Cosette: An Automated Solver for SQL

Shumo
Chu

Konstantin
Weitz

Chenglong
Wang

Alvin
Cheung

Dan
Suciu

`cosette.cs.washington.edu`

```
SELECT ...
FROM   ...
WHERE  ...
```

Q1

```
SELECT ...
FROM   ...
WHERE  ...
```

Q2

$$\forall\, D \,.\, Q1(D) = Q2(D)$$
$$\exists\, D \,.\, Q1(D) \neq Q2(D)$$

?

MySQL

Query Optimizers

edX UDACITY coursera

Autograders

MEMCACHED

Application Caches

Boris Trakhtenbrot

Deciding the equality of two arbitrary relational queries is undecidable.

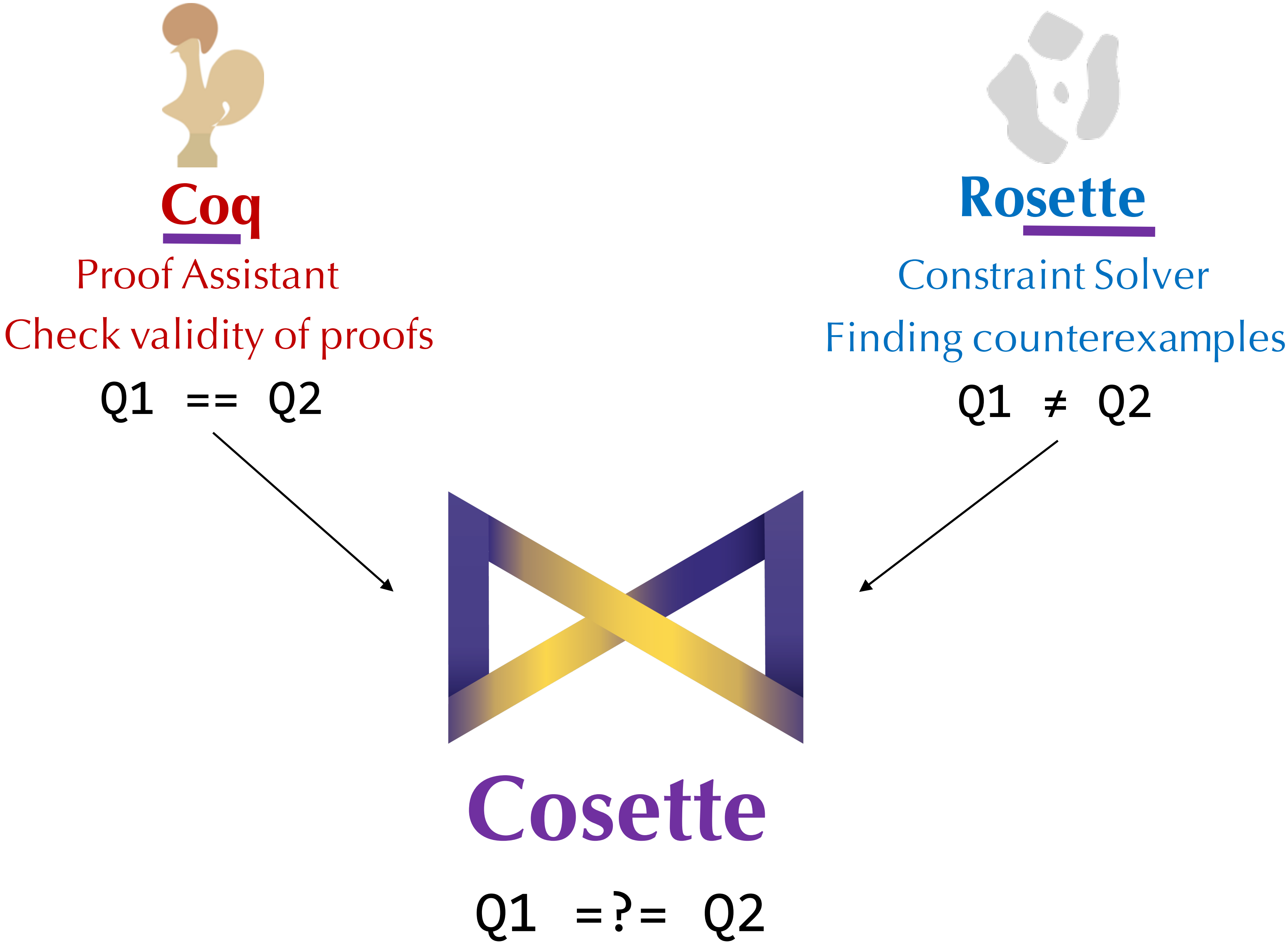
Full decision procedure exists for conjunctive queries

