Updated 2020/2/11

T2: Complexity of Query Evaluation
L11: Tree pattern queries

Wolfgang Gatterbauer

CS7240 Principles of scalable data management (sp20)

https://northeastern-datalab.github.io/cs7240/sp20/

Date: 2020/2/11

Islands of Tractability of CQ Evaluation

- Major Research Program: Identify tractable cases of the combined complexity of conjunctive query evaluation.
- Note: Over the years, this program has been pursued by two different research communities:
  - The Database Theory community
  - The Constraint Satisfaction community
- Explanation:

Constraint Satisfaction Problem

≡ (Feder-Vardi, 1993)

Homomorphism Problem

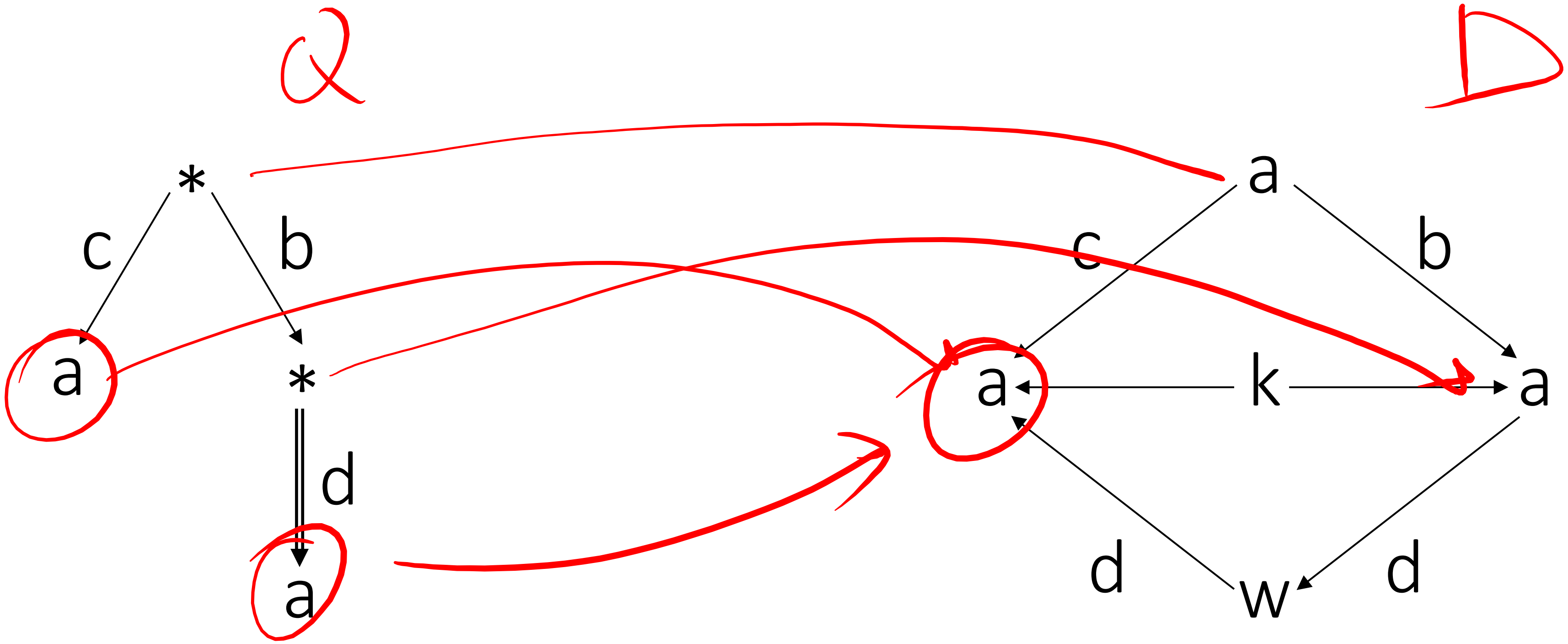
≡ (Chandra-Merlin, 1977)

Conjunctive Query Evaluation

Outline: Complexity of Query Equivalence

- Query equivalence and query containment
  - Graph homomorphisms
  - Homomorphism beyond graphs
  - CQ containment
  - Beyond CQs
  - CQ equivalence under bag semantics
  - CQ minimization
  - Nested queries
  - Tree pattern queries

Tree pattern queries



From: “Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data” by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.

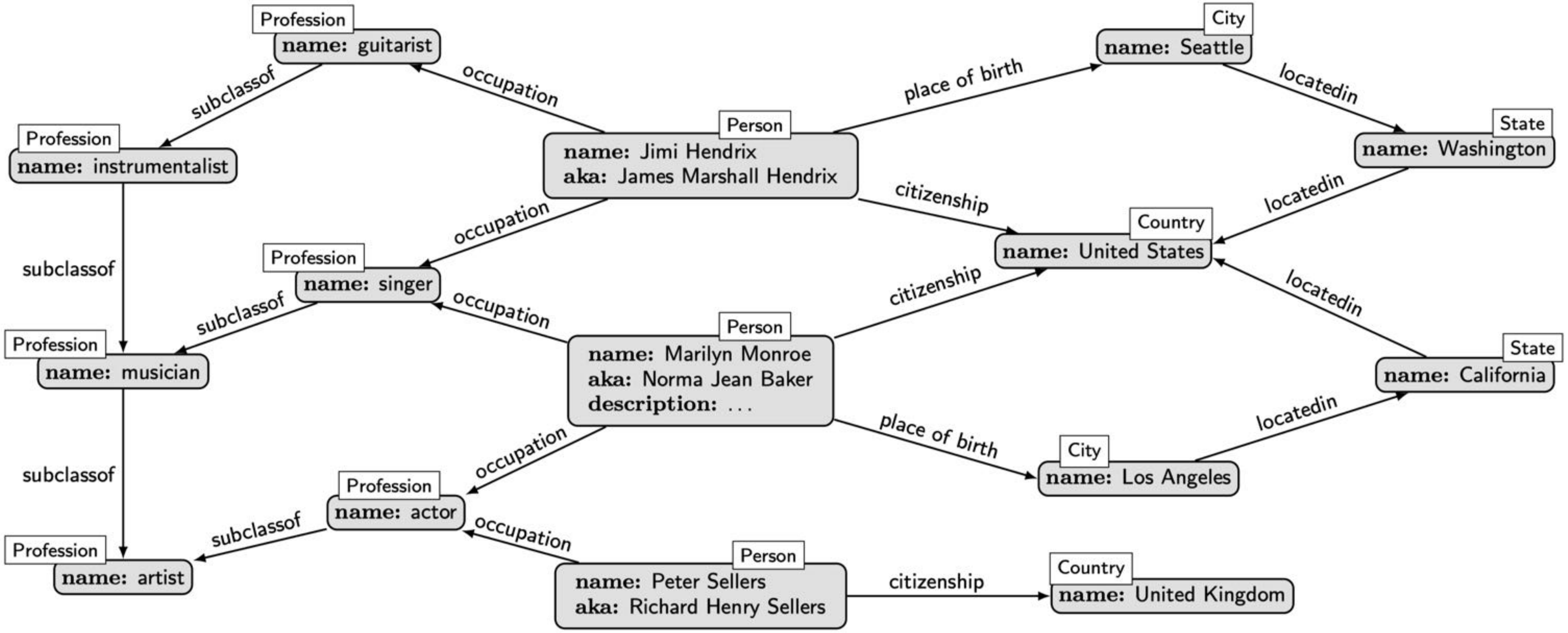
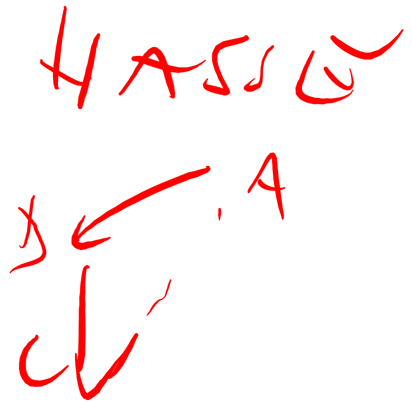


