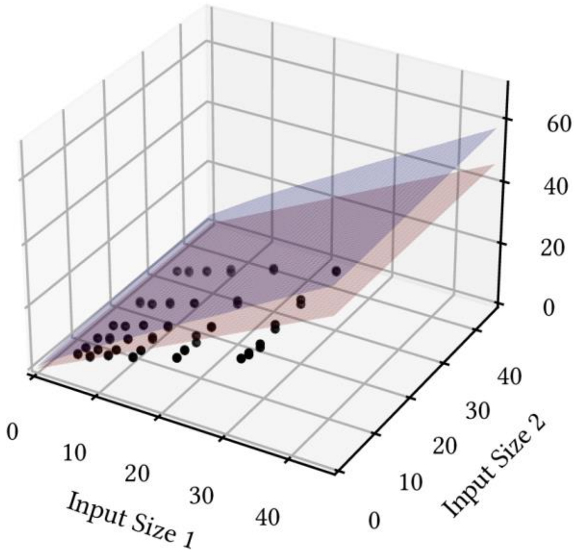
Robust Resource Bounds with Static Analysis and Bayesian Inference



Long Pham, Feras Saad, Jan Hoffmann
Carnegie Mellon University

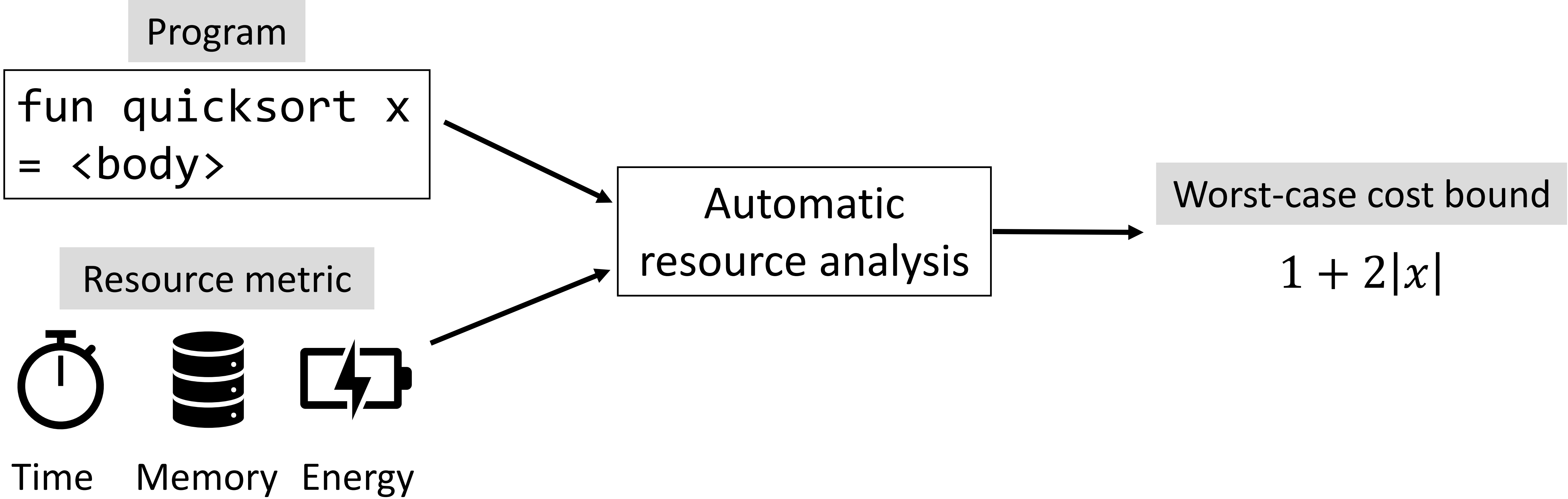
PLDI 2024

June 28, 2024

Resource Analysis

Goal of resource analysis:

Infer a worst-case bound of the cost of a program as a function of inputs



Applications of Resource Analysis

1. Detect algorithmic complexity attacks by inferring worst-case resource usage / inputs

2. Estimate job size for job scheduling in cloud computing

3. **Infer** tool used at Meta/Facebook

https://github.com/facebook/infer

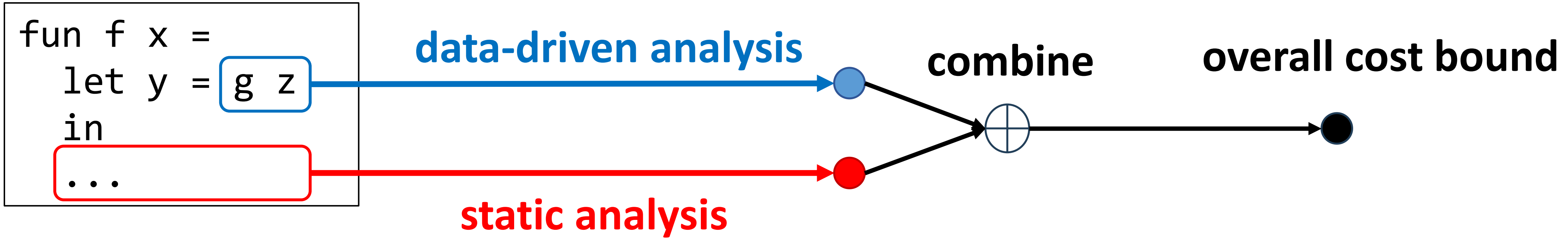
4. Worst-case execution time (WCET) of safety-critical embedded systems

Contribution: Hybrid Resource Analysis

Static analysis of the source code
+ Sound: any result is a valid bound
- Incomplete: cannot handle all programs

Data-driven analysis of runtime data
+ Always returns a result
- No soundness guarantee

Our contribution:
Integrate static analysis and data-driven analysis to combine their complementary strengths and mitigate their respective weaknesses



Outline

- ☑ Motivation for Hybrid Resource Analysis
- ❑ State-of-the-Art Resource Analysis
  - Static Analysis
  - Optimization-Based Data-Driven Analysis
- ❑ Contribution 1: Bayesian Data-Driven Analysis
- ❑ Contribution 2: Hybrid Analysis
- ❑ Evaluation

State of the Art in Static Analysis

Static analysis examines the source code, constructs constraints defining the worst-case behavior, and solves them

- Type systems (e.g., AARA by Hoffmann, Hofmann, Jost et al.)
- Recurrence relations (e.g., COSTA by Albert et al.)
- Ranking functions (e.g., AProVE and KoAT by Giesl et al.)

Advantage:

+ Soundness guarantee

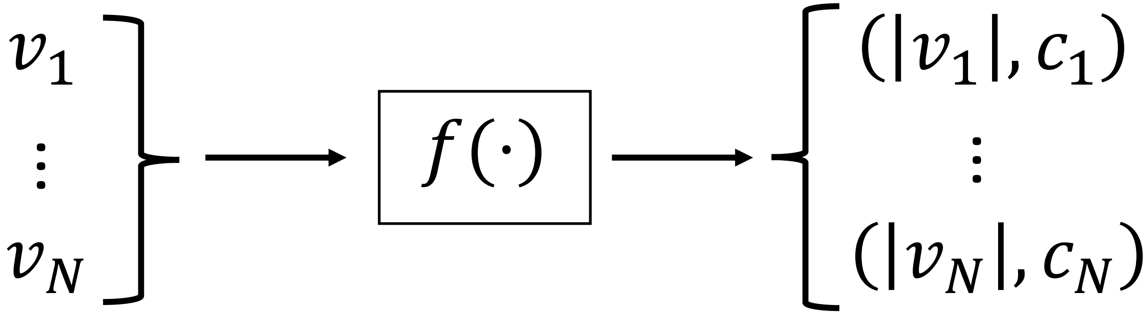
Disadvantages:

- Incomplete due to the undecidability of resource analysis
- Rewriting a program is difficult for non-expert users

State of the Art in Data-Driven Analysis (Optimization)

Examples: Input-sensitive profiling (Coppa et al.), Algorithmic profiling (Zaparanuks et al.), Dynaplex (Ishimwe et al.)