Simple heuristics can already prove many common cases

Operating Systems

Language Compilers

Distributed Algorithms



Coq

Proof Assistant

Check validity of proofs

Q1 == Q2

Rosette

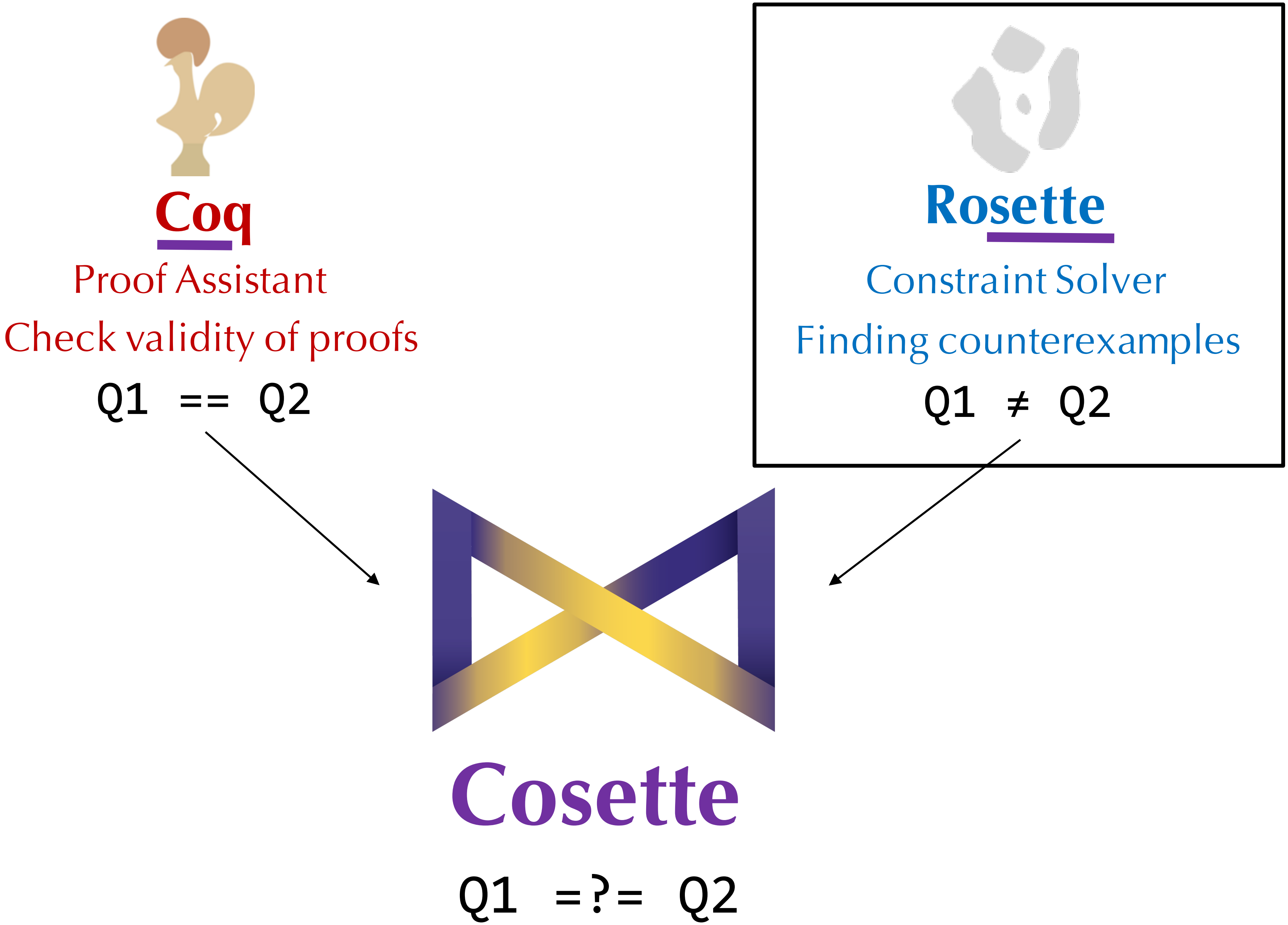
Constraint Solver

Finding counterexamples

Q1 ≠ Q2

Cosette

Q1 =?= Q2



Coq

Proof Assistant

Check validity of proofs

Q1 == Q2

Rosette

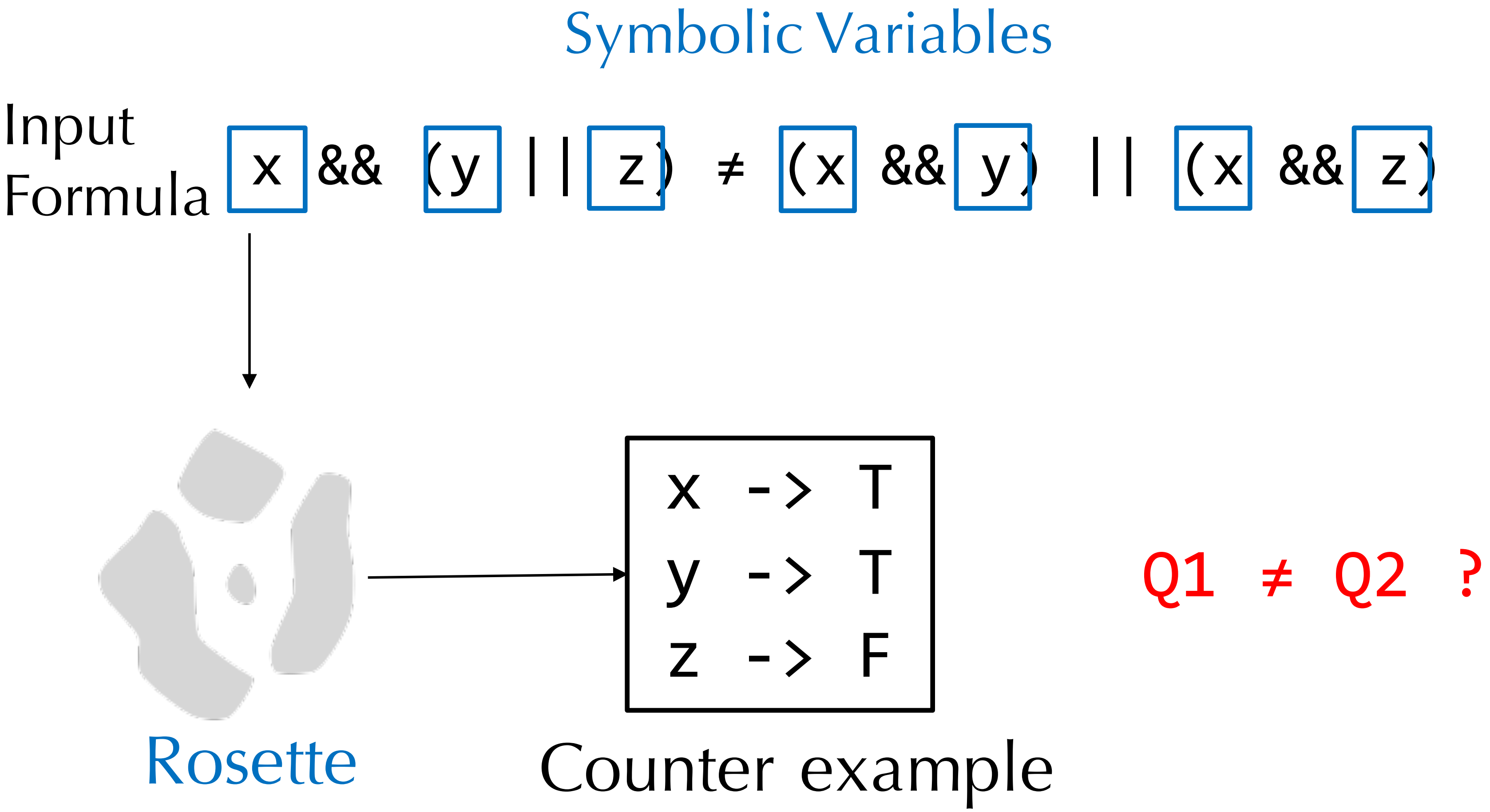
Constraint Solver

Finding counterexamples

Q1 ≠ Q2