Figure 1: A graph database (as a *property graph*), inspired on a fragment of WikiData



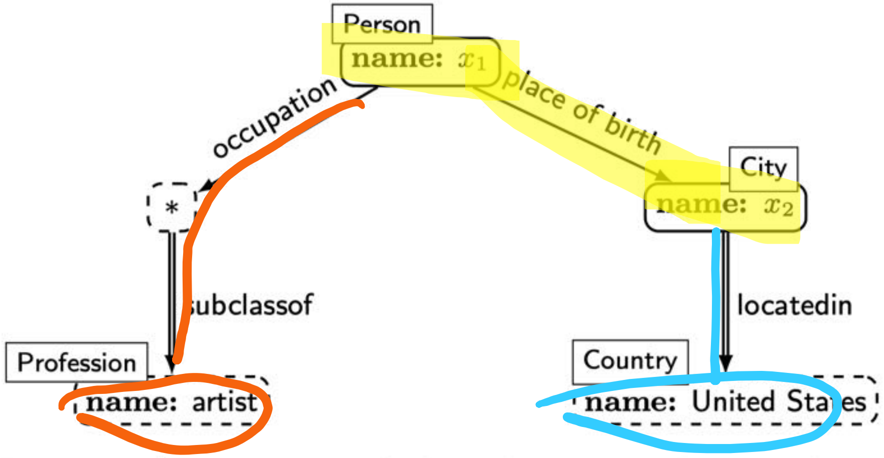


Figure 2: A tree pattern finding the artists who were born in the United States. The query returns the person names and the cities where they were born. (Fully circled nodes are return nodes.)

From: "Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data" by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.

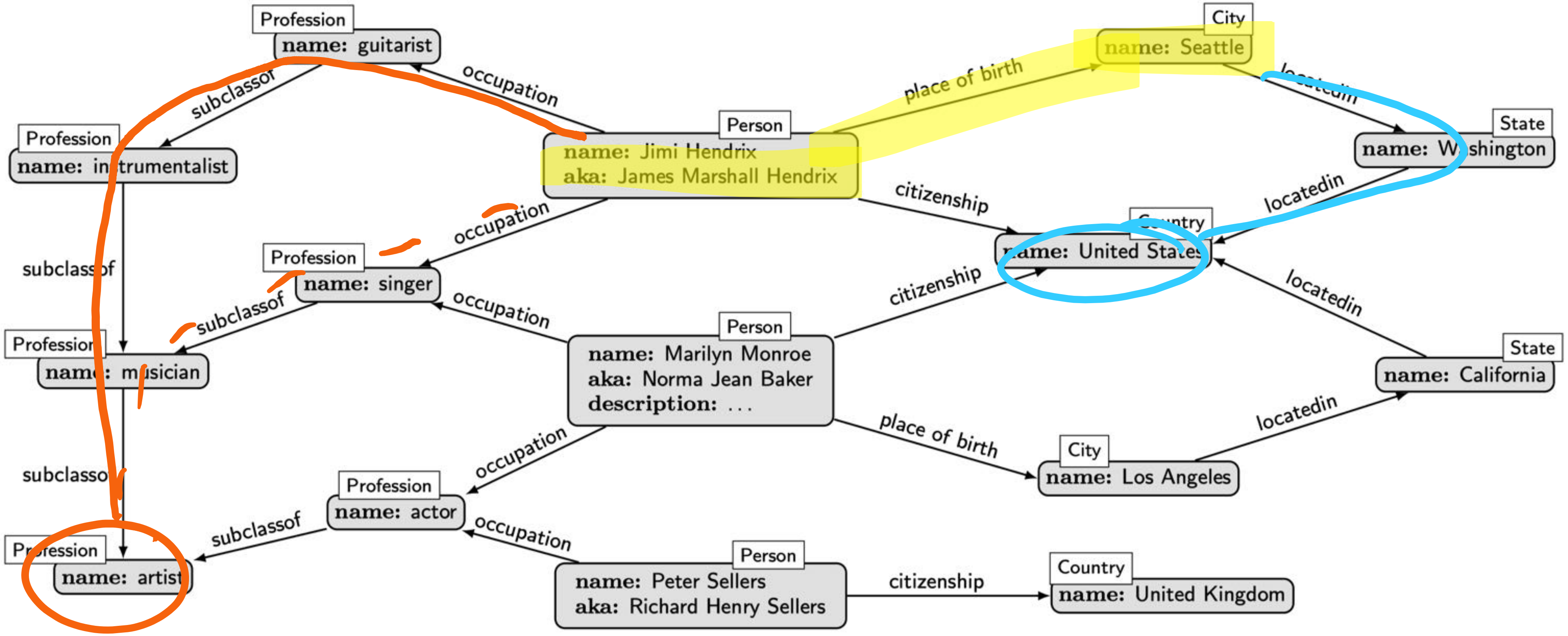


Figure 1: A graph database (as a *property graph*), inspired on a fragment of WikiData

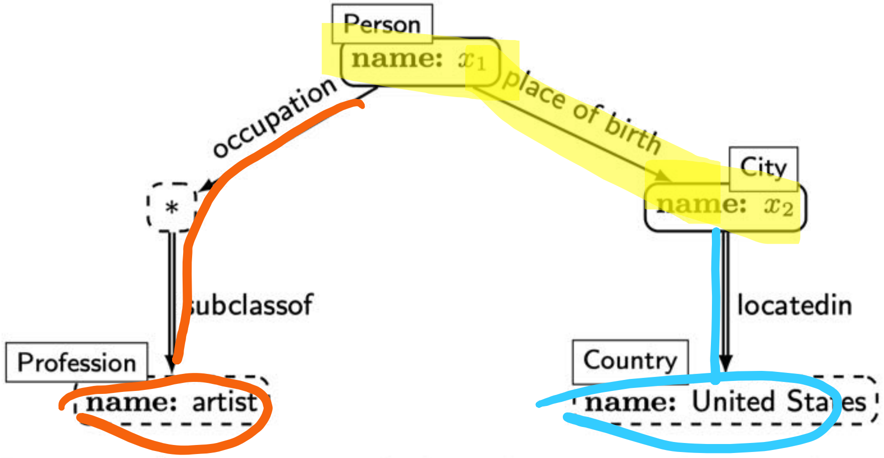


Figure 2: A tree pattern finding the artists who were born in the United States. The query returns the person names and the cities where they were born. (Fully circled nodes are return nodes.)

From: "Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data" by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.

