Semantics of Conjuctive Query

Yihao Sun, Utah State University, Sept 2026

Computer Science vs. Software Engineering

- Computer Scientist know the meaning of every term they use.
- They can then judge what is correct.
- Knowing the precise meaning will give us clue for next-step and implementation.

What need to be precisely defined?

relation edge(from: City, to: City).

relation two-hop(from: City, to: City).

two-hop(from, to) :-
    edge(from, mid),
    edge(mid, to).

Type, relation, variable, comma …
Syntax : **How we write**
Semantics: **How we interpret**

Database Type?

- A typing judgment states that an expression has a type.
- In our many-sorted logic, sorts classify terms.

I assigns each sort τ a nonempty carrier (or domain).

$$\tau^{\mathcal{I}} = [[\tau]]^{\mathcal{I}}$$

Example: Sort of cities

All city can possibly appear in DB

$$City^{\mathcal{I}} = \{\text{Logan}, \text{Provo}, \text{Ogden}, \ldots\}\,.$$

Symbol City is a **sort** (type), $\text{City}^{\mathcal{I}}$ is the entire set of values.

Relation

- A relation symbol receives its meaning in an interpretation.

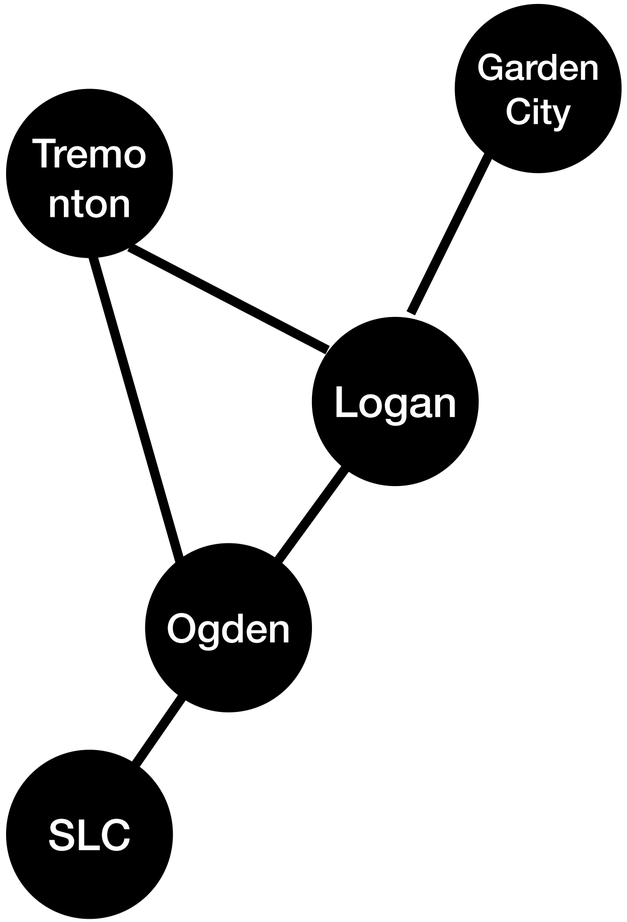
  A relation schema:

$$\mathscr{R}(A_1 : \tau_1, \ldots, A_n : \tau_n)$$

  A finite relation instance:

$$I(\mathscr{R}) \subseteq \tau_1^I \times \cdots \times \tau_n^I$$

Instance for edge



$\mathcal{I}(Edge)$

(SLC, Ogden), (Ogden, Logan),
(Ogden, Tremonton), (Logan, Tremonton)
(Logan, Garden City)

…. in reverse

Interpretation of Database

- Fix a database schema and the carriers of its sorts.
- An instance I assigns every relation symbol R a relation I(R) of the declared sorts.
- I is the ‘world’ in which we evaluate the query.