Cosette

Q1 =?= Q2



Symbolic Variables

Input Formula $\boxed{x}\ \&\&\ (\boxed{y}\ ||\ \boxed{z}) \neq (\boxed{x}\ \&\&\ \boxed{y})\ ||\ (\boxed{x}\ \&\&\ \boxed{z})$

Rosette

```
x  ->  T
y  ->  T
z  ->  F
```

Counter example

Q1 ≠ Q2 ?

Queries and relations?

Encoding Relations and Queries

Tuple	list of symbolic variables
Relation	list of tuples
Query	operations over relations

Emp (id, salary)

id	salary
sv0	sv1
sv2	sv3

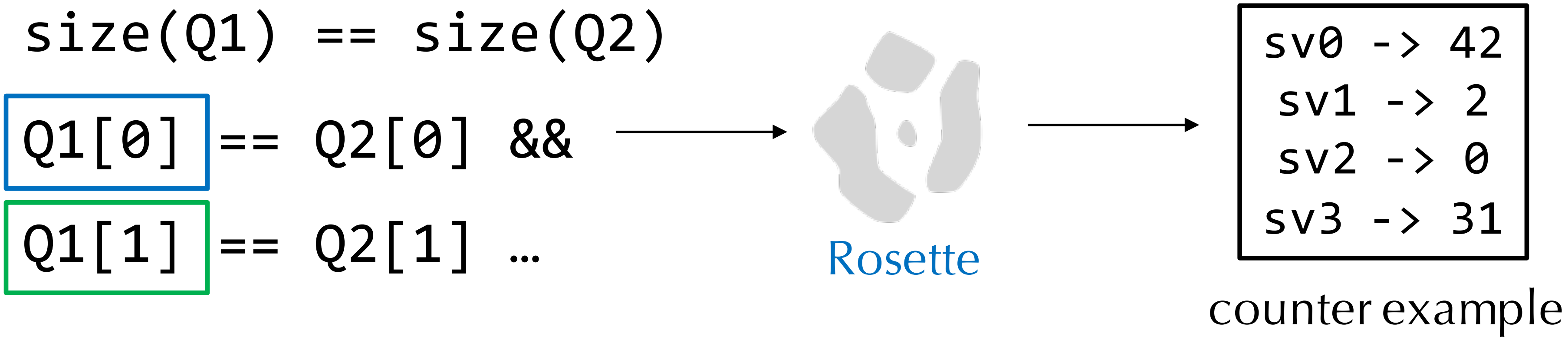
```
Q1 = SELECT id
     FROM   Emp
     WHERE salary > 10000
```

```
if sv1 > 10000:
  assert Q1[0] == sv0
  if sv3 > 10000:
    assert Q1[1] == sv2
else if (sv3 > 10000)
  assert Q1[0] == sv2
```

symbolic constraints

```
Q1 = SELECT ...
Q2 = SELECT ...
```

Q1 ≠ Q2 ?



```
size(Q1) == size(Q2)
Q1[0] == Q2[0] &&
Q1[1] == Q2[1] …
```

```
sv0 -> 42
sv1 -> 2
sv2 -> 0
sv3 -> 31
```

counter example

Optimizations

Incremental solving

Q1 ≠ Q2 ?

