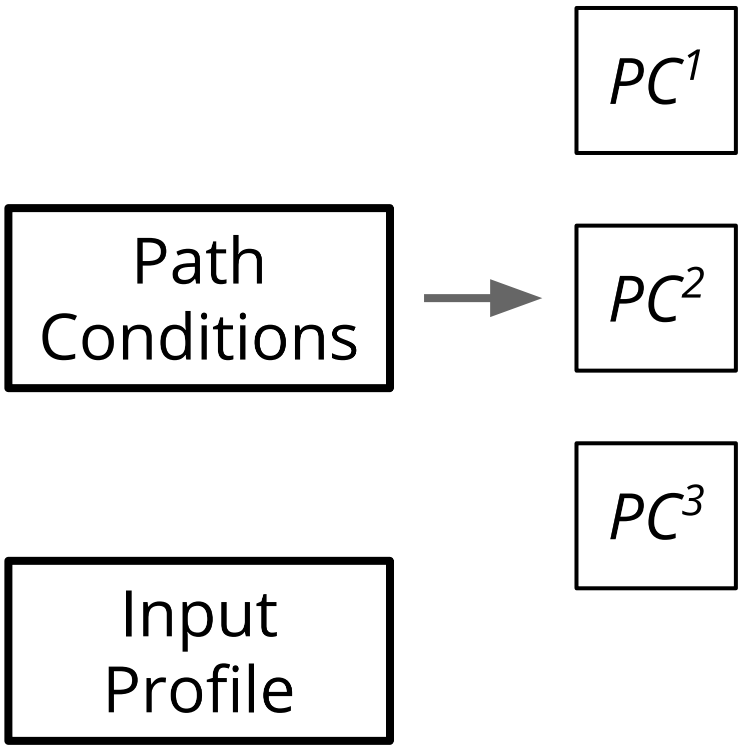
Contribution



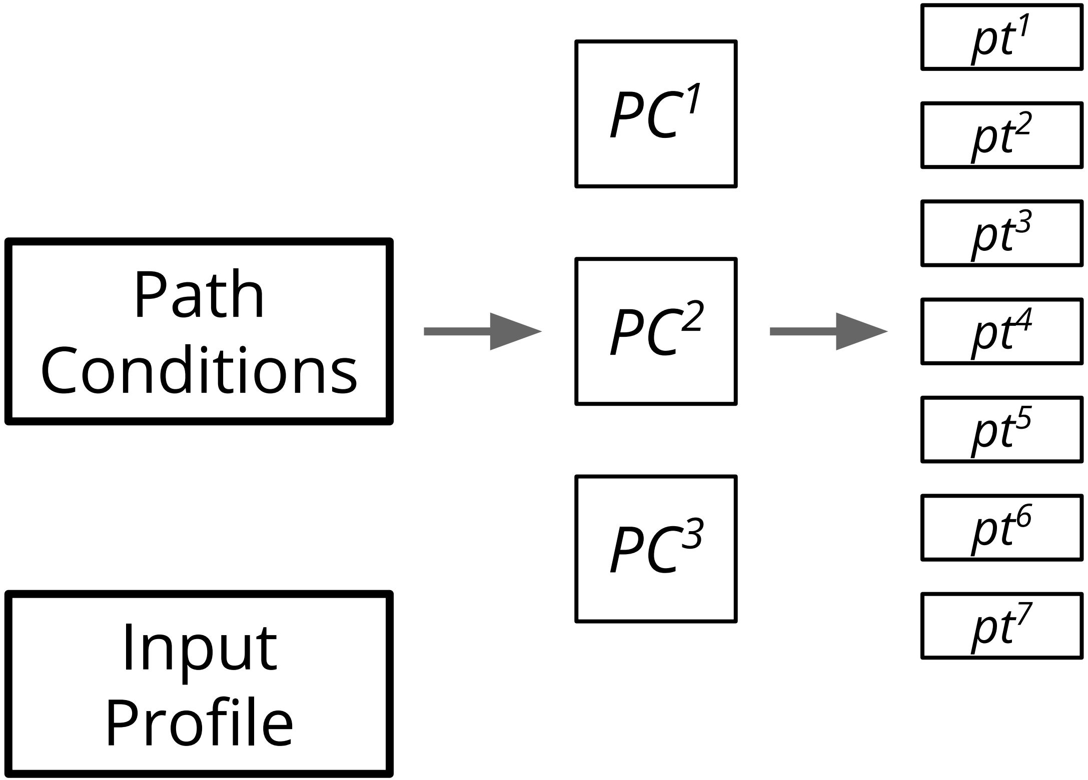
Supports arbitrarily complex constraints