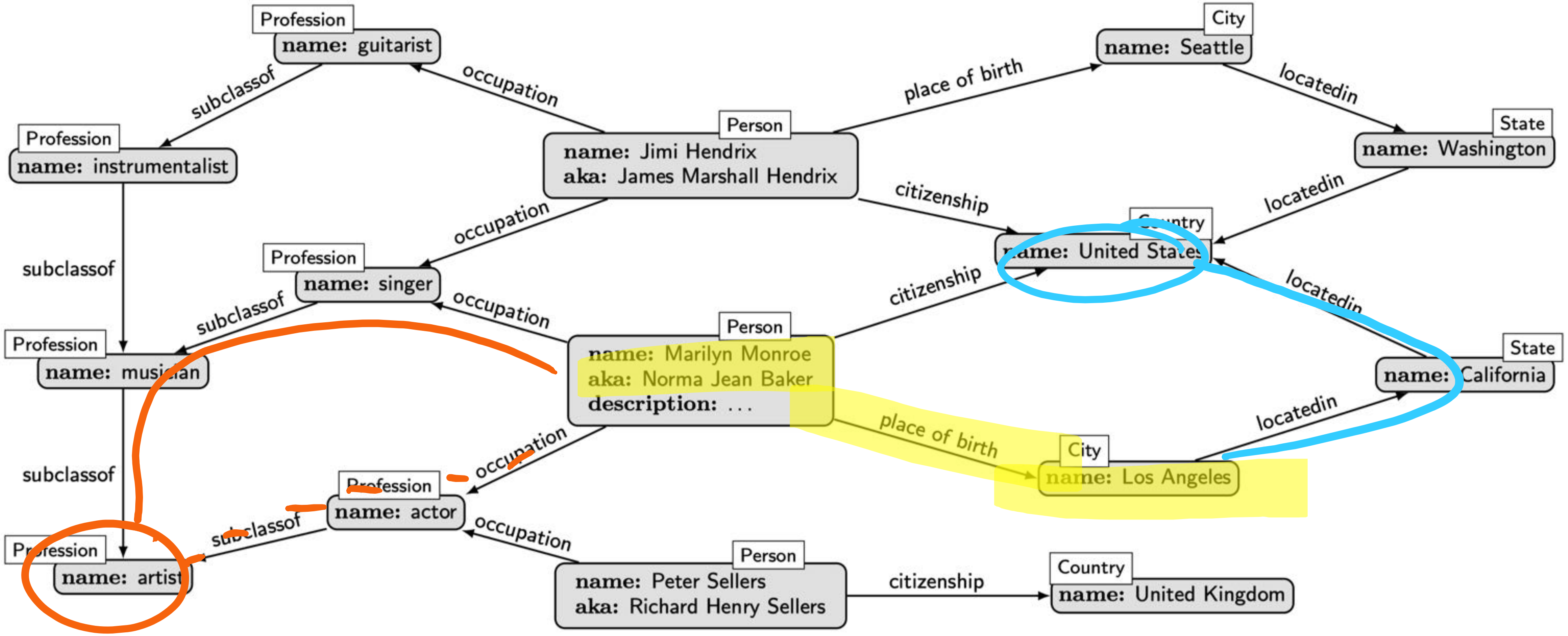


Figure 1: A graph database (as a *property graph*), inspired on a fragment of WikiData

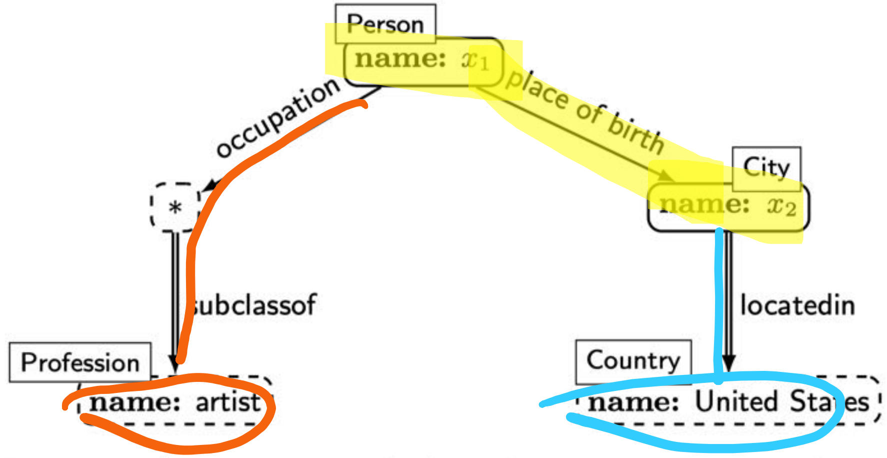


Figure 2: A tree pattern finding the artists who were born in the United States. The query returns the person names and the cities where they were born. (Fully circled nodes are return nodes.)

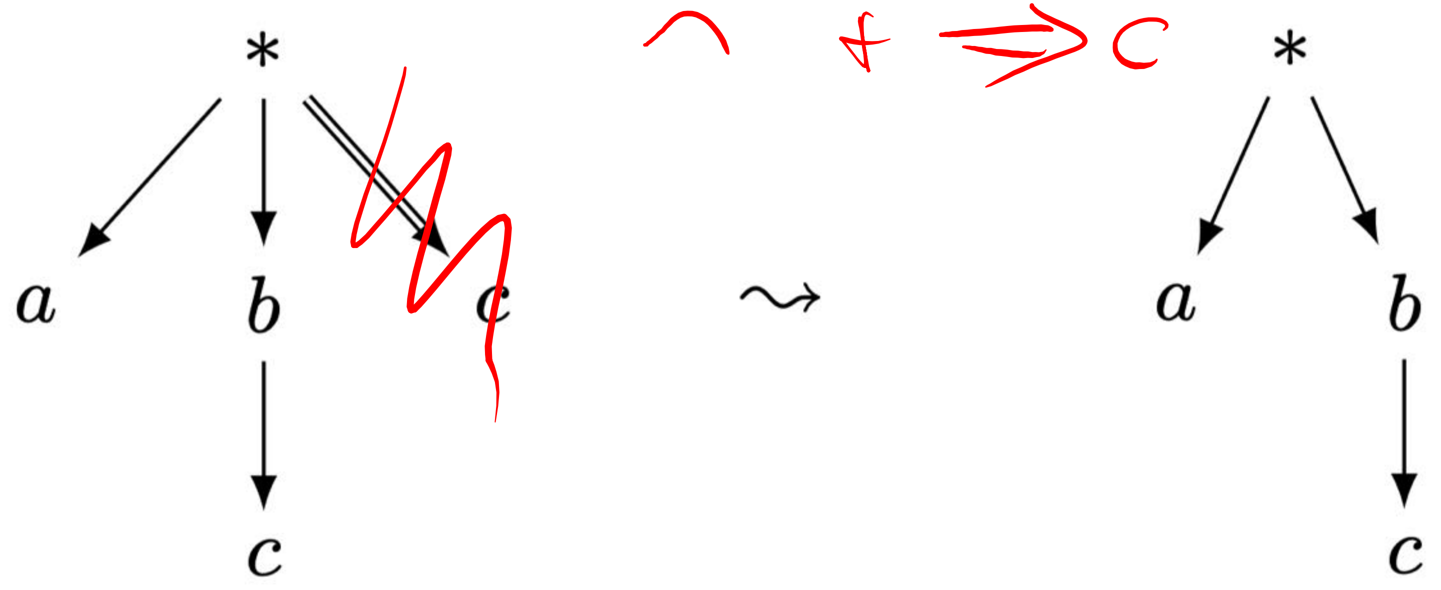
From: “Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data” by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.