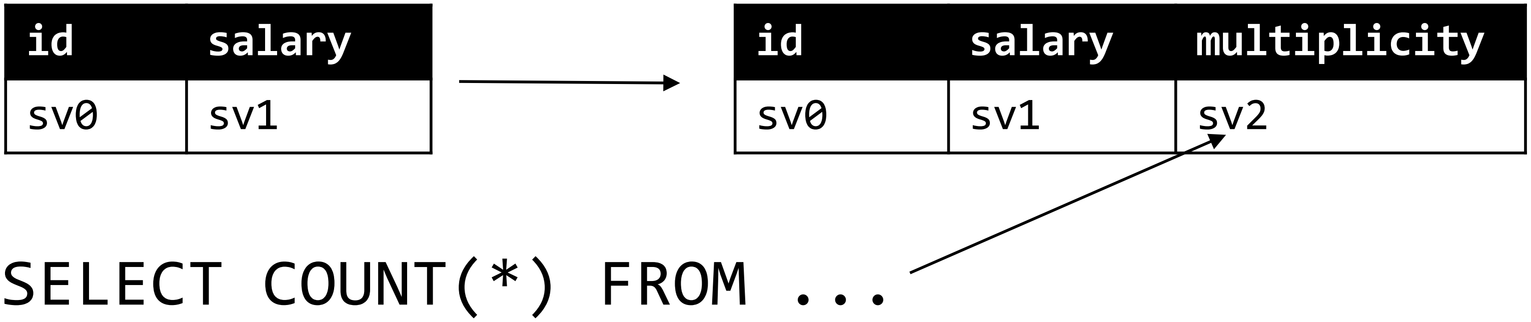
id	salary
sv0	sv1

id	salary
sv0	sv1
sv2	sv3

id	salary
sv0	sv1
sv2	sv3
sv4	sv5

...