Computes accurate estimates efficiently

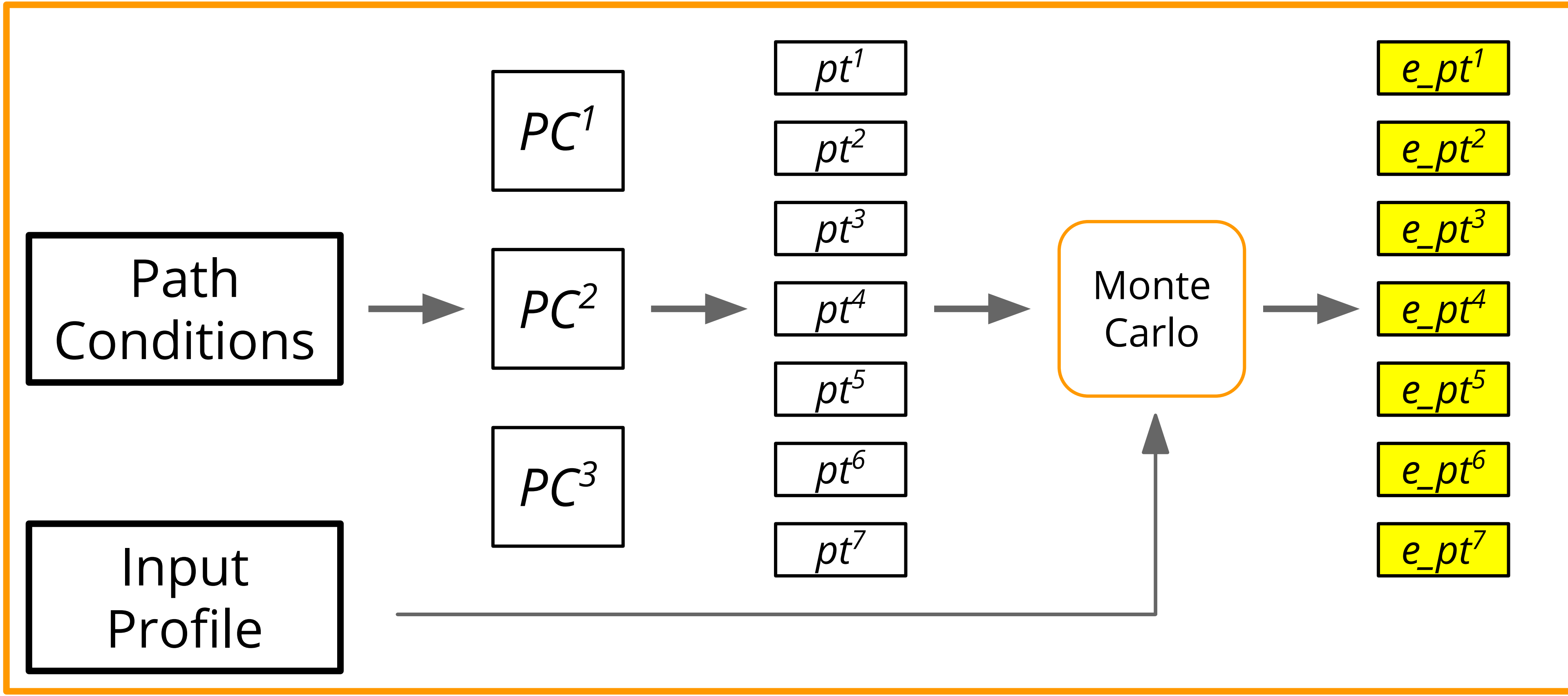
High Level View: Divide



High Level View: Divide



High Level View: Divide



High Level View: Conquer

e_pt^1

e_pt^2

e_pt^3

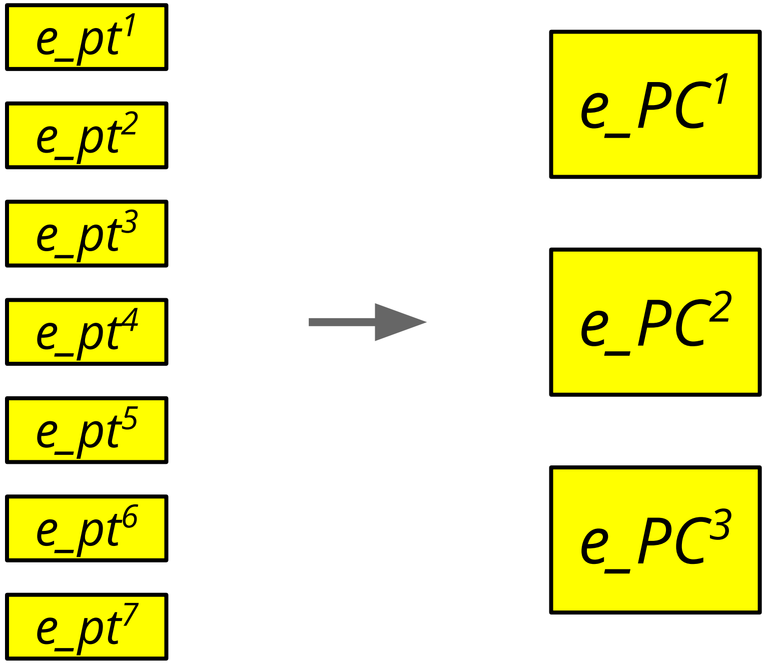
e_pt^4

e_pt^5