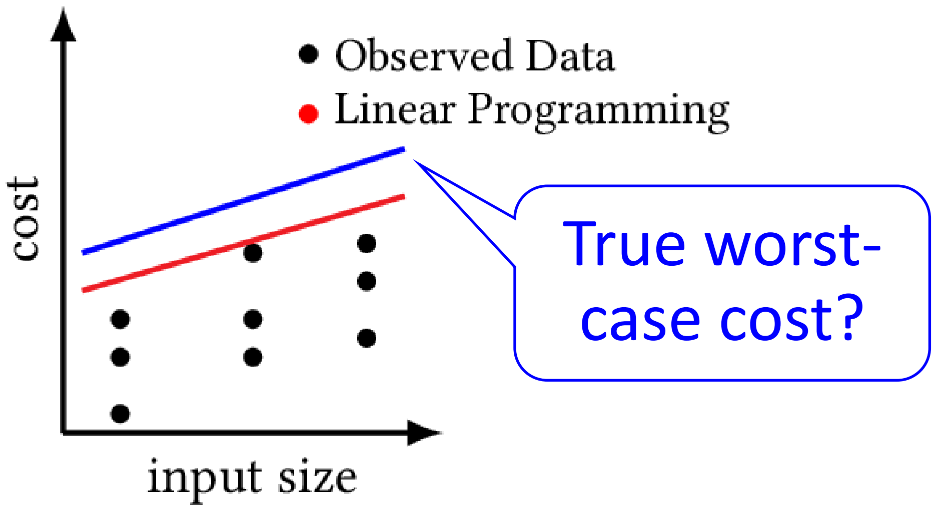
1. Collect cost measurements of inputs $v_1, \ldots, v_N$



2. Optimize cost bound (red line)

Minimize red line − black dots

Subject to red line ≥ black dots



Disadvantages of optimization:

- Does not incorporate the user's domain knowledge
- No quantitative measure of statistical uncertainty

Contribution: Bayesian Data-Driven Analysis

Our contribution: Bayesian data-driven resource analysis

1. Define a probabilistic model $\pi(\theta, D)$
   θ: latent parameter (cost bound)
   D: observed data (cost measurements)

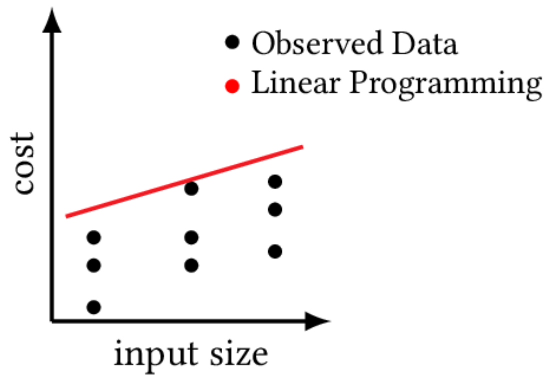
2. Collect observed data D_{obs}

3. Compute/approximate the posterior distribution
   Bayes' rule: $\pi(\theta \mid D = D_{\text{obs}}) = \frac{\pi(\theta, D{=}D_{\text{obs}})}{\int \pi(\theta, D{=}D_{\text{obs}})\,\mathrm{d}\theta}$
   Draw posterior samples: $\theta_1, \ldots, \theta_M \sim \pi(\theta \mid D = D_{\text{obs}})$

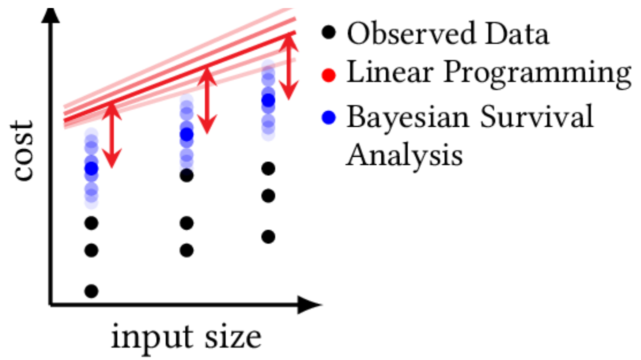
Advantages over optimization:
+ Can incorporate the domain knowledge in the probabilistic model
+ Posterior distribution captures statistical uncertainty

Data-Driven Analysis: Overview

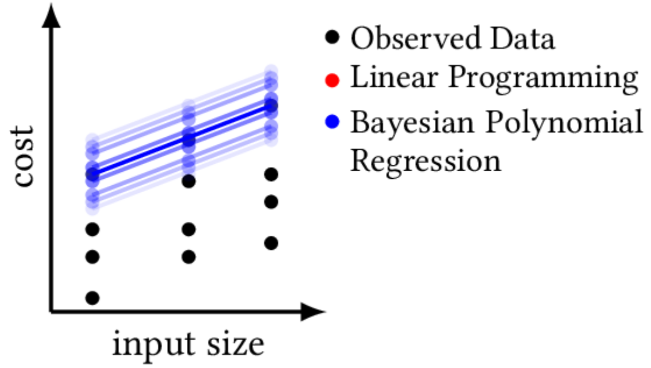
Previous: Optimization (Opt)