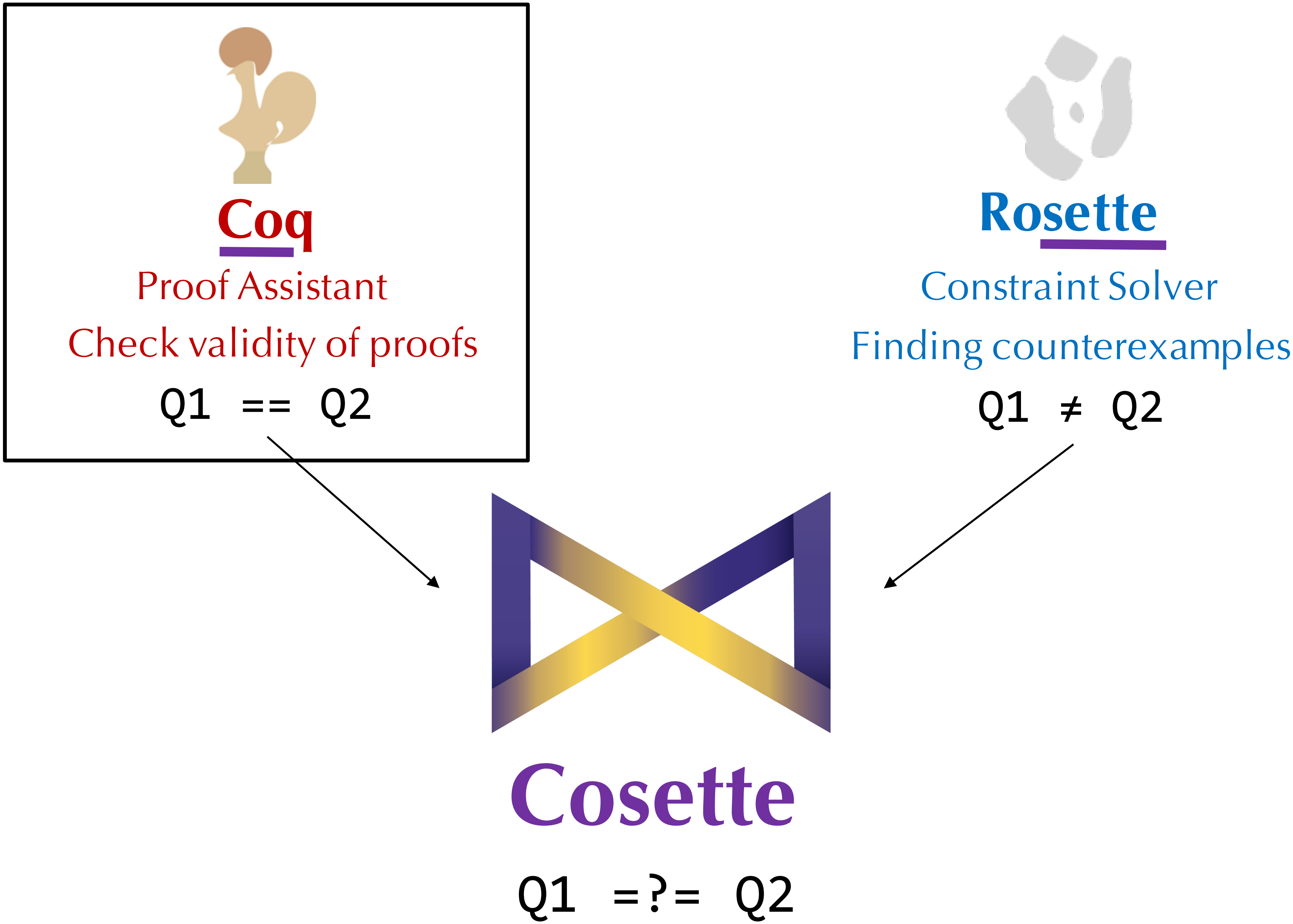
Encode bags with multiplicities



id	salary
sv0	sv1

id	salary	multiplicity
sv0	sv1	sv2

```
SELECT COUNT(*) FROM ...
```



Coq

Proof Assistant

Check validity of proofs

Q1 == Q2

Rosette

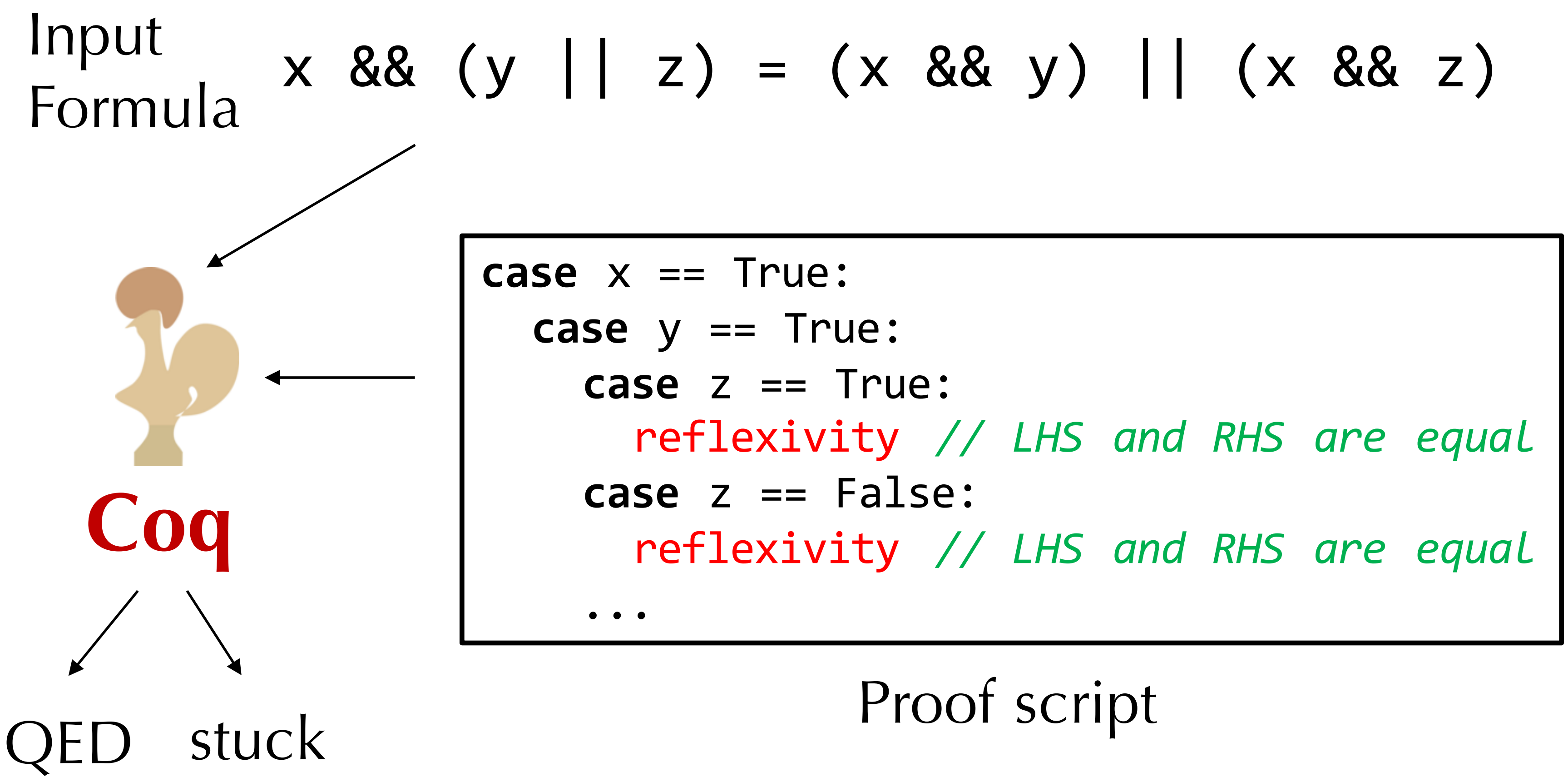
Constraint Solver

Finding counterexamples

Q1 ≠ Q2

Cosette

Q1 =?= Q2



Input Formula

`x && (y || z) = (x && y) || (x && z)`

Coq

QED

stuck

```
case x == True:
  case y == True:
    case z == True:
      reflexivity // LHS and RHS are equal
    case z == False:
      reflexivity // LHS and RHS are equal
    ...
```

Proof script

`Q1 = Q2 ?`

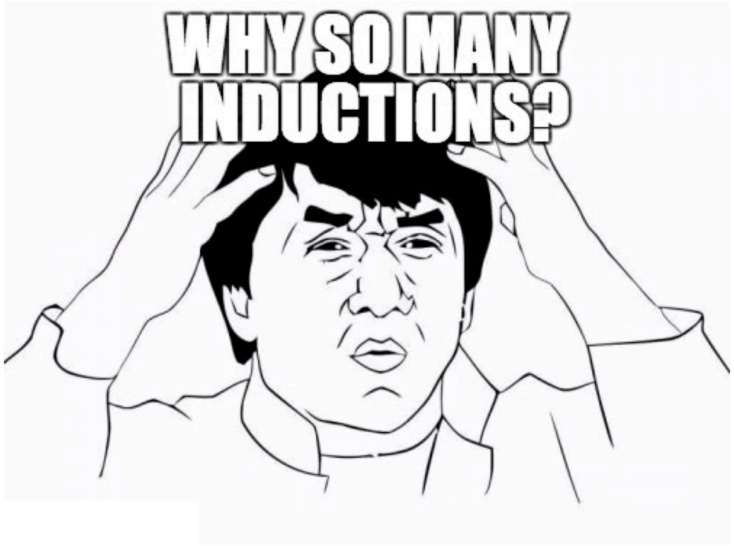
Queries and relations?

Proving Query Equivalences

```
Q1 = SELECT *                    Q2 = (SELECT * FROM R WHERE b)
     FROM (R UNION ALL S)              UNION ALL
     WHERE b                          (SELECT * FROM S WHERE b)
```

Q1 = Q2 ?

```
Induction on R:
  Assume Q1 == Q2 when R has N tuples
  Then when R is of size N+1:
    ...
Induction on S:
  Assume Q1 == Q2 when S has N tuples
  Then when S is of size N+1:
    ...
```



Reason about the contents of R and S

Relation tuple → ℕ

0 just means the tuple isn't in the relation

Predicate tuple → 1/0

Green et al
Provenance semirings
PODS 2007

```
Q1 = SELECT *
     FROM (R UNION ALL S)
     WHERE b
```