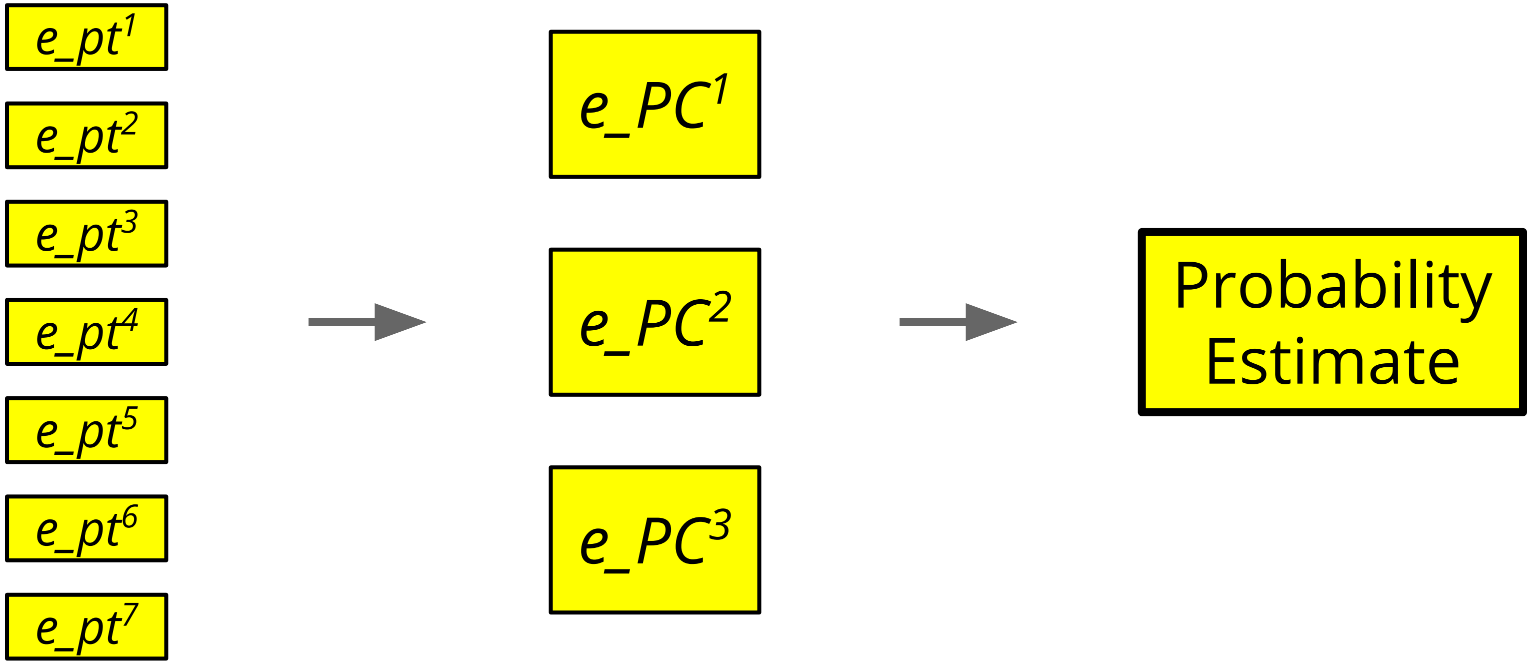
e_pt^6

e_pt^7

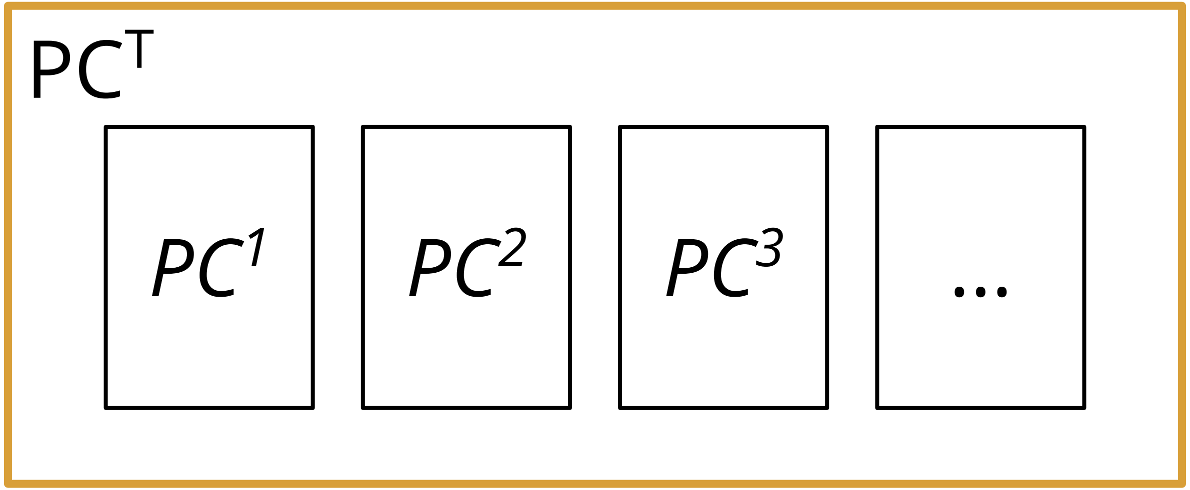
High Level View: Conquer



High Level View: Conquer



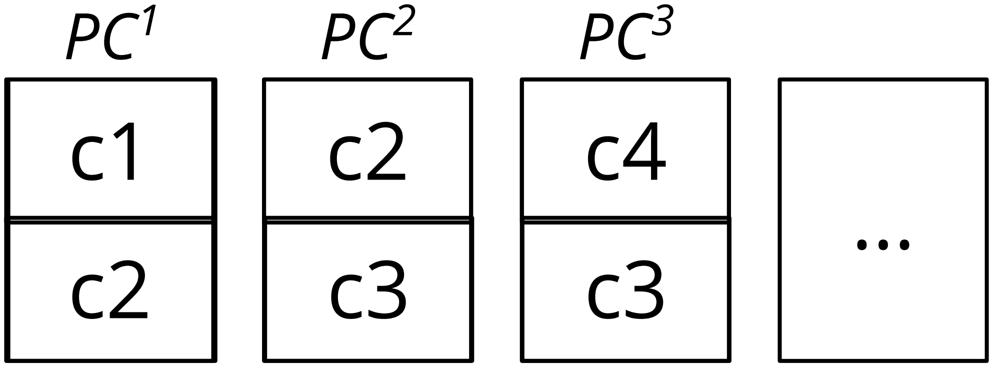
Working With Disjunctions



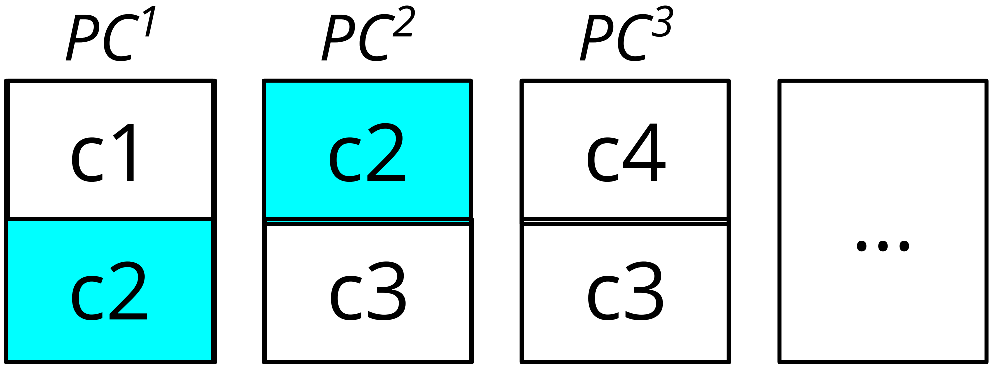
