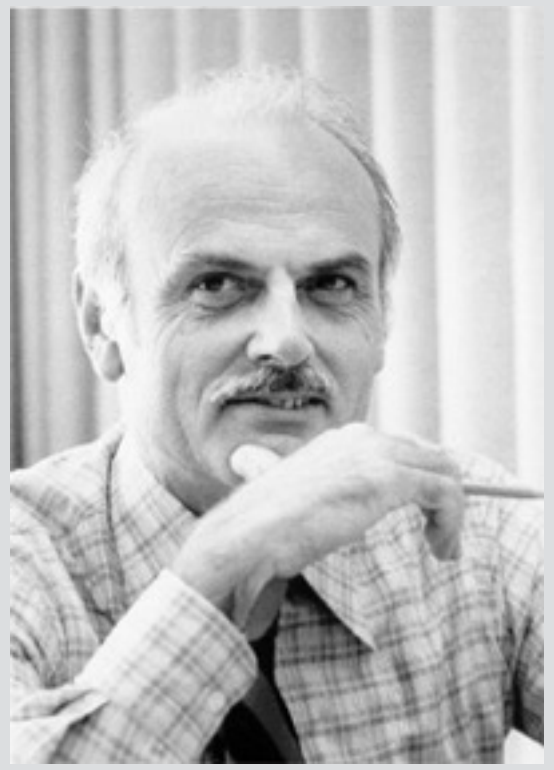
Cosette: An Automated Solver for SQL

Shumo Chu, Chenglong Wang, Konstantin Weitz, Alvin Cheung, and Dan Suciu

Automated Solver for SQL: Q1 = Q2?

Motivation

- SQL is great
- A restricted abstraction enabling powerful optimizations
- Goal: formally reason SQL equivalences with automation:
  - Verify/find bugs in query optimization
  - Test generation
  - Auto grading
  - …