Atom

```
relation edge(from: City, to: City).

relation two-hop(from: City, to: City).

two-hop(from, to) :-
    edge(from, mid),
    edge(mid, to).
```

What is atom?
Type of atom?

edge(“SLC”, x)



edge(“SLC”, “Logan”)

Valuation

v assigns each variable a value of its sort.

$$I \vDash r(t_1, \ldots, t_n)[v]$$

Exactly when:

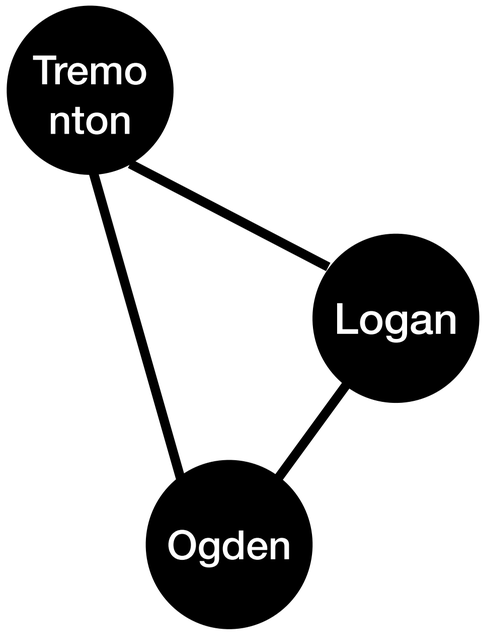
$$v(t_1, \ldots, t_n) \in \mathcal{I}(r)$$

All occurrences of a variable share one value.

Literals keep their interpretations.

Atom (partially)

In this three-city instance J:



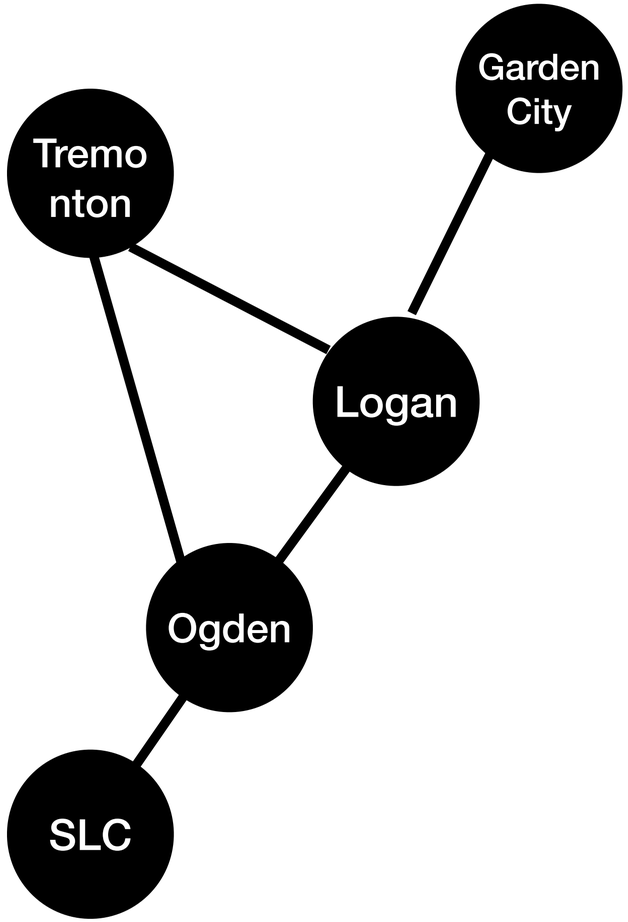
$$J \models \text{edge}(\text{"Logan"}, x)[v]$$

Literal argument

edge(“Logan”, x)

Which values of x satisfy the atom?

More valuations



edge(from, “Logan”),
edge(“Logan”, to).

edge(from, from).

edge(“Logan”, mid),

edge(mid, “SLC”).

Not excited?

- Imagine your favorite function in normal programming language.
- Use its name as relation name and arguments as logic variable.
- Add return value as last logic variable.

Power(x, 3, 8).