All elements in PC^T are disjoint

Estimates can be computed individually

Working With Conjunctions

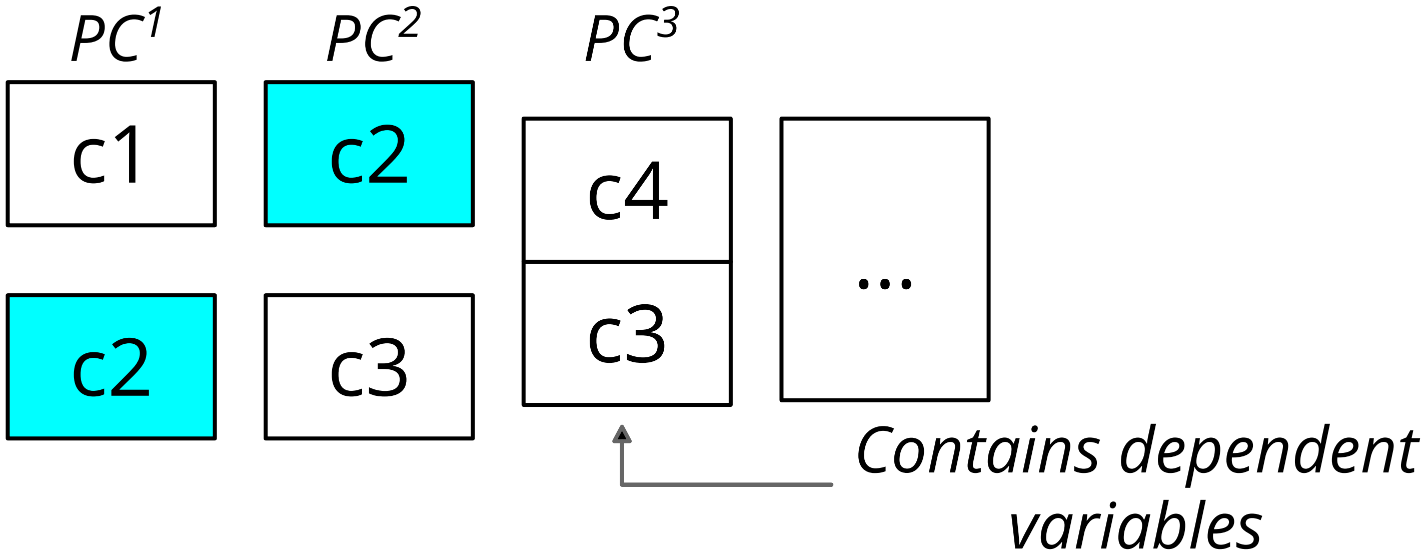


Working With Conjunctions



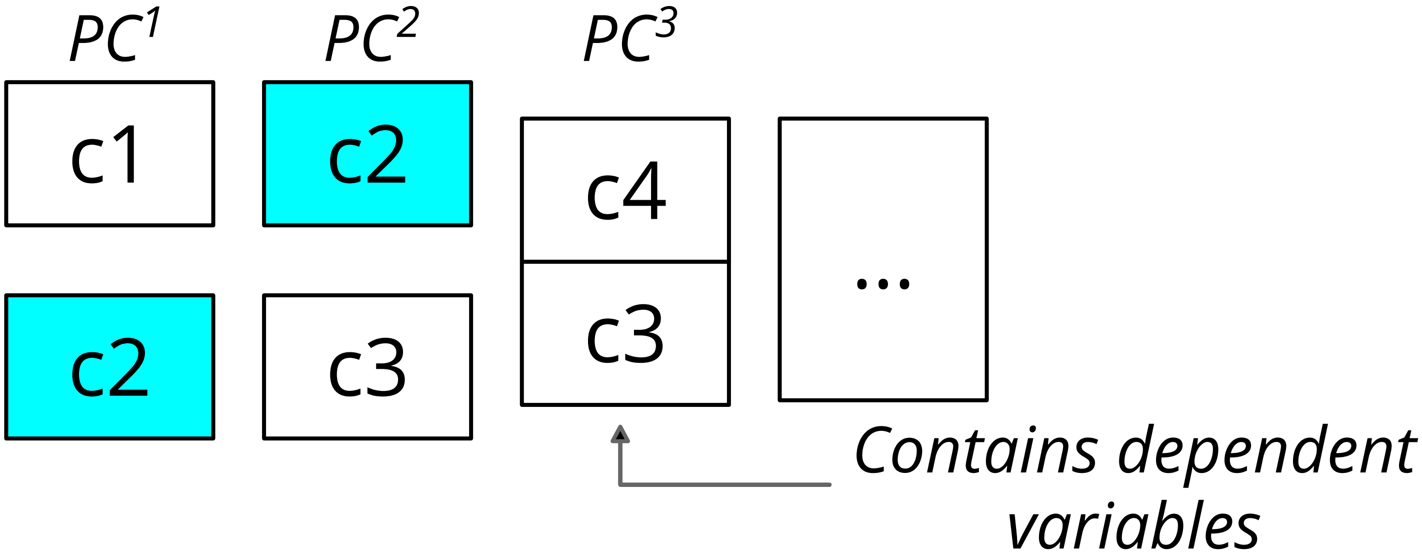
Working With Conjunctions

```
pt1: c1
pt2: c2
pt3: c3
pt4: c3 && c4
```