30 years
database research

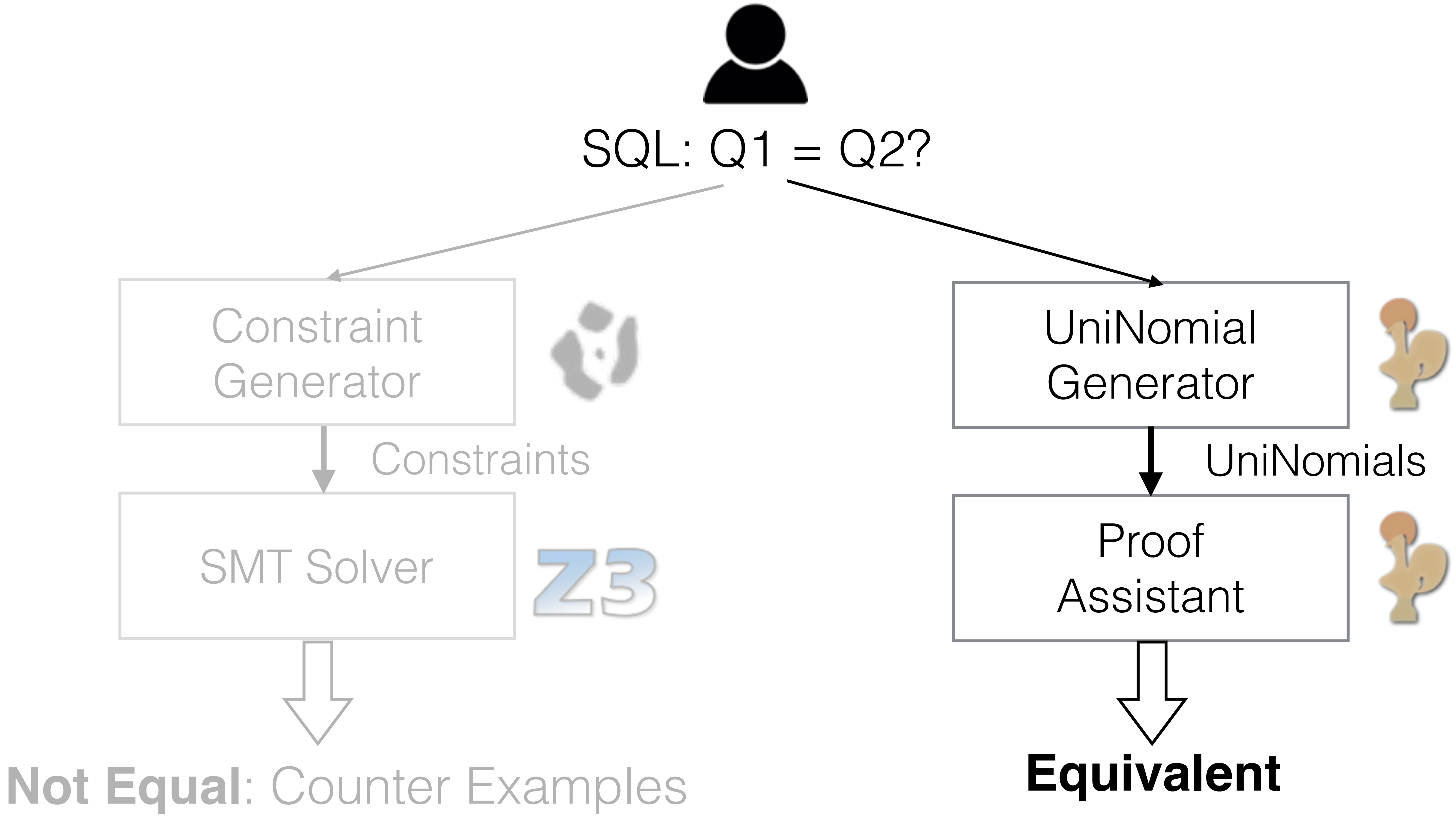
Challenges

- Deciding the equality of two relational queries are undecidable
- Rich language features
  - Aggregation and Group By
  - Index
  - Correlated Subqueries
  - Foreign keys
  - …….