New: Bayesian inference of worst-case costs (BayesWC)



New: Bayesian inference of polynomial coefficients (BayesPC)



Data-Driven Analysis via BayesWC

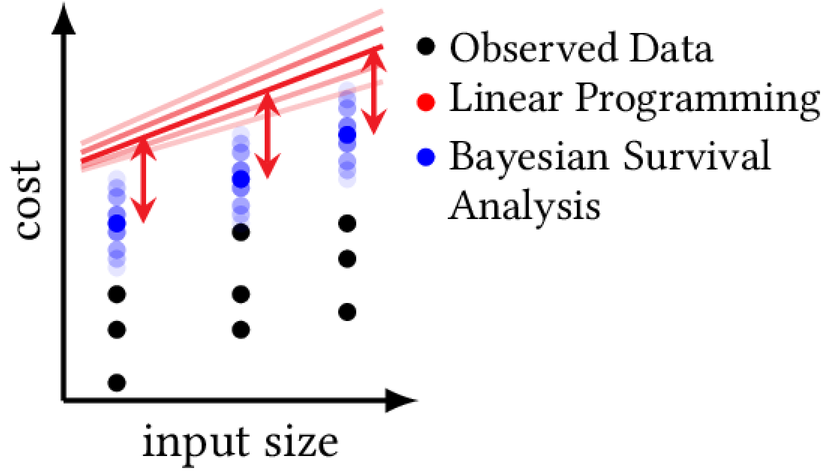
Bayesian inference of worst-case costs (BayesWC)

1. Define a probabilistic model

$$\pi(\theta, \boldsymbol{c}^{\max}, \boldsymbol{c})$$

$\boldsymbol{c}^{\max}$: hidden worst-case cost (blue dots)

$\boldsymbol{c}$: observed cost (black dots)



2. Draw posterior samples of $\boldsymbol{c}^{\max}$

3. Optimize cost bound (red line)

Minimize $\quad$ red line − blue dots

Subject to $\quad$ red line ≥ blue dots

Data-Driven Analysis via BayesPC

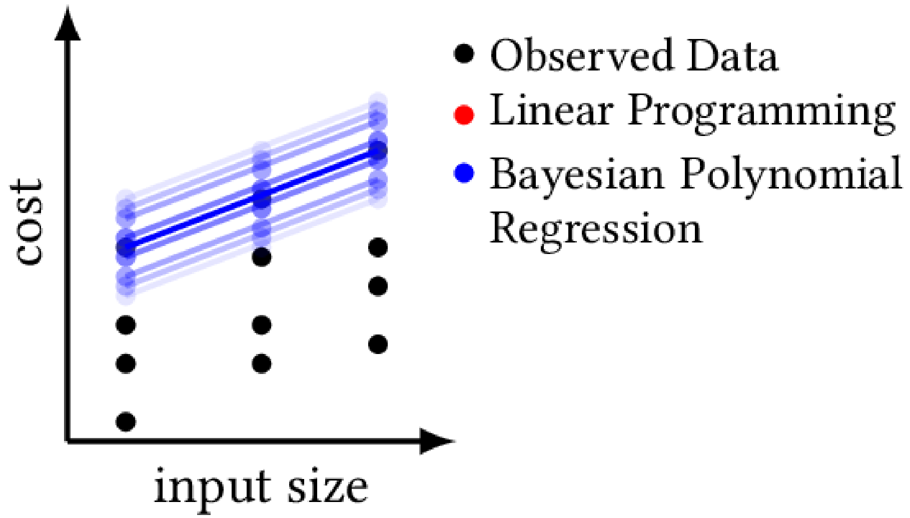
Bayesian inference of polynomial coefficients (BayesPC)

1. Define a probabilistic model

   $\pi(p, \boldsymbol{c})$

   p: cost bond (blue line)

   $\boldsymbol{c}$: observed cost (black dots)