Working With Conjunctions

```
pt1: c1
pt2: c2
pt3: c3
pt4: c3 && c4
```



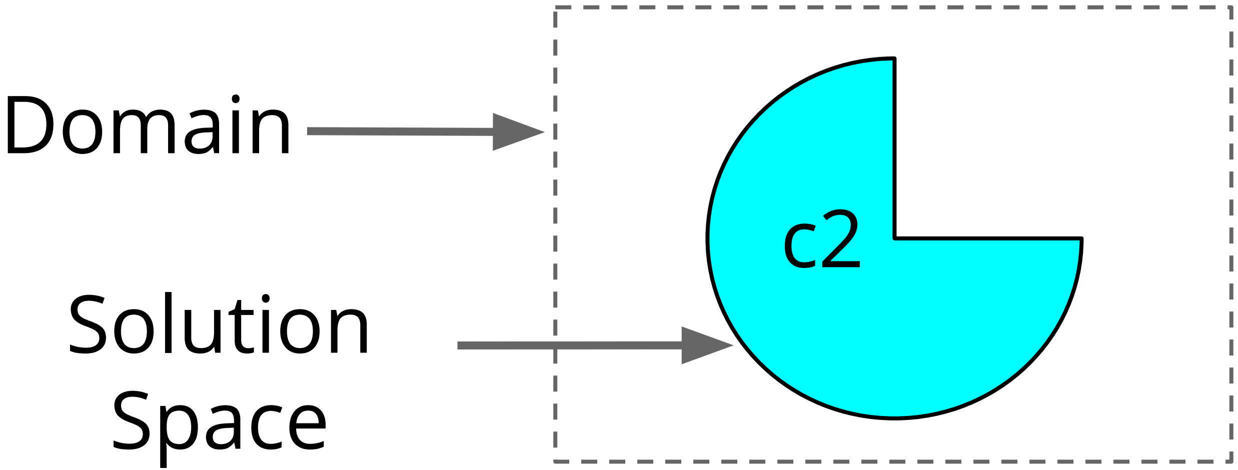
Partitions can be analyzed faster

Estimates can be efficiently re-used

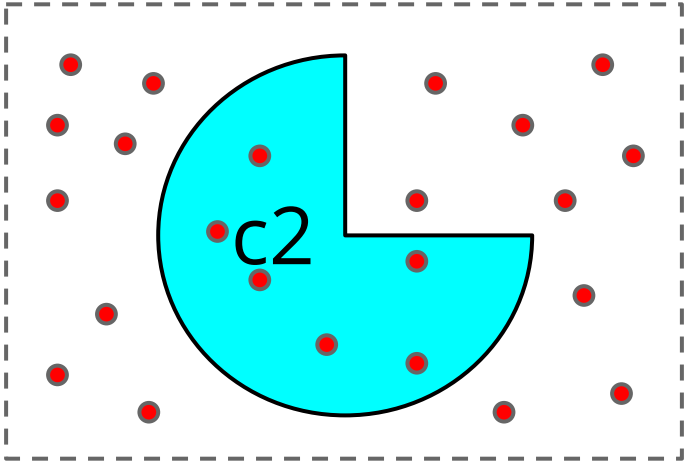
Quantifying Constraints

c2

Quantifying Constraints

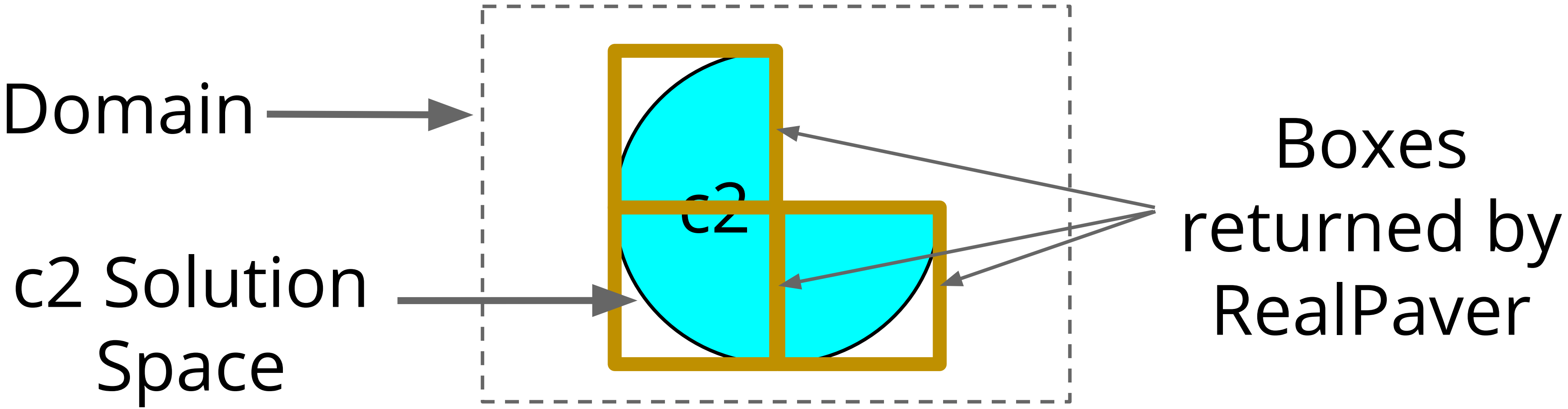


Hit-or-Miss Monte Carlo



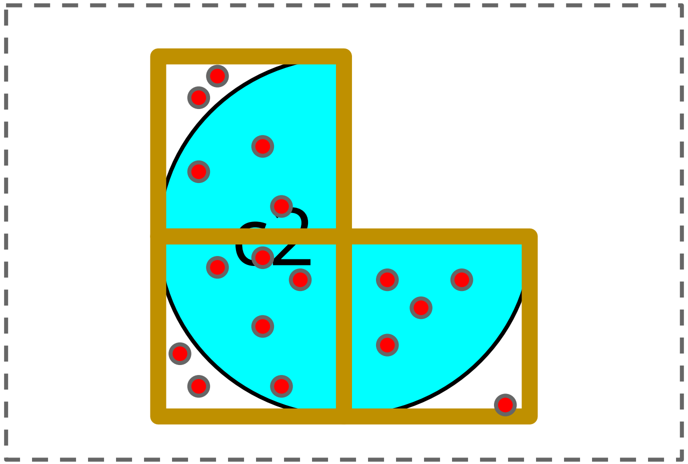
E[X] = #hits / #samples

Stratified Sampling



Remove infeasible areas with RealPaver

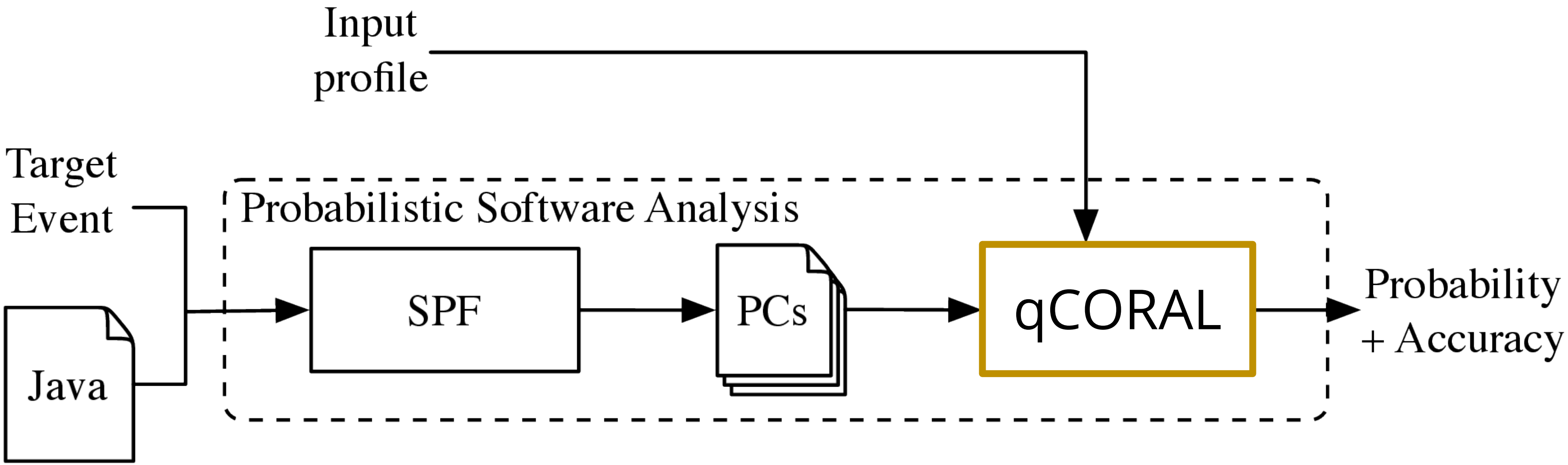
Stratified Sampling



Remove infeasible areas with RealPaver

Increase precision with Stratified Sampling

SPF Toolchain (with qCORAL)



Illustrative Example

```
// 0 <= x,y,z <= 9
f(x,y,z):
  if x < 5:
    if y < 3:
      abort()
    elif z + y > 10:
      abort()
```

Probability that f(x,y,z) calls abort()?