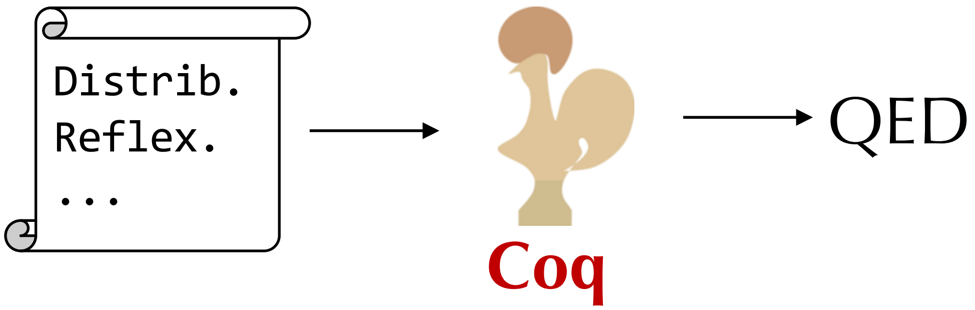
```
Q1(t): (R(t) + S(t)) x b(t)
```

```
Q2 = (SELECT * FROM R WHERE b)
      UNION ALL
     (SELECT * FROM S WHERE b)
```

```
Q2(t): R(t) x b(t) + S(t) x b(t)
```

Q1 = Q2 ?

Algebraic reasoning



Distrib.
Reflex.
...

Coq

QED

Optimizations

Using Homotopy Types to represent $\mathbb{N}$

Generate proof scripts automatically

Heuristics to speed up the proof script search

Bug	3 real-world optimizer rewrite bugs	Inequiv. Rewrites
XData	query and mutant pairs from a test generator	
Exams	questions from undergraduate DB class	Equiv. Rewrites
Rules	23 query rewrite rules from DB papers and real-world optimizers	

Inequivalent Rewrites

Dataset	Total #	Average time taken
Bugs	3	8.3s
XData	9	< 1s
Exams	5	1.3s

Most rewrites can be automatically decided

Most solved within very short time

Equivalent Rewrites

Dataset	Total #	Automatically Decided		# Interactively Decided
		#	Avg time taken	
Exams	4	3	< 1s	1
Rules	23	17	< 1s	6

```
SELECT pnum
FROM Parts
WHERE qoh =
  (SELECT COUNT(shipdate)
   FROM Supply
   WHERE Supply.pnum = Parts.pnum
         AND shipdate < 10)
```

✗
==

```
WITH Temp AS
SELECT pnum, COUNT(shipdate) AS ct
FROM Supply
WHERE shipdate < 10
GROUP BY pnum

SELECT pnum
FROM Parts, Temp
WHERE Parts.qoh = Temp.ct
      AND Parts.pnum = Temp.pnum;
```

Won Kim
On optimizing an SQL-like nested query
TODS 1982

Richard A. Ganski, Harry K. T. Wong
Optimization of Nested SQL Queries Revisited
SIGMOD 1987

5. Bugs in Kim's Algorithm NEST-JA and their Solutions

5.1. The COUNT bug

Cosette

Supply

pnum	shipdate
2	0

5 years

15,778,476x faster

10 secs

P. Seshadri, J. Hellerstein, H. Pirahesh, T. Y. Leung,
R. Ramakrishnan, D. Srivastava, P. Stuckey, S. Sudarshan

Cost-Based Optimization for Magic: Algebra and Implementation.
SIGMOD 1996

Introduction of θ-semijoin: ✓

$R_1 \ltimes$

Pushing θ-

$(R_1 \ltimes$ $\theta_2 \; R_3$

$R_2' =$

```
Dear Praveen, Joe, Hamid, Cliff, Raghu,
Divesh, Peter, and Sudarshan:

We have proven the correctness of your
semijoin rewrite rules using Cosette. I
hope you can now sleep in peace.

Regards,
The Cosette Team
```

Pushing θ-semijoin through aggregation: ✓

$${}_{\bar{g}}\mathcal{F}_{\bar{f}}(R_1) \ltimes_{c_1=c_2} R_2 \equiv {}_{\bar{g}}\mathcal{F}_{\bar{f}}(R_1 \ltimes_{c_1=c_2} R_2)$$

cosette.cs.washington.edu