Optimizing tree patterns

* → b → c

∩ * ⇒ c

$\Downarrow$ {1, 2} ~~{1, 2, 3}~~



A ∩ ~~B~~

ab ~~⊇ a~~

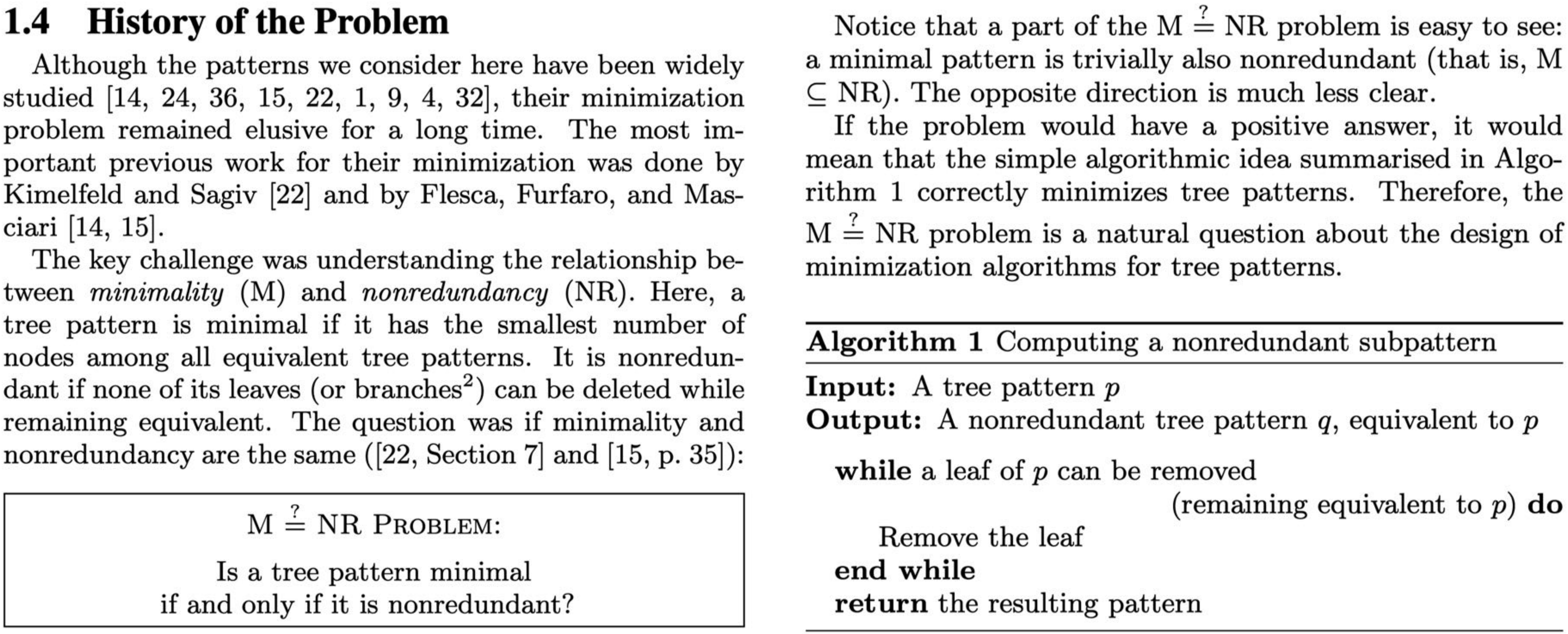
TREE PATTERN MINIMIZATION

Given: A tree pattern p and $k \in \mathbb{N}$

Question: Is there a tree pattern q, equivalent to p, such that its size is at most k?

From: “Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data” by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.

Minimality =? Nonredundancy



1.4 History of the Problem

Although the patterns we consider here have been widely studied [14, 24, 36, 15, 22, 1, 9, 4, 32], their minimization problem remained elusive for a long time. The most important previous work for their minimization was done by Kimelfeld and Sagiv [22] and by Flesca, Furfaro, and Masciari [14, 15].

The key challenge was understanding the relationship between *minimality* (M) and *nonredundancy* (NR). Here, a tree pattern is minimal if it has the smallest number of nodes among all equivalent tree patterns. It is nonredundant if none of its leaves (or branches[2]) can be deleted while remaining equivalent. The question was if minimality and nonredundancy are the same ([22, Section 7] and [15, p. 35]):

$\mathrm{M} \overset{?}{=} \mathrm{NR}$ PROBLEM:

Is a tree pattern minimal
if and only if it is nonredundant?

Notice that a part of the $\mathrm{M} \overset{?}{=} \mathrm{NR}$ problem is easy to see: a minimal pattern is trivially also nonredundant (that is, $\mathrm{M} \subseteq \mathrm{NR}$). The opposite direction is much less clear.

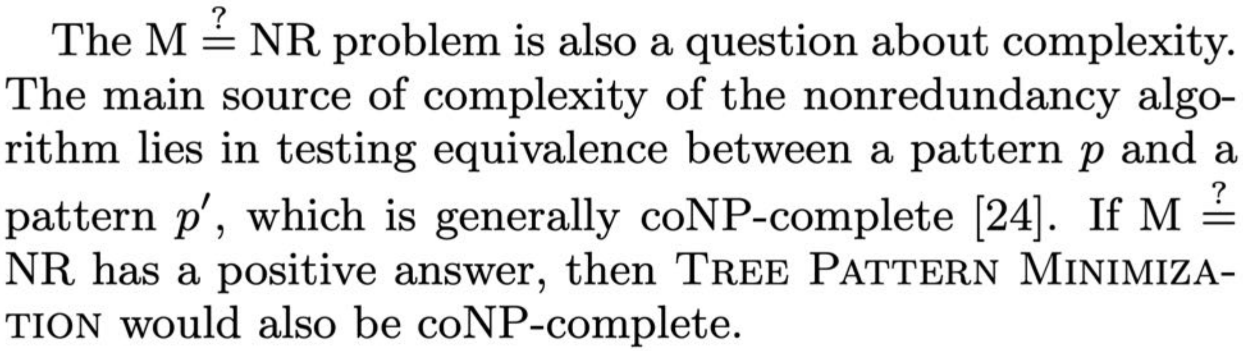
If the problem would have a positive answer, it would mean that the simple algorithmic idea summarised in Algorithm 1 correctly minimizes tree patterns. Therefore, the $\mathrm{M} \overset{?}{=} \mathrm{NR}$ problem is a natural question about the design of minimization algorithms for tree patterns.

Algorithm 1 Computing a nonredundant subpattern

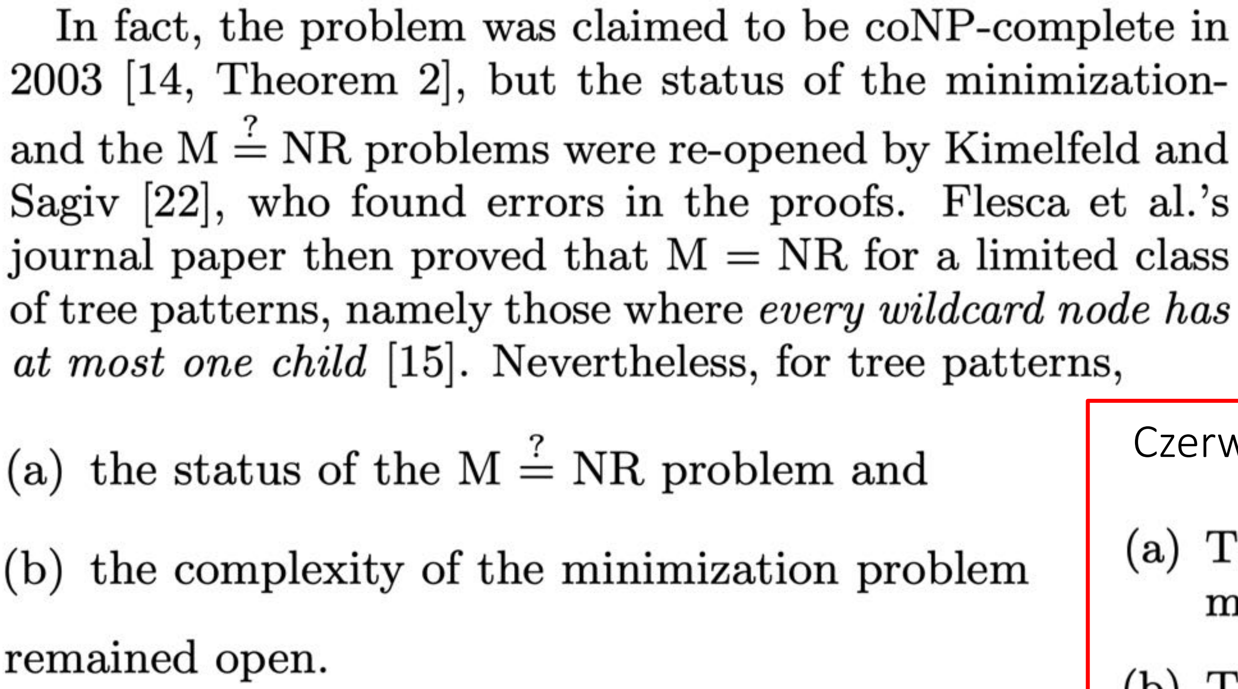
Input: A tree pattern p
Output: A nonredundant tree pattern q, equivalent to p

```
while a leaf of $p$ can be removed
                    (remaining equivalent to $p$) do
    Remove the leaf
end while
return the resulting pattern
```