Illustrative Example

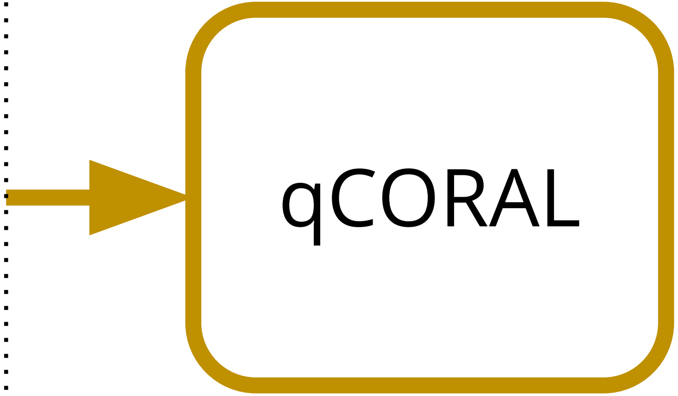
```
// 0 <= x,y,z <= 9
f(x,y,z):
  if x < 5:
    if y < 3:
      abort()
    elif z + y > 10:
     abort()
```

Probability that
f(x,y,z) calls abort()?

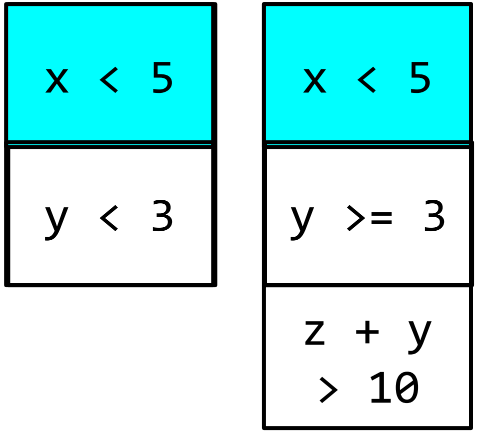
```
pc1: x < 5 && y < 3
pc2: x < 5 && y >= 3
  && z + y > 10
```

Illustrative Example

```
//0 <= x,y,z <= 9
pc1: x < 5
  && y < 3
pc2: x < 5
  && y >= 3
  && z + y > 10
```

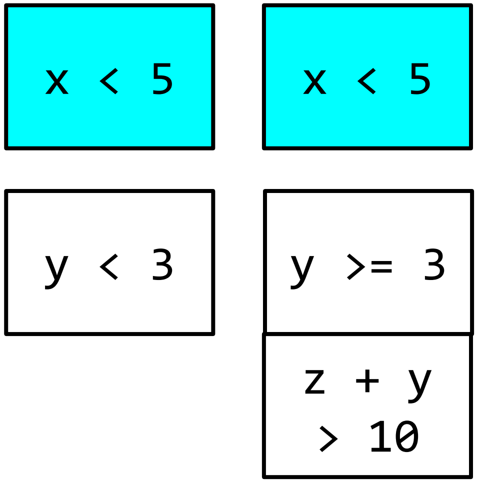


Illustrative Example




```
pc1: x < 5 && y < 3
pc2: x < 5 && y >= 3 && z + y > 10
```