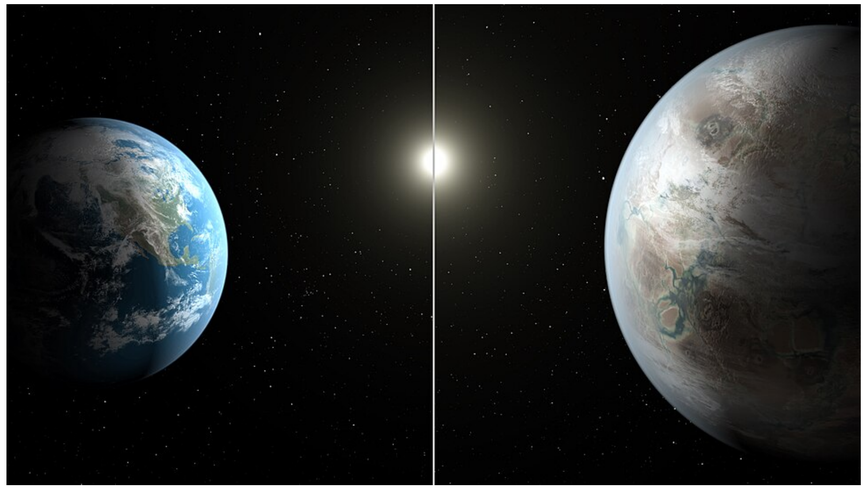
Power(x, y, 8).

Why a relation is named relation?

- Even if before, we have seen its denoted as set operations.

World

earth_like(“Kepler-458”)



- Whether a proposition is true?
- “The truth” is a hard question to answer.
- Logician make a theory about truth: Model theory

In a world I, a ground atom R(ā) is true exactly when $\bar{a} \in I(R)$.

Closed World Assumption (CWA)

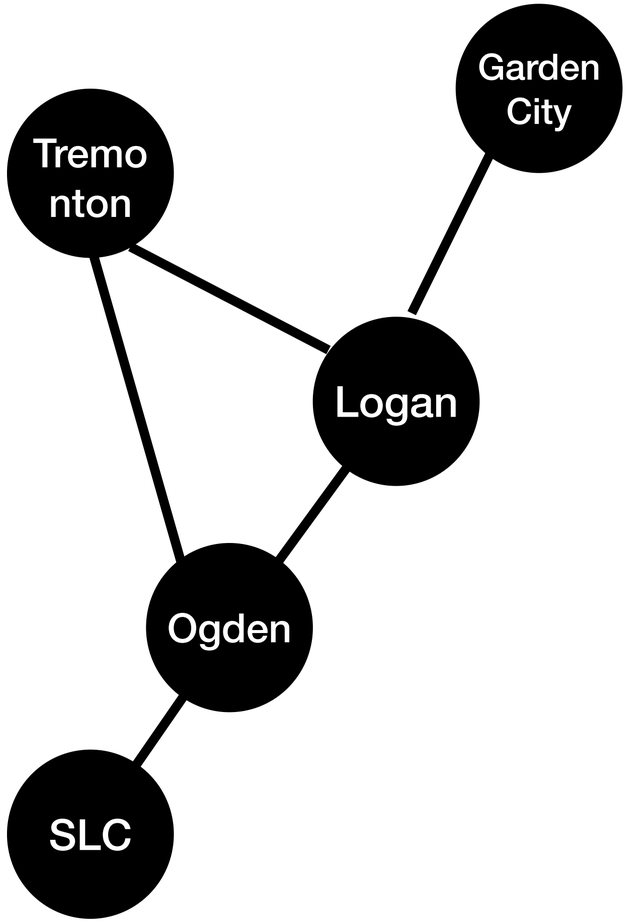
The instance records all true ground facts.

edge(“Logan”, “São Paulo”) 

For a ground tuple ā, absence from I(R) means false.

$$I \models \neg R(\bar{a}) \quad \iff \quad \bar{a} \notin I(R)$$

Negation



edge(from, mid),
¬ edge(mid, “Tremonton”).

The head

```
two-hop(from, to) :-
    edge(from, mid),
    edge(mid, to).
```

One successful valuation

from	mid	to
SLC	Ogden	Logan

The head tuple data is based on the valuation inside bodies.

Safety

To force rule, logical make sense, we need some extra restrictions.

```
?-hop(from, destination) :-
    edge(from, mid),
    edge(mid, to).
```

$$Vars(head) \subseteq Vars(body)$$

Safety



```
% Administrators can edit any document.
can_edit(User, Document) :-
    role(User, admin).

% Editors can edit documents they own.
can_edit(User, Document) :-
    role(User, editor),
    owns(User, Document).
```

Can we also make this works in our language?

Together: Conjunctive Query

$$q:\ ans(\bar{h}) :- R_1(\bar{t}_1), \ldots, R_m(\bar{t}_m)$$

$$R_i \in \mathscr{R}, \qquad ans \notin \mathscr{R}, \qquad m \geq 0$$

Terms are variables or constants of the required sorts.

$$Vars(\bar{h}) \subseteq Vars(body)$$

The query result

$$q(I) = \{\ v(\bar{h}) \mid I \models B[v]\ \}$$

Body B and ordered head tuple $\bar{h}$

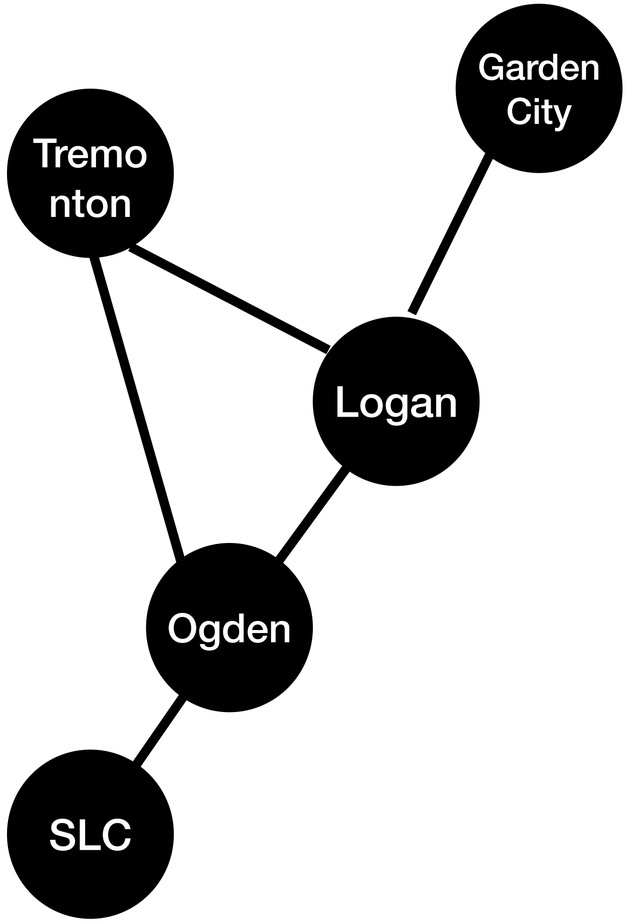
v ranges over well-sorted valuations of all query variables.

q
written query

I
input instance

q(I)
result relation

Beyond Set Semantics



- Is set the only semantic that make sense?
- What if … we want database know how how many routes are possible between two cities?

Language extensions

```
edge(from, “Logan”),
edge(“Logan”, to),
if to != “SLC”.
```

```
edge_length(from, “Logan”, n),
edge_length(“Logan”, to, n + 100),
```



```
let x = “logan”
edge(from, x),
edge(x,  to).
```

Rust expression

Note for today's class

- https://pldi.me/modern-query-processing-notes/02-the-relation-that-wasnt-stored.html
- Next week will be about how to do proc macro in Rust