From: "Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data" by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.



The M $\overset{?}{=}$ NR problem is also a question about complexity. The main source of complexity of the nonredundancy algorithm lies in testing equivalence between a pattern p and a pattern p', which is generally coNP-complete [24]. If M $\overset{?}{=}$ NR has a positive answer, then TREE PATTERN MINIMIZATION would also be coNP-complete.



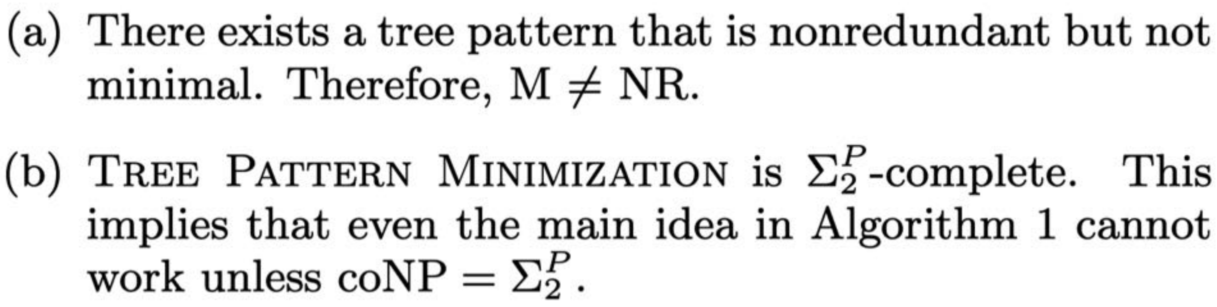
In fact, the problem was claimed to be coNP-complete in 2003 [14, Theorem 2], but the status of the minimization- and the M $\overset{?}{=}$ NR problems were re-opened by Kimelfeld and Sagiv [22], who found errors in the proofs. Flesca et al.'s journal paper then proved that M = NR for a limited class of tree patterns, namely those where *every wildcard node has at most one child* [15]. Nevertheless, for tree patterns,

(a) the status of the M $\overset{?}{=}$ NR problem and

(b) the complexity of the minimization problem

remained open.

Czerwinski, Martens, Niewerth, Parys [PODS 2016}

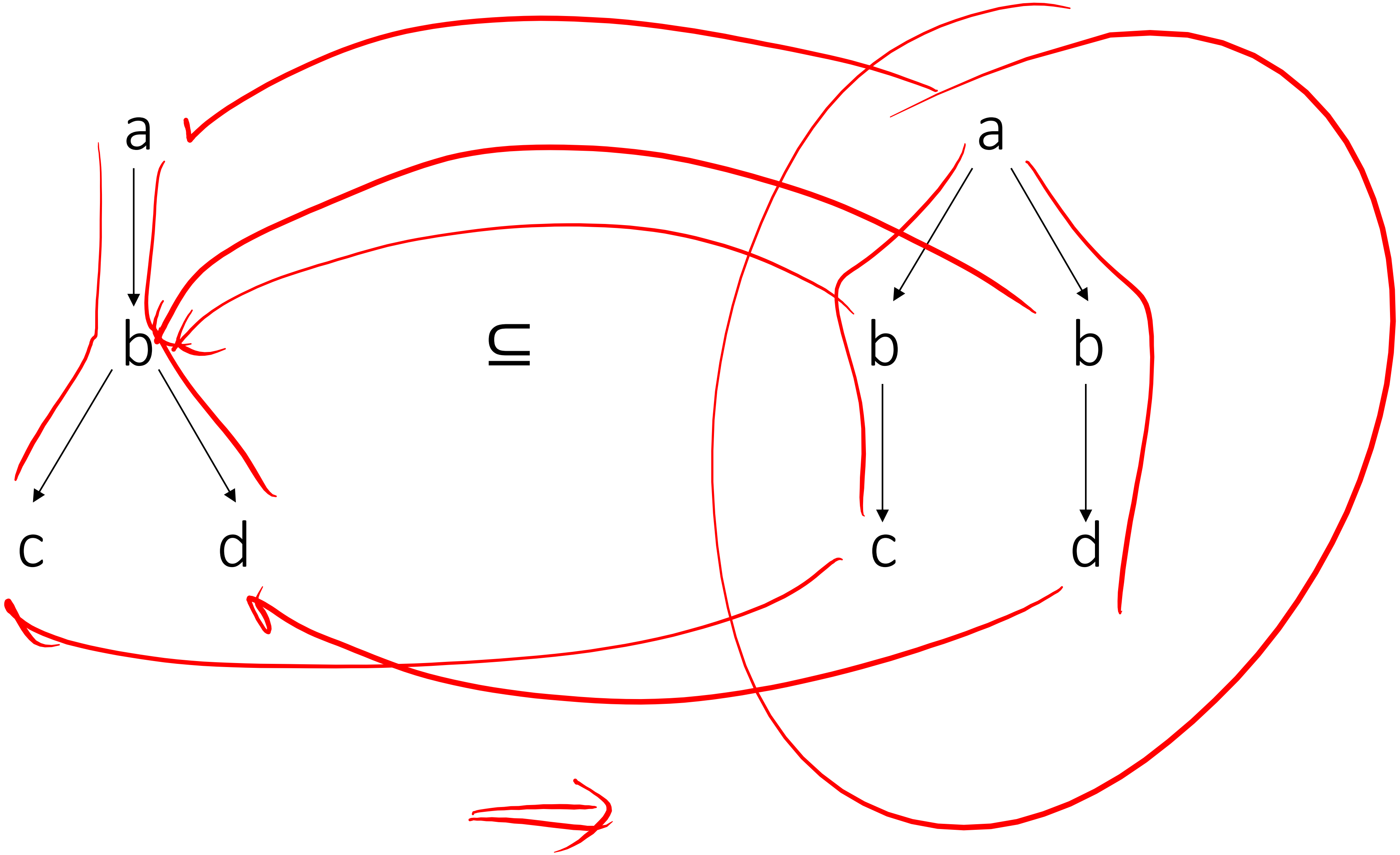


(a) There exists a tree pattern that is nonredundant but not minimal. Therefore, M $\neq$ NR.

(b) TREE PATTERN MINIMIZATION is Σ_2^P-complete. This implies that even the main idea in Algorithm 1 cannot work unless coNP = Σ_2^P.