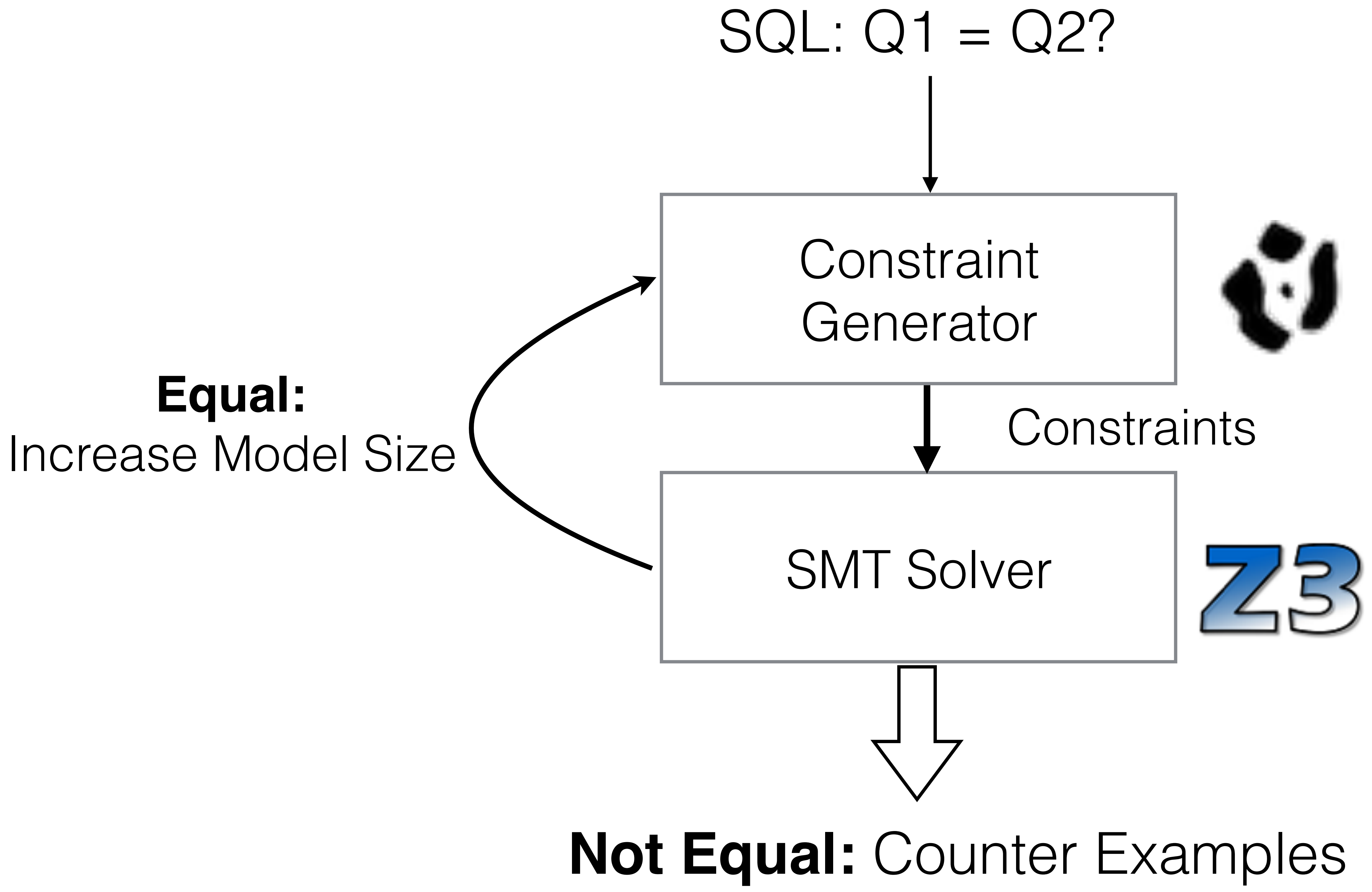
Boris A.
Trakhtenbrot

Cosette: Coq + Rosette

- An (almost) automated solver for SQL by combining constraint solver and proof assistant



Finding Counter Examples with SMT Solver



SQL: Q1 = Q2?