2. Draw posterior samples of cost bound p (blue line)

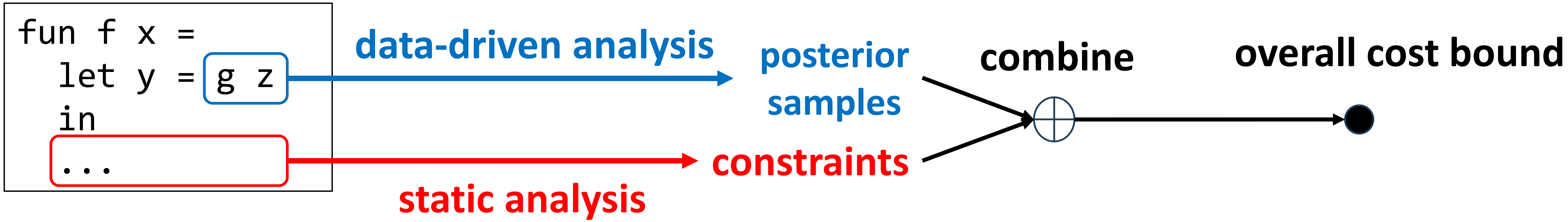
Outline

- ☑ Motivation
- ☑ State-of-the-Art Resource Analysis
  - Static Analysis
  - Optimization-Based Data-Driven Analysis
- ☑ Contribution 1: Bayesian Data-Driven Analysis
- ❑ Contribution 2: Hybrid Resource Analysis
- ❑ Evaluation

Hybrid Analysis: Challenge

Hybrid analysis needs an interface between:

1. Bayesian data-driven analysis draws posterior samples
2. Static analysis solves constraints



Key challenge:

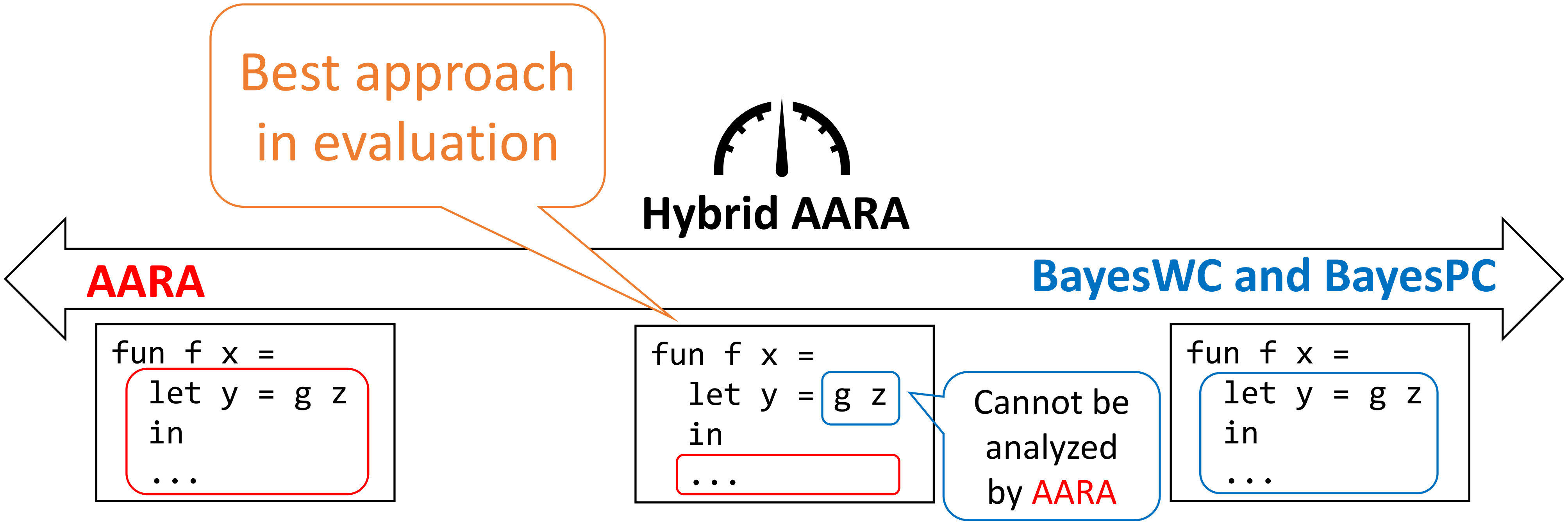
How do we coherently combine constraints and posterior samples?