From: "Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data" by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.

Tree pattern containment



From: "Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data" by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.

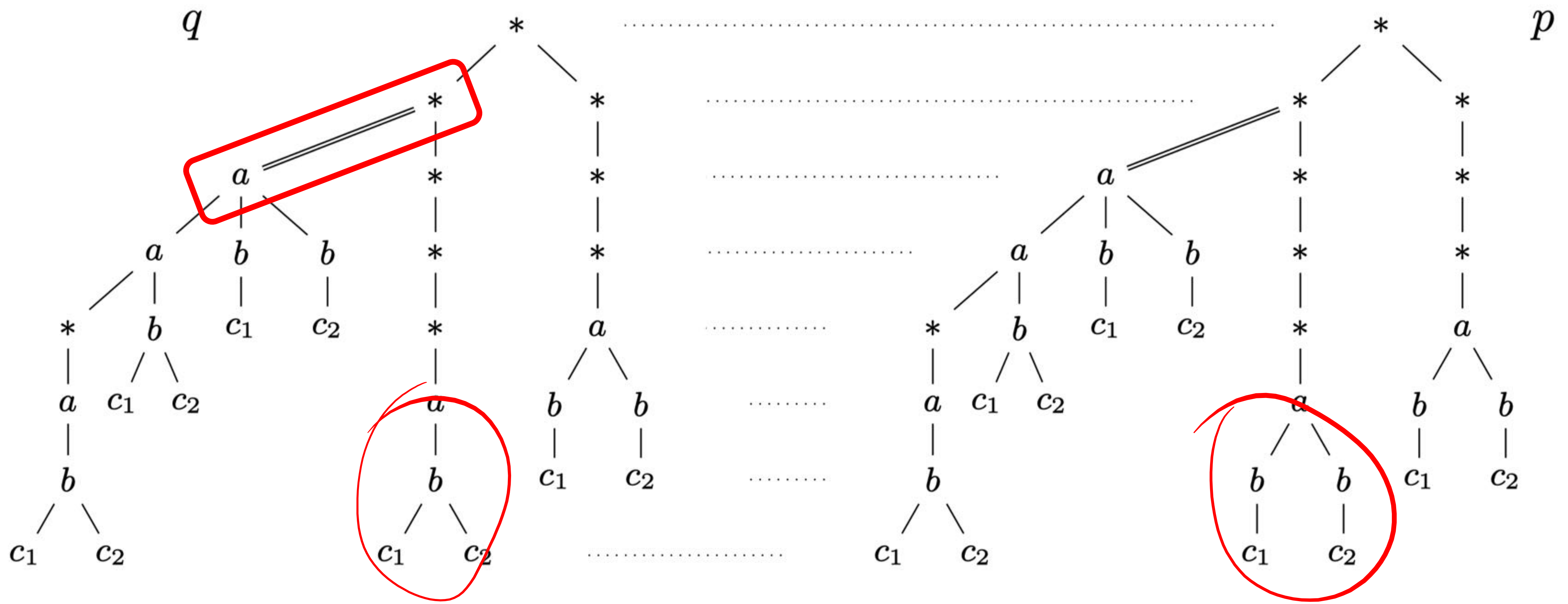


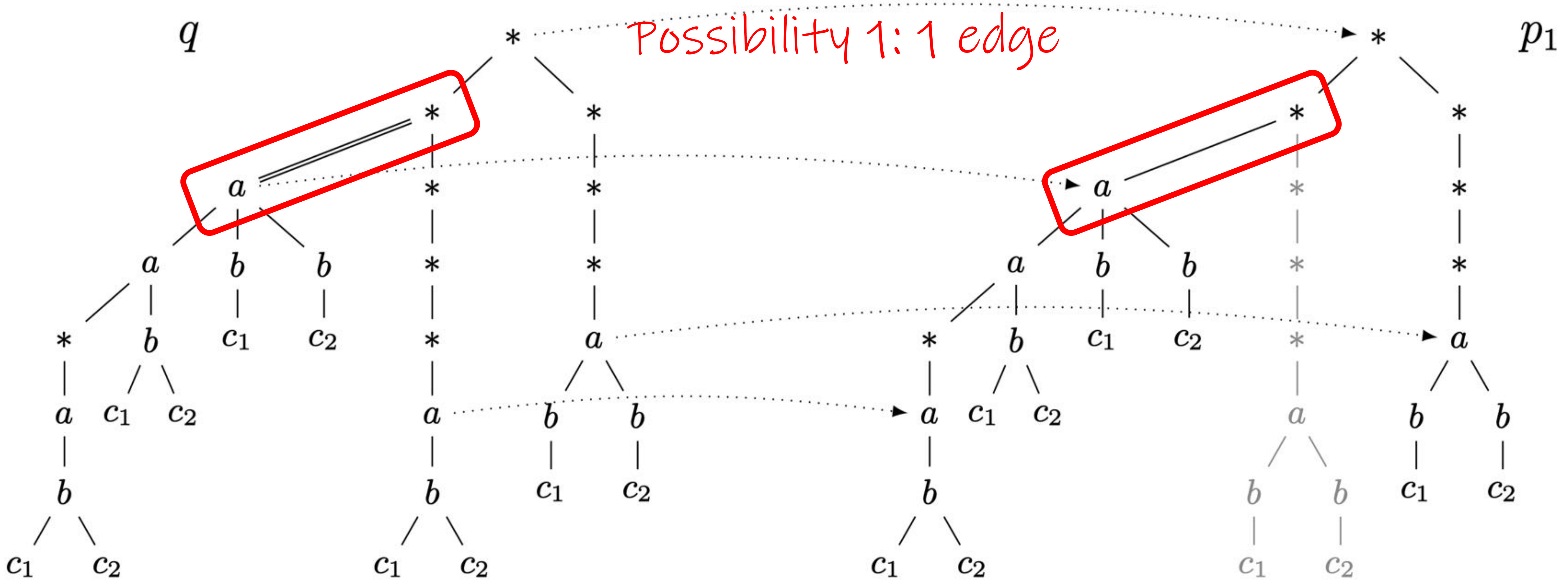
Figure 7: A non-redundant tree pattern p (right) and an equivalent tree pattern q that is smaller (left)

$q \subseteq p$ from previous argument. To be shown: $q \supseteq p$

To be shown $q \supseteq p$: (idea: whenever p matches, then also q)