Constraint Generator

Constraints

SMT Solver

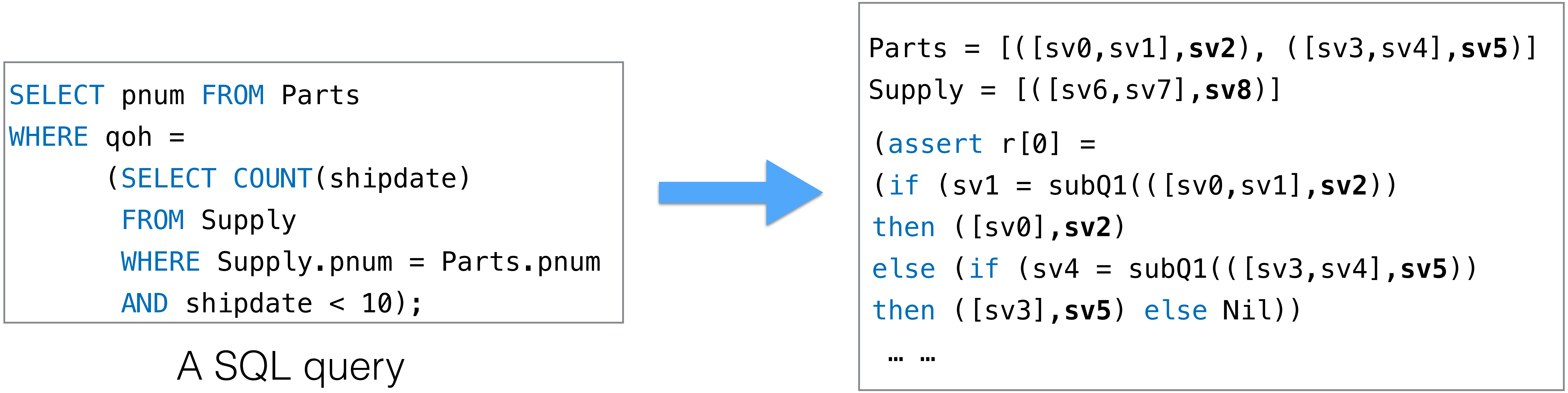
Equal: Increase Model Size

Not Equal: Counter Examples

Encoding SQL

- A tuple as a list $\texttt{Tuple := List <}\overset{\texttt{sv}}{\cancel{\texttt{Integer}}}\texttt{>}$
- A relation as a bag $\texttt{Relation := List <Pair<Tuple, }\overset{\texttt{sv}}{\cancel{\texttt{Integer}}}\texttt{>>}$
- A SQL query as operations over symbolic values

Encoding SQL



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

A SQL query

```
Parts = [([sv0,sv1],sv2), ([sv3,sv4],sv5)]
Supply = [([sv6,sv7],sv8)]

(assert r[0] =
(if (sv1 = subQ1(([sv0,sv1],sv2))
then ([sv0],sv2)
else (if (sv4 = subQ1(([sv3,sv4],sv5))
then ([sv3],sv5) else Nil))
 … …
```

SMT Constraints

Example: The Count Bug

- An infamous query optimization bug (*Kim, W. ACM Trans. Database System 1982*)

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

Q1

≠

```
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;
```

Q2

Q1 and Q2 are not equal since Q2 ignores the cases when the count of a group is zero

Cosette:

pnum	qoh	**multiplicity**
0	0	**8**
2	2	**15**

Parts

pnum	shipdate	**multiplicity**
2	0	**2**

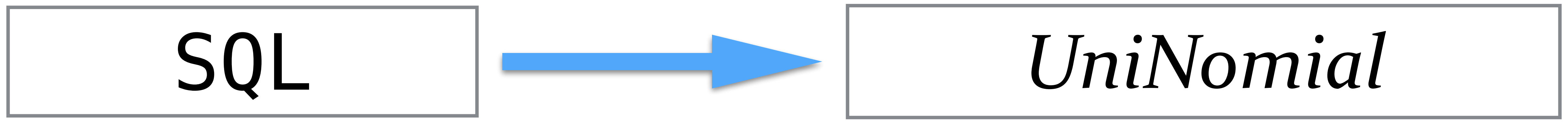
Supply

What about equivalent queries?

Say UniNonials

Proving Equivalences with Proof Assistant

- Unbounded verification with proof assistant
- SQL where relations are modeled as lists requires finding invariants
- Inspired by K-Relation, We developed SQL semantics that eases reasoning equivalences:



SQL

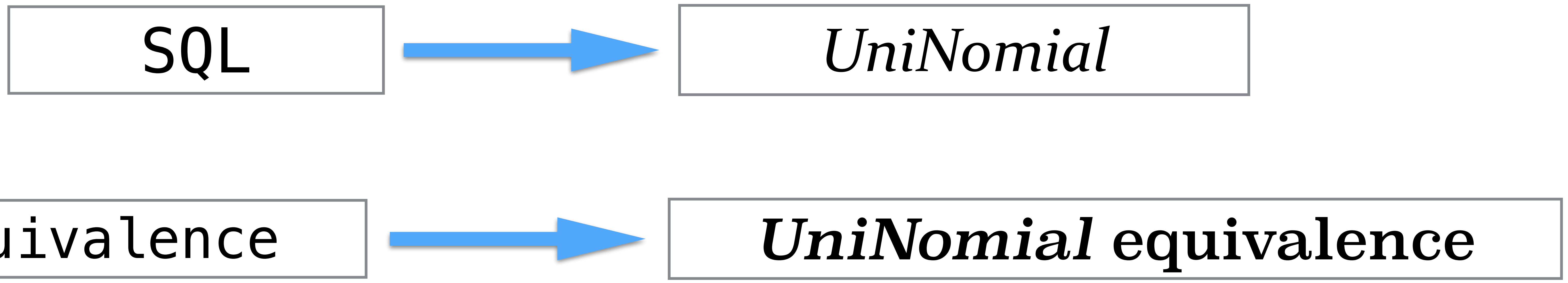
UniNomial

Proving Equivalences with Proof Assistant

SQL → *UniNomial*

`a:Relation`	`⟦a⟧:Tuple -> ~~ℕ~~` **HoTT Type**
`b: Predicate`	`⟦b⟧:Tuple -> {0, 1}`
`SELECT * FROM a WHERE b`	`λt. ⟦a⟧ t × ⟦b⟧ t`
`a0 UNION ALL a1`	`λt. ⟦a0⟧ t + ⟦a1⟧ t`
`SELECT k FROM a`	$\lambda t. \sum_{t':Tuple}$ `if t’.k = t.k then ⟦a⟧ t else 0`

Proving Equivalences with Proof Assistant



```
    SELECT *
    FROM (a0 UNION ALL a1)
    WHERE b
```

```
λt. (⟦a0⟧ t + ⟦a1⟧ t) × ⟦b⟧ t
```

change order

```
(SELECT * FROM a0 WHERE b)
          UNION ALL
(SELECT * FROM a1 WHERE b)
```

```
λt. ⟦a0⟧ t × ⟦b⟧ t + ⟦a1⟧ t × ⟦b⟧ t
```

```
Proof:  function_extensionality; rewrite assoc_sum;  reflexivity. Qed.
```

Evaluating **Cosette**

- **Bug**: 3 real-world optimizer bugs
- **XData**: query and mutant pairs collected from XData, a test generation framework
- **Exams**: a set of questions from the undergraduate data management class
- **Rules**: 23 query rewrite rules from database literatures and real-world optimizers

Unequal SQLs

Equivalent SQLs

Evaluating **Cosette**

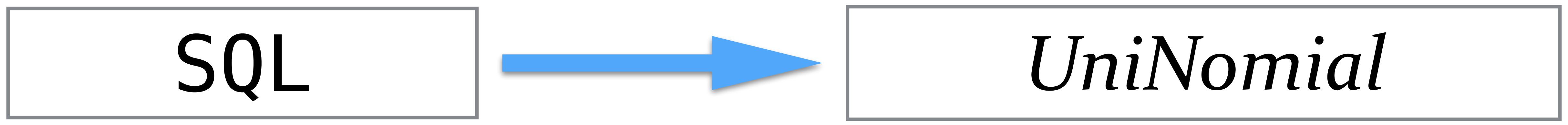
Dataset	Equiv?	Total Number	Automatically Decided		Interactively Decided
			No.	Avg. Time	
Bugs	No	3	3	8.3 s	—
Exams	No	5	5	1.3 s	—
XData	No	9	9	< 1 s	—
Rules	Yes	23	17	< 1 s	6
Exams	Yes	4	3	< 1 s	1

400 LOC to 15 LOC

Conclusions and Future Work

- Cosette: The first SQL solver combining SMT solver and proof assistant
- Automatically generating a verified query optimizer for new system
- Synthesize new optimization rules
- Website: cosette.cs.washington.edu

Why HoTT?



`a:Relation`

`⟦a⟧:Tuple -> Type`

`SELECT name FROM a`

$$\lambda t. \sum_{t':Tuple} \texttt{if t'.name = t.name then ⟦a⟧ t else 0}$$

name	salary
“James”	$10,000
“Alex”	$20,000
“James”	$30,000
“Alex”	$40,000
“Alex”	$50,000

name
“James”
“Alex”
“James”
“Alex”
“Alex”