Contribution: Hybrid AARA

We design, implement, and evaluate Hybrid AARA, which integrates

- Bayesian data-driven analysis (BayesWC and BayesPC) and
- Automatic Amortized Resource Analysis (AARA), a type-based static analysis

by a novel interface between sampling algorithms and linear programming



Static Analysis: AARA

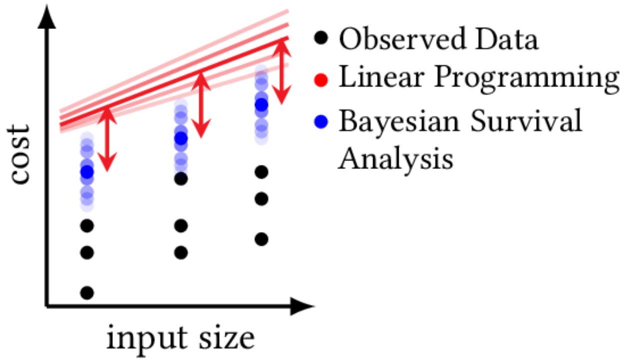
- Each variable is equipped with a polynomial potential function (from amortized analysis of algorithms)

- Infer polynomial coefficients by solving linear programs

Why we choose AARA for static analysis:
+ Compositionality offered by types
+ Automatic bound inference by LP solving
+ Precise cost bounds by amortized analysis
+ Soundness guarantee
+ Cost-bound certificates in the form of type derivations

Hybrid AARA: AARA + BayesWC

Combine the linear constraints of AARA and BayesWC



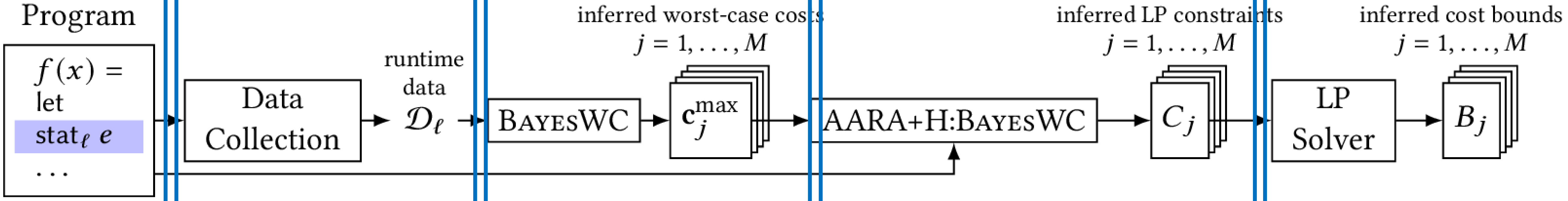
Annotate stat e

1. Collect runtime data D

2. Draw posterior samples $\boldsymbol{c}_j^{\max}$

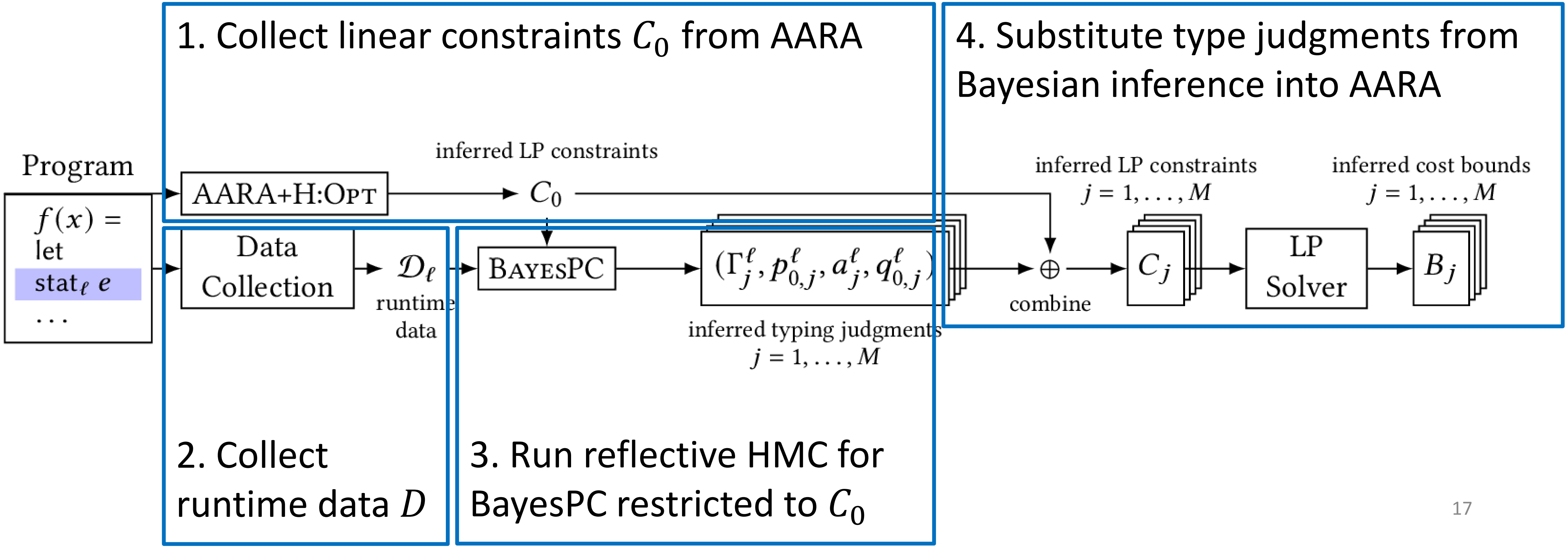
3. Aggregate the linear constraints from AARA and from BayesWC