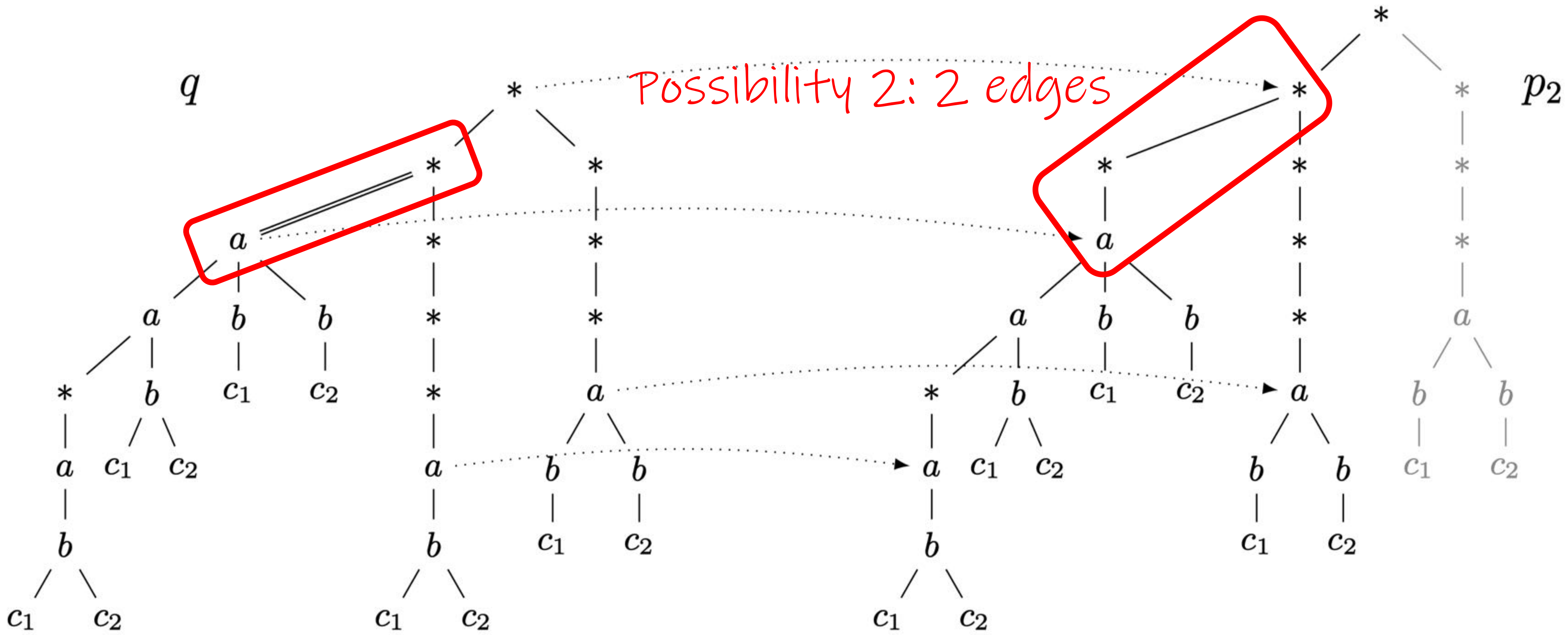
Idea: a=⋆ can be matched in 3 ways in a graph

From: "Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data" by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.



(a) How q can be matched if p_1 can be matched

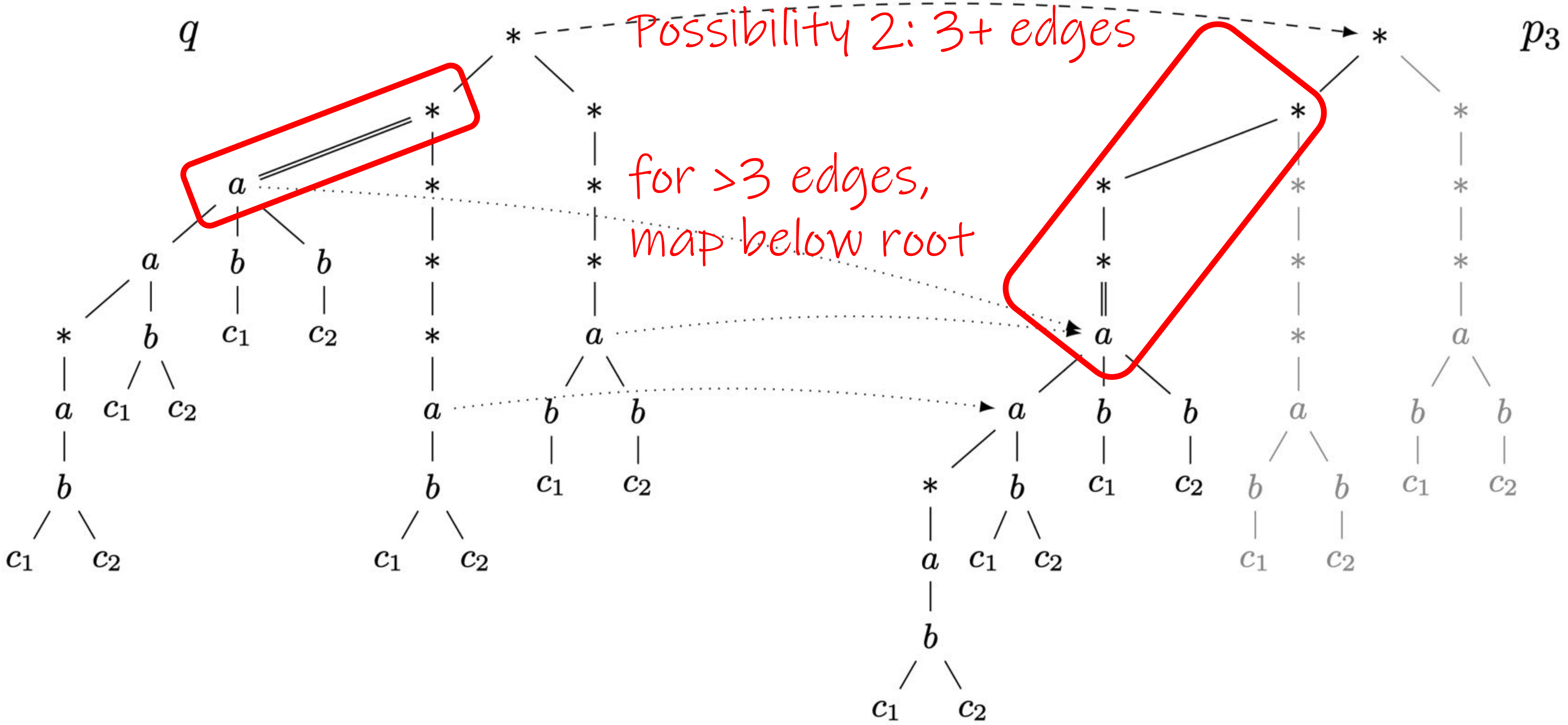
From: “Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data” by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.



Possibility 2: 2 edges

(b) How q can be matched if p_2 can be matched

From: “Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data” by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.

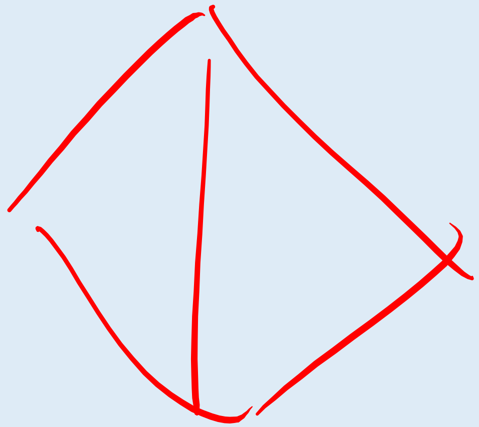


Possibility 2: 3+ edges

for >3 edges,
map below root

(c) How q can be matched if p_3 can be matched

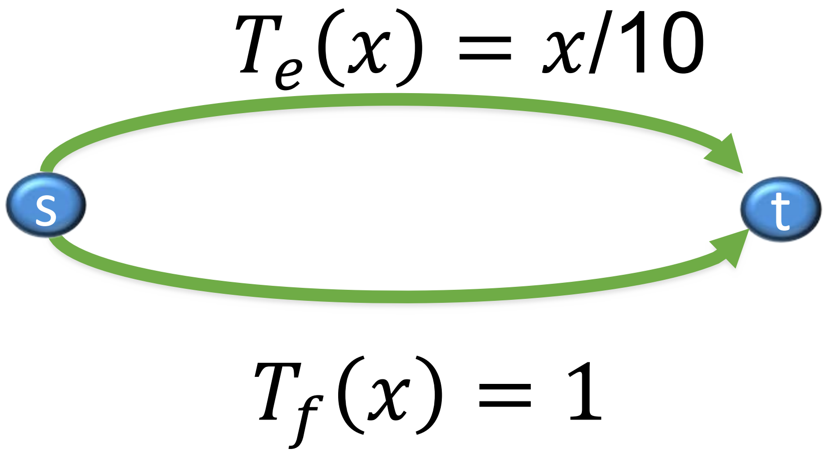
From: “Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data” by Czerwinski, Martens, Niewerth, Parys. SIGMOD record 2017.



