Graph pattern matching semantics

First openCypher Implementers Meeting

Oskar van Rest
Oracle
February 8, 2017
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Graph pattern matching semantics

Isomorphism
• Two query vertices may not map to the same data vertex
  – Semantic for multi-graphs is undefined

Cyphermorphism
• Two query edges may not map to the same data edge

Homomorphism
• No restriction

For most queries, this works very well: users who are not aware of the semantic still get the desired result.

less restrictive (more powerful)

easier to get started with
When does the semantic fail?

### Isomorphism

**codeGraph**

```
MATCH (n:Function) -[:calls]-> (m)
WHERE n.name = 'factorial'
RETURN m.name
```

**Result with isomorphism**

- `m.name` = 'factorialHelper'

**Result with homomorphism**

- `m.name` = 'factorial'

### Cyphermorphism

**familyTree**

```
MATCH
  (n:Person) -[:has_parent]-> () -[:has_parent]- (grandparent)
  (m:Person) -[:has_parent]-> () -[:has_parent]- (grandparent)
WHERE n <> m
RETURN grandparent.name
```

**Query:** “find common grandparents”

- `grandparent.name` = 'Amber'

**Result with cyphermorphism**

- `grandparent.name` = 'Amber'

**Result with homomorphism**

- `grandparent.name` = 'Amber'
Pros and cons of the different semantics

• They all have the same worst-case time complexity: $O(n^k)$ ($n =$ num. data vertices, $k =$ num. query vertices)
  – However, if we apply iso/cypher-morphism to recursive path queries, things blow up

• Isomorphism & cyphermorphism have limitations (see previous slide)

• Homomorphism may return more matches than expected and requires additional non-equality constraints

• Isomorphism & cyphermorphism don’t translate as well to/from SQL
Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.