4. LP solving



Hybrid AARA: AARA + BayesPC

Run reflective Hamiltonian Monte Carlo (Chalkis et al., 2023), which draws samples from a probability distribution within a bounded convex polytope



Example Evaluation for Quicksort

Resource metric: comparisons, each of which varies between 0.5 and 1.0

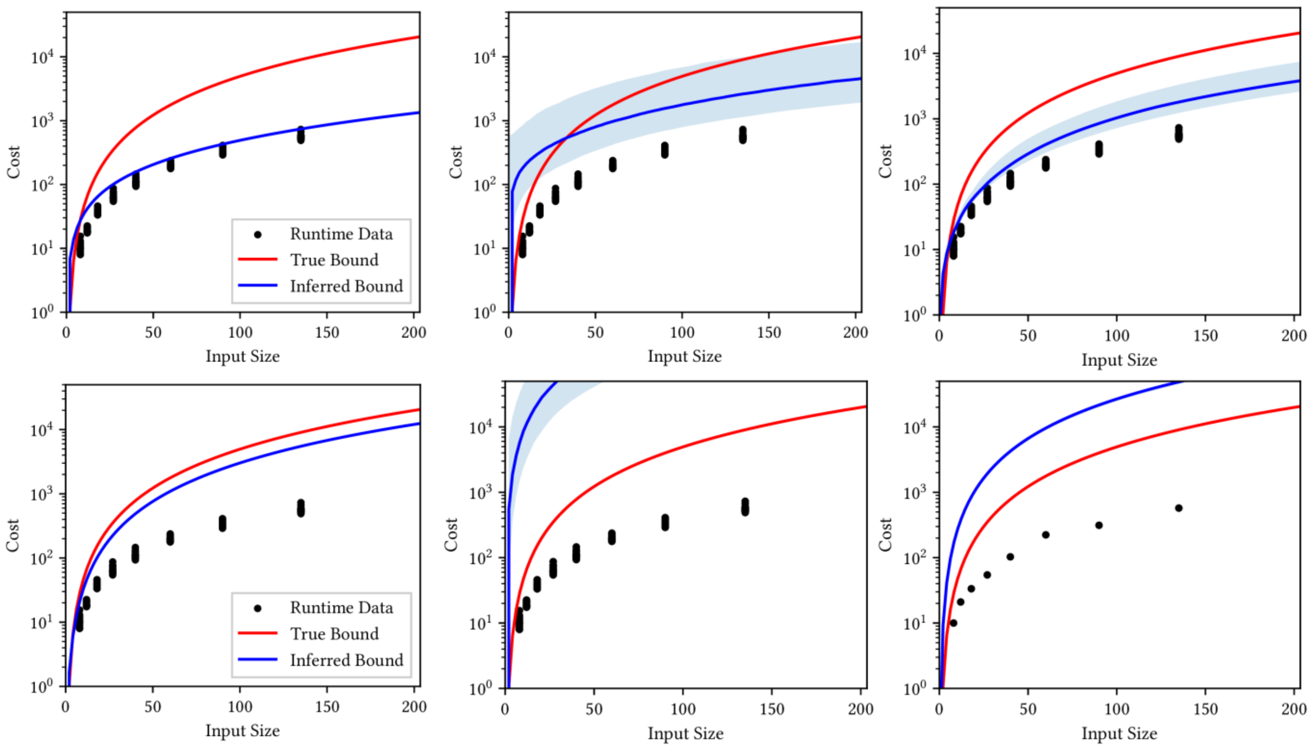
1. Bayesian analysis is more accurate than optimization

2. Hybrid analysis is more accurate than data-driven analysis

Opt | BayesWC | BayesPC

Data-driven

Hybrid



Evaluation: Proportions of Sound Cost Bounds

Table 1. Percentage of inferred cost bounds that are sound and analysis runtime for 10 benchmark programs.