Braess' paradox

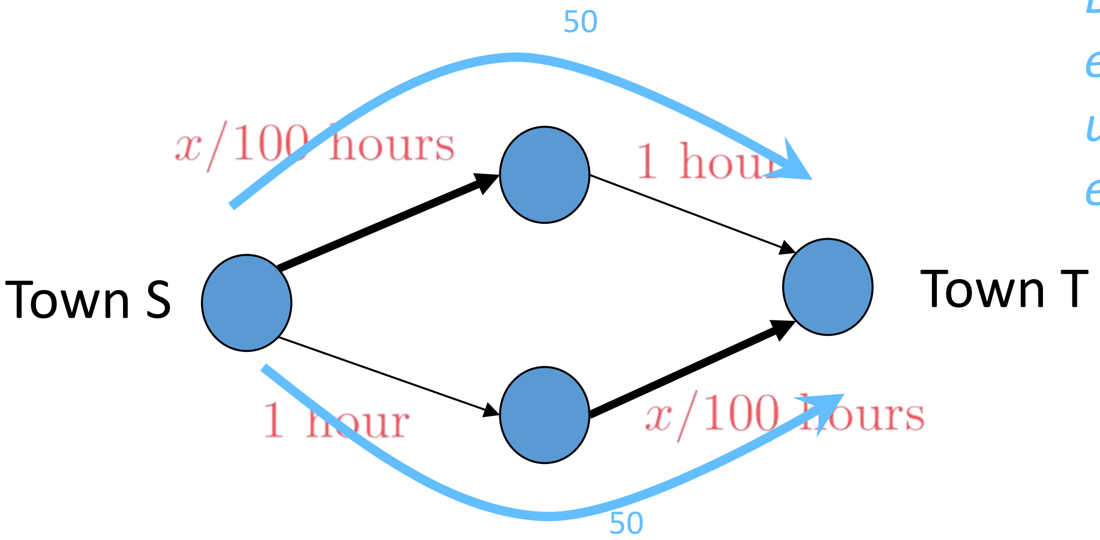
Routing networks

- Traffic wants to flow from some source to some destination
  - Here, the people at s want to drive to t
- Edges have latency or delay



  - Latency of upper edge *e* depends on how many choose it (e.g. in hours)
  - Latency of lower edge *f* always 1 hour

Traffic Routing



Delay is 1.5 hours for everybody at the unique Nash equilibrium

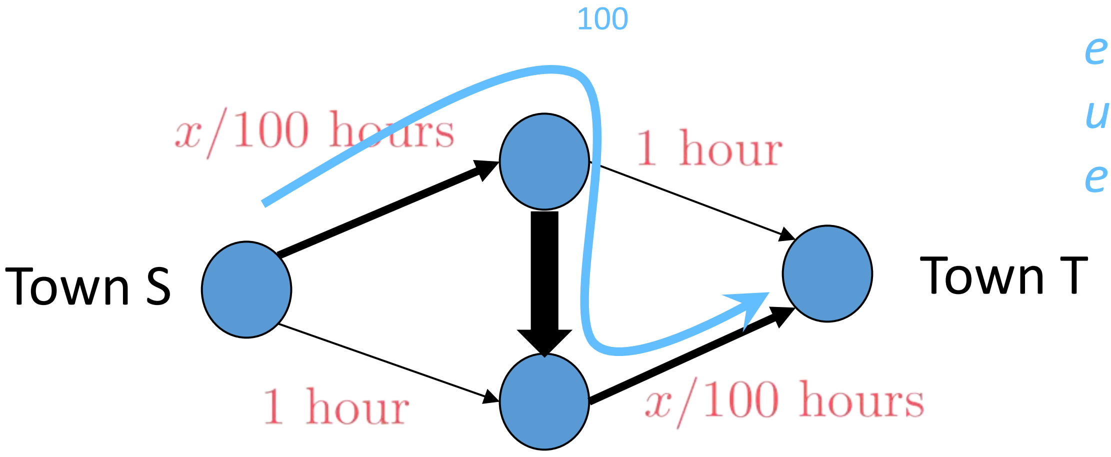
Suppose 100 drivers leave from town S towards town T.

Every driver wants to minimize her own travel time.

What is the traffic on the network?

In any unbalanced traffic pattern, all drivers on the most loaded path have incentive to switch their path.

Traffic Routing



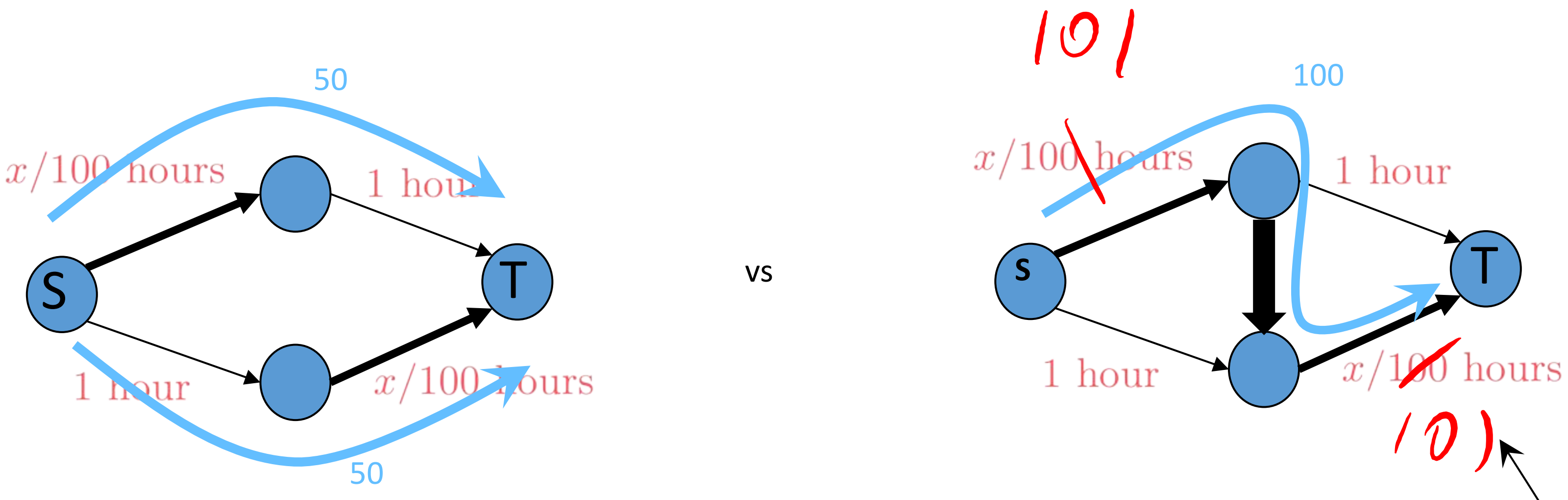
Delay is 2 hours for everybody at the unique Nash equilibrium

A benevolent governor builds a superhighway connecting the short roads of the network.

What is the traffic on the network now?

No matter what the other drivers are doing it is always better for me to follow the zig-zag path.

Traffic Routing



Adding a fast road on a