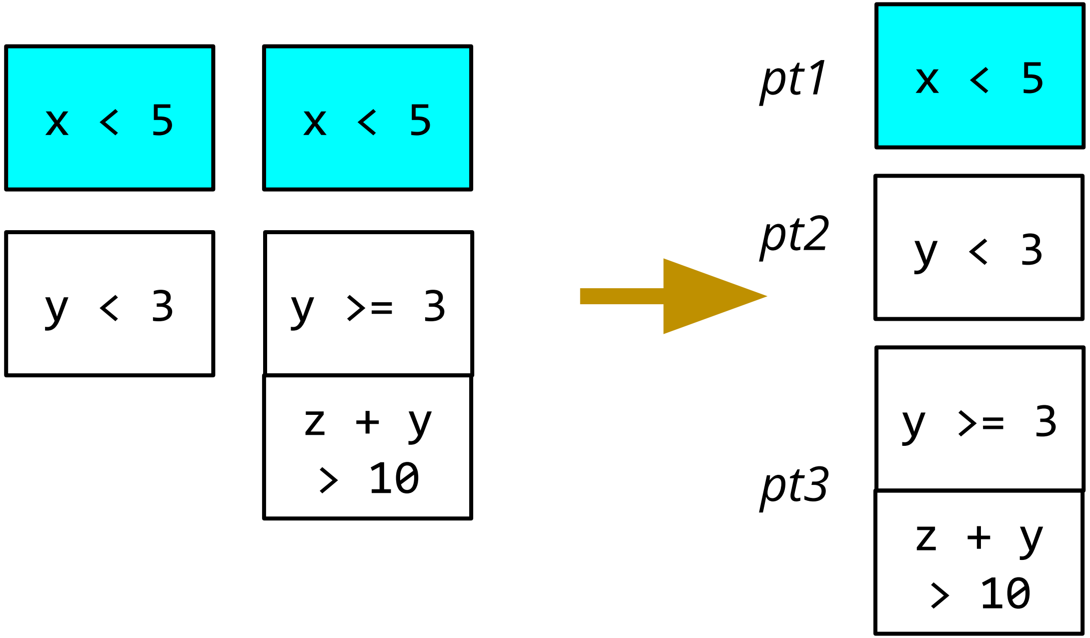
Illustrative Example



pc1: x < 5 && y < 3

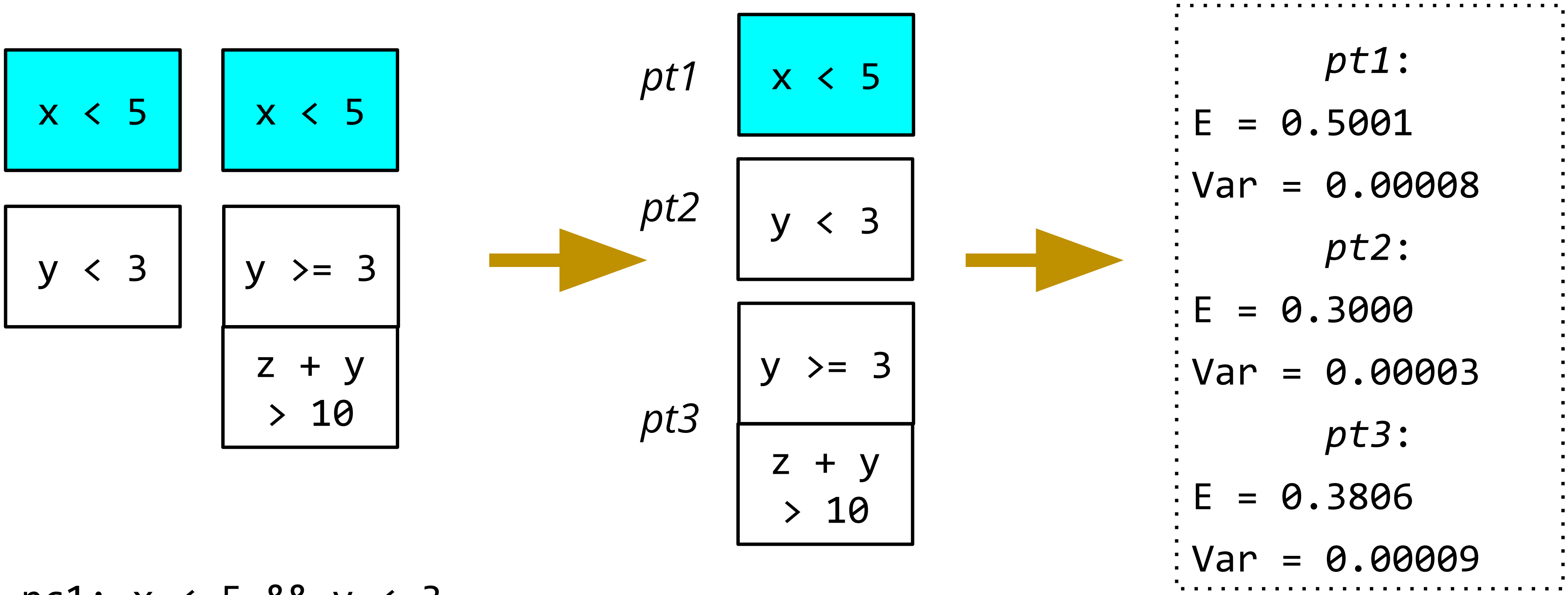
pc2: x < 5 && y >= 3 && z + y > 10

Illustrative Example




```
pc1: x < 5 && y < 3
pc2: x < 5 && y >= 3 && z + y > 10
```

Illustrative Example




```
pc1: x < 5 && y < 3
pc2: x < 5 && y >= 3 && z + y > 10
```

Illustrative Example

pt1:

E = 0.5001

Var = 0.00008

pt2:

E = 0.3000

Var = 0.00003

pt3:

E = 0.3806

Var = 0.00009

```
pc1: x < 5 && y < 3
pc2: x < 5 && y >= 3 && z + y > 10
```

Illustrative Example

pt1:

E = 0.5001

Var = 0.00008

pt2:

E = 0.3000

Var = 0.00003

pt3:

E = 0.3806

Var = 0.00009

pc1:

E = 0.1501

Var = 0.00013

pc2:

E = 0.1927

Var = 0.00022

```
pc1: x < 5 && y < 3
pc2: x < 5 && y >= 3 && z + y > 10
```

Illustrative Example

pt1:
E = 0.5001
Var = 0.00008
pt2:
E = 0.3000
Var = 0.00003
pt3:
E = 0.3806
Var = 0.00009

→

pc1:
E = 0.1501
Var = 0.00013
pc2:
E = 0.1927
Var = 0.00022

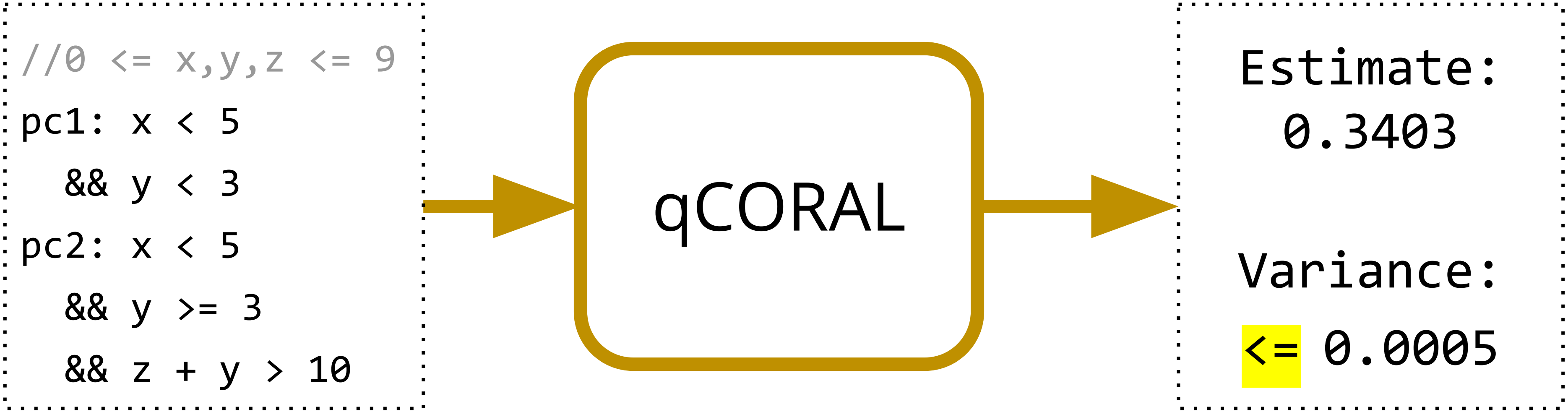
→

Estimate:
0.3403

Variance:
<= 0.0005

pc1: x < 5 && y < 3
pc2: x < 5 && y >= 3 && z + y > 10

Illustrative Example



Evaluation

RQ1: qCORAL is competitive with other tools?

RQ2: qCORAL features help with complex constraints?

RQ1: qCORAL is competitive?

VolComp Benchmark (PLDI'13)

Techniques/Tools:

- ➔ Mathematica (*NIntegrate*)
- ➔ VolComp
- ➔ qCORAL

RQ1: qCORAL is competitive?

VolComp Benchmark (PLDI’13)

Techniques/Tools:

- ➔ Mathematica (*NIntegrate*) ← Baseline
- ➔ VolComp
- ➔ qCORAL

RQ1: qCORAL is competitive?

	NIntegrate	VolComp	qCORAL	
	solution	*bounds*	*avg. est.*	*avg.* **σ**
ARTRIAL	0.9350	[0.9340, 0.9364]	0.9352	1.63e-04
CART	0.9826	[0.9470, 1.0000]	0.9818	1.11e-02
CORONARY	0.0001	[0.0001, 0.0001]	0.0001	4.29e-07
EGFR-EPI	0.1264	[0.1264, 0.1264]	0.1262	3.29e-04
PACK	0.2462	[0.2522, 0.2800]	0.2663	2.72e-05
VOL	1.0005	[0.0000, 1.0000]	1.0001	5.18e-03

RQ1: qCORAL is competitive?

	NIntegrate	VolComp	qCORAL	
	solution	*bounds*	*avg. est.*	*avg.* $\boldsymbol{\sigma}$
ARTRIAL	0.9350	[0.9340, 0.9364]	0.9352	1.63e-04
CART	0.9826	[0.9470, 1.0000]	0.9818	1.11e-02
CORONARY	0.0001	[0.0001, 0.0001]	0.0001	4.29e-07
EGFR-EPI	0.1264	[0.1264, 0.1264]	0.1262	3.29e-04
PACK	0.2462	[0.2522, 0.2800]	0.2663	2.72e-05
VOL	1.0005	[0.0000, 1.0000]	1.0001	5.18e-03

RQ1: qCORAL is competitive?

	NIntegrate	VolComp	qCORAL	
	solution	*bounds*	*avg. est.*	*avg.* $\boldsymbol{\sigma}$
ARTRIAL	0.9350	[0.9340, 0.9364]	0.9352	1.63e-04
CART	0.9826	[0.9470, 1.0000]	0.9818	1.11e-02
CORONARY	0.0001	[0.0001, 0.0001]	0.0001	4.29e-07
EGFR-EPI	0.1264	[0.1264, 0.1264]	0.1262	3.29e-04
PACK	0.2462	[0.2522, 0.2800]	0.2663	2.72e-05
VOL	1.0005	[0.0000, 1.0000]	1.0001	5.18e-03

RQ1: qCORAL is competitive?

	NIntegrate	VolComp	qCORAL
	time	*time*	*avg. time*
ARTRIAL	4,179.36	771.10	4.14
CART	7.66	33.74	4.39
CORONARY	0.86	1.99	0.57
EGFR EPI	1.98	0.60	1.61
PACK	5,066.20	104.80	68.79
VOL	1,245.30	3.76	821.11

RQ1: Observations

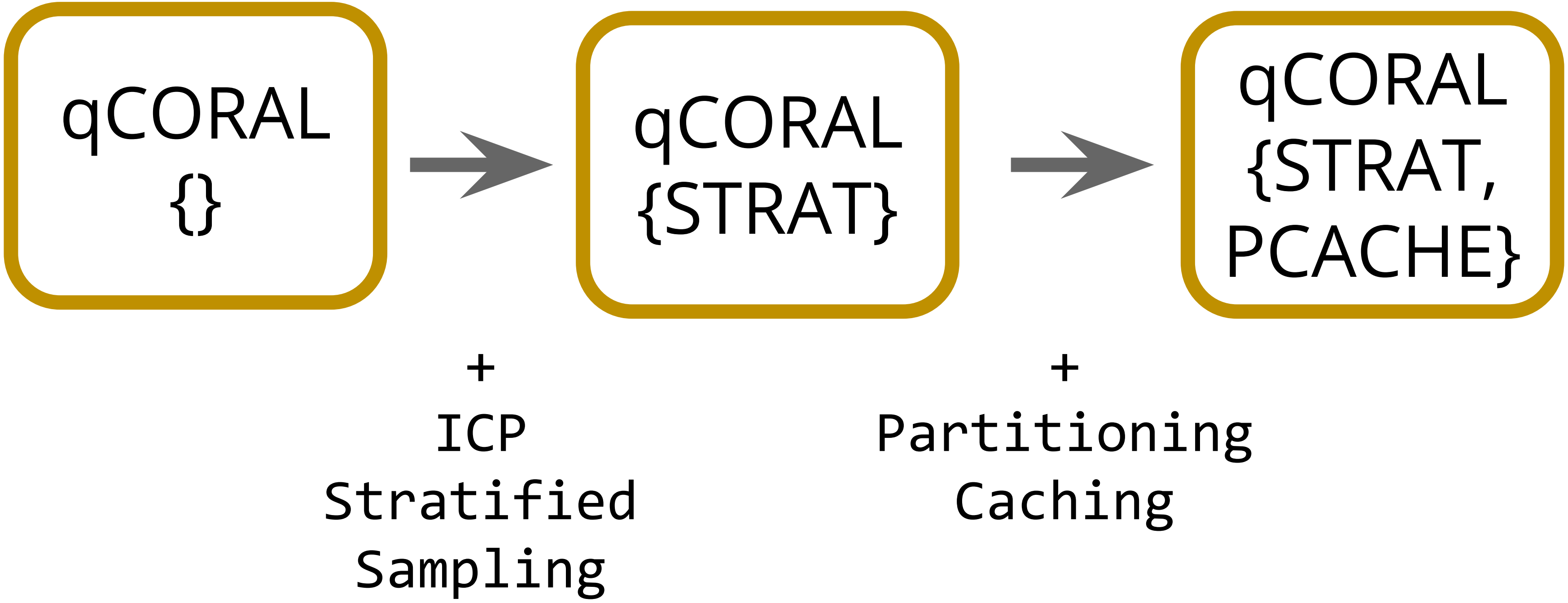
qCORAL estimates:

- ➔ are very close to the results reported by NIntegrate
- ➔ almost always fall within the VolComp interval

RQ2: Evaluation

- ➔ Subjects from the aerospace domain
- ➔ Picked 70% of the paths to avoid bias
- ➔ Reported results for 30 executions (avg. estimate and standard error)

RQ2: Evaluated configurations



RQ2: Subjects Considered

Subject	LOC	#pcs analyzed **(70%)**	complex functions
Apollo	~2,600	5,779	`sqrt`
TSAFE - Conflict	~50	23	`cos,pow, sin, sqrt,tan`
TSAFE - Turnlogic	~50	225	`atan2`

RQ2: Conclusions

Impact of features depends on the subject

{STRAT} can reduce variance (*x50* in Conflict)

➔ There is a time overhead, however

{PCACHE} can reduce time (*x2* in Apollo)

➔ Savings increase with number of samples

(Most Recent) Related Work

Sankaranarayanan *et al.* (PLDI’13)

→ Supports only linear constraints

Adje *et al.* (VSTTE’13)

→ Supports only the four basic arithmetic operations

Conclusions

qCORAL

New approach to solution space quantification

Acceleration procedure improves accuracy

More details at pan.cin.ufpe.br/qcoral

Extra Slides

Probability of a Target Event

P(*ever*

P(*path*

ies of the
e event

solution
domain

And if the number of paths is infinite?

Bound the symbolic execution and measure the confidence!

(see Filieri et al, ICSE 2013)

And the Variance?

Use Chebyshev's inequality:

"…at least 1 - $1/k^2$ of the distribution's values are within *k* standard deviations of the mean"

Target application

Sometimes knowing only if an event happens is not very useful!

- ➔ randomized behavior
- ➔ probabilistic profile of the environment