Benchmark Program	Conventional AARA	Analysis Method	Fraction of Sound Inferred Bounds: Data-Driven	Fraction of Sound Inferred Bounds: Hybrid	Analysis Runtime: Data-Driven	Analysis Runtime: Hybrid
MapAppend	*Cannot Analyze*	Opt	0%	0%	0.01 s	0.01 s
		BayesWC	68.5%	100%	1.87 s	12.44 s
		BayesPC	75.5%	100%	51.83 s	360.80 s
Concat	*Cannot Analyze*	Opt	0%	0%	0.00 s	0.01 s
		BayesWC	67.3%	96.7%	2.54 s	14.73 s
		BayesPC	96%	100%	113.53 s	125.28 s
InsertionSort2	*Wrong Degree*	Opt	0%	0%	0.01 s	0.02 s
		BayesWC	57.6%	100%	1.53 s	5.46 s
		BayesPC	21%	57.5%	10.68 s	220.66 s
QuickSort	*Cannot Analyze*	Opt	0%	0%	0.01 s	0.11 s
		BayesWC	4%	96%	2.20 s	144.88 s
		BayesPC	0%	100%	13.72 s	274.51 s
QuickSelect	*Cannot Analyze*	Opt	0%	0%	0.02 s	0.19 s
		BayesWC	0.2%	98.2%	1.83 s	222.47 s
		BayesPC	0%	100%	12.39 s	277.20 s
MedianOfMedians	*Cannot Analyze*	Opt	0%	0%	0.17 s	0.21 s
		BayesWC	11.5%	71.3%	2.36 s	93.89 s
		BayesPC	0%	100%	70.39 s	896.98 s
ZAlgorithm	*Wrong Degree*	Opt	0%	0%	0.09 s	0.13 s
		BayesWC	13.7%	95.9%	1.96 s	72.21 s
		BayesPC	28%	100%	11.11 s	509.29 s
BubbleSort	*Cannot Analyze*	Opt	0%	*Cannot Analyze*	0.01 s	∅
		BayesWC	40.1%	*Cannot Analyze*	2.69 s	∅
		BayesPC	31.5%	*Cannot Analyze*	11.70 s	∅
Round	*Cannot Analyze*	Opt	0%	*Cannot Analyze*	0.01 s	∅
		BayesWC	58.3%	*Cannot Analyze*	1.91 s	∅
		BayesPC	81%	*Cannot Analyze*	12.87 s	∅
EvenOddTail	*Wrong Degree*	Opt	0%	*Wrong Degree*	0.01 s	∅
		BayesWC	65.1%	*Wrong Degree*	1.98 s	∅
		BayesPC	70%	*Wrong Degree*	11.79 s	∅

Bayesian vs Optimization

BayesWC and BayesPC have higher proportions of sound bounds than Opt.

Hybrid vs Data-Driven

Hybrid BayesWC and BayesPC have higher proportions of sound bounds than data-driven BayesWC and BayesPC.

Takeaways

1. Bayesian resource analysis is more robust than the opt-based technique

2. Hybrid resource analysis = static (AARA) + data-driven (Bayesian)
    - is more accurate and robust than data-driven analysis
    - mitigates the incompleteness of static analysis

Further details in the paper:

- Type-based formulation of Hybrid AARA
- Two soundness theorems of Hybrid AARA
- Full experiment results

Static Analysis: AARA

Automatic Amortized Resource Analysis (AARA):
Type-based resource analysis that automates the potential method of amortized analysis

Example: partition function

Cost: one unit per element

Input: [1,2,3,4]

Output: ([1,2], [3,4])

Typing judgment:

partition: $(\text{int} \times L^2(\text{int})) \rightarrow (L^1(\text{int}) \times L^1(\text{int}))$

Input potential: $2 \cdot n$

Output potential: $1 \cdot n_1 + 1 \cdot n\_2$

Static Analysis: AARA

1. Assign variables

$$\text{partition}: (\text{int} \times L^{p}(\text{int})) \to (L^{q_1}(\text{int}) \times L^{q_2}(\text{int}))$$

2. Collect linear constraints

Input potential

$$p \geq 1 + q_1$$
$$p \geq 1 + q_2$$

Output potential

Cost

Sound: any cost bound inferred by AARA is a valid worst-case cost bound

Incomplete: there exists a polynomial-cost program that AARA cannot infer because resource analysis is undecidable in general