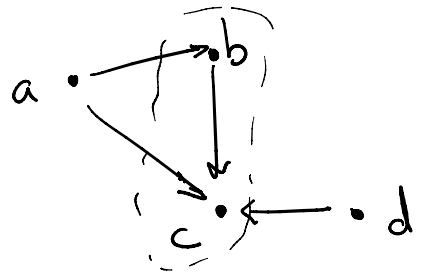
Lecture 23

Tuesday, November 10, 2020 3:58 PM

Graph Query Languages

edge
a → b
b → c
a → c
d → c

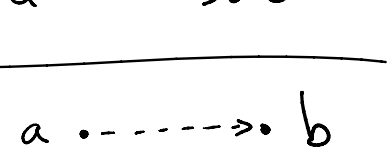


reachable	
a	b
a	c
b	c
d	c

— "All nodes which can be traced back to a".

"All nodes which are reachable from a".

— "Recursion".



Paths of length 1 a ⇢ b

— "Get stuck in a cycle". a → b b ⇢ c

— Not stuck in a cycle. a •——→• b b •- - - - -→ c

Paths of length n+1. a •- - - - -→ • c

— "Are we assuming a DAG. / generic graph?"

— "Until" / Fixpoint

$f(a) = a$

← Do something to it

Start with the table

End up with the same table.

"Until nothing can be added ..."

— Some stopping criterion exists.

Termination is guaranteed*

* conditions apply.

while (———) do {

until this becomes false.

| ← Repeat instructions

}

Repeat

|

until nothing changes.

"All nodes which are reachable from a in 2 steps.

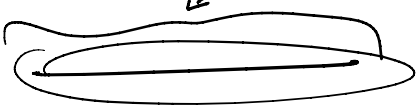
$$\Pi_{1,4}(\text{edge} \bowtie \text{edge})$$

"All nodes which are reachable from a in 5 steps"

$$\Pi_{1,10}(\text{edge} \bowtie \text{edge} \bowtie \text{edge} \bowtie e \bowtie e)$$

You can't decide the SQL query before knowing the diameter.

~~"regular"~~ path query.

Cypher: match 

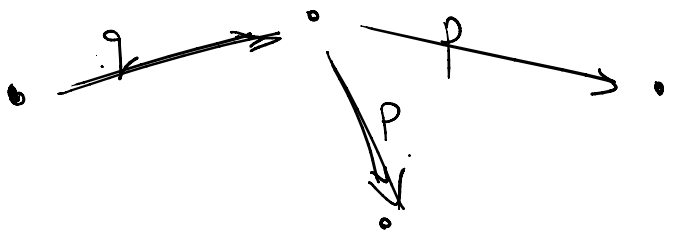
where _____

return _____

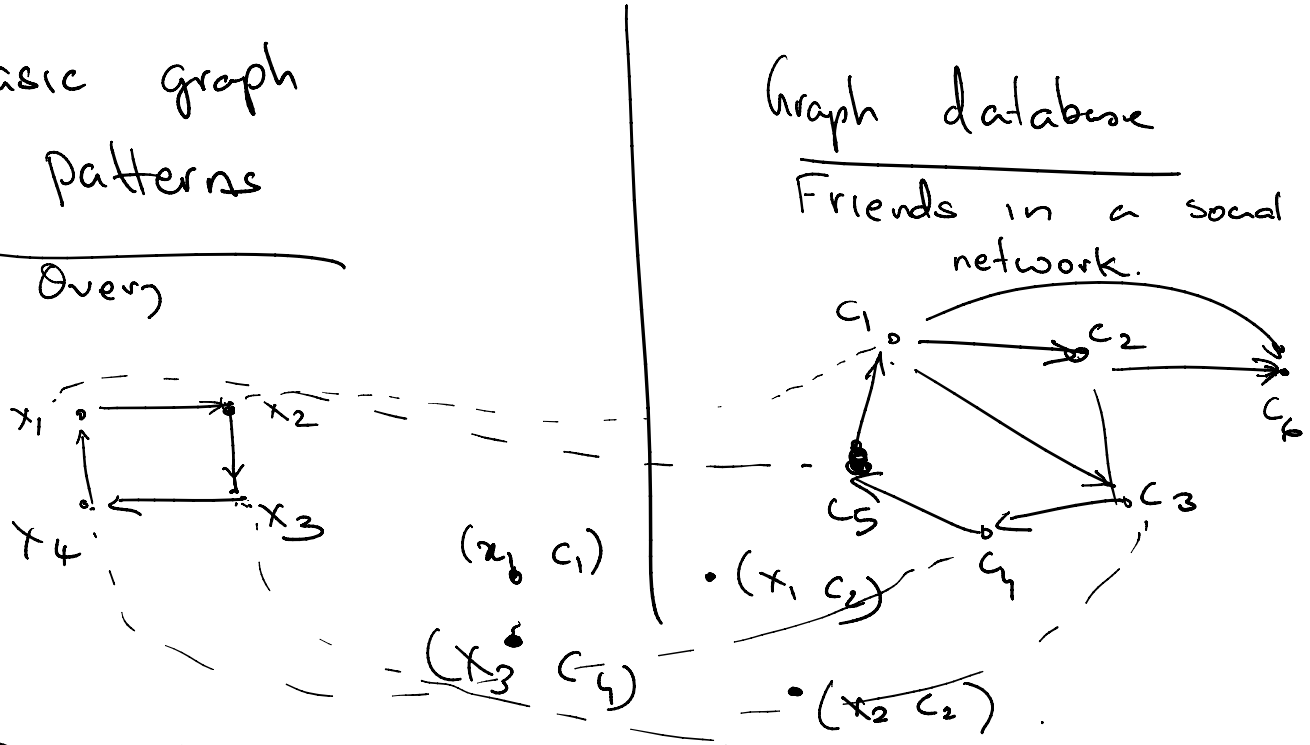
SQL: select ____

from ____

where ____



How to evaluate graph queries ?



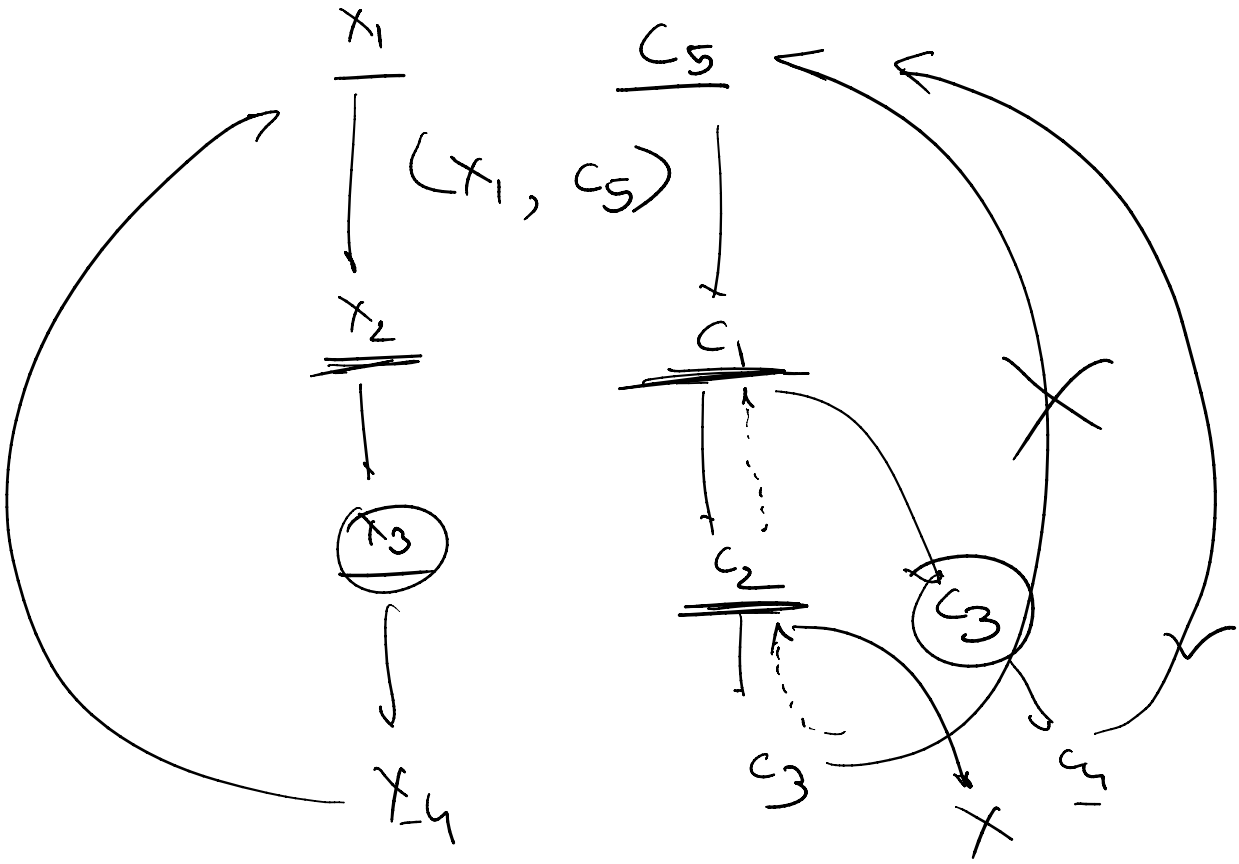
Q1: Does this BGP match the database? ✓

Q2: What is the algorithm ?

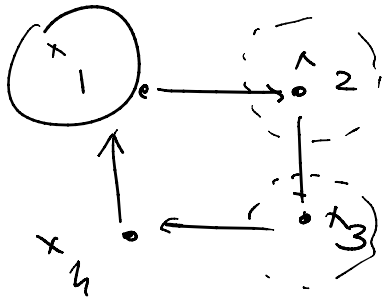
"Brute force".

- Product of the BGP & the data.
- Do some kind of backtracking search.

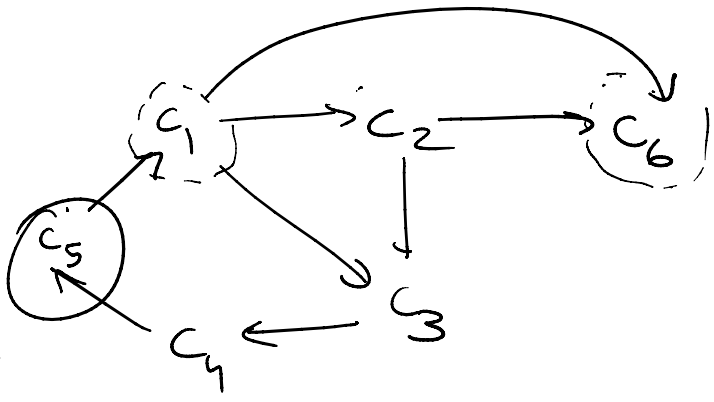
— Complexity? $|V| + |E|$



Quem

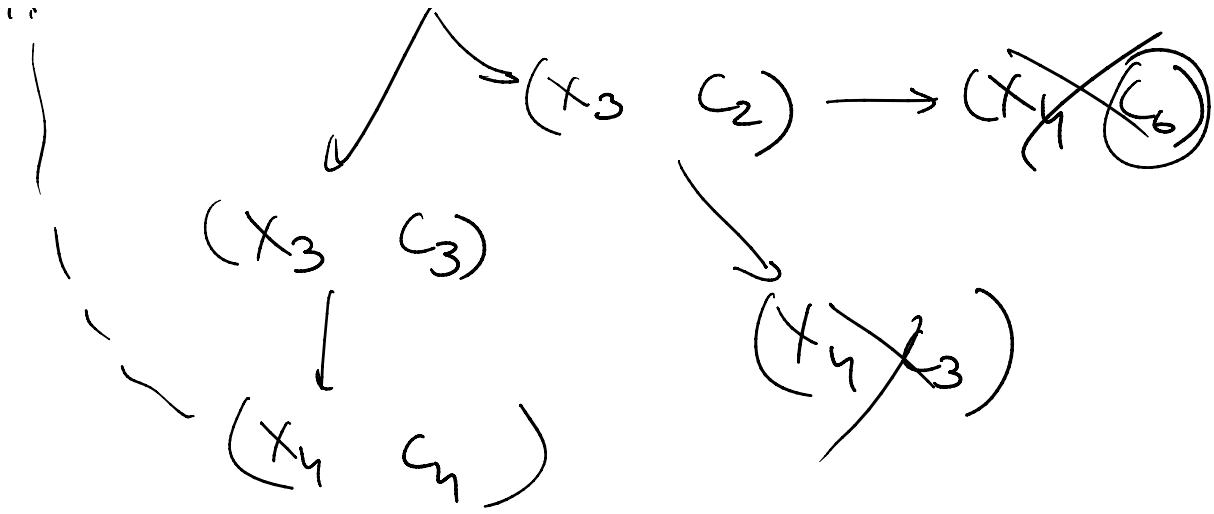


Data



$(x_1\ c_5) \longrightarrow (x_2\ c_1) \longrightarrow (x_3\ c_6)$

$\longrightarrow (x_2\ c_2) \longrightarrow (x_4$



Proposition: NP-complete.

As we have defined it

query evaluation = subgraph isomorphism.

21 famous NP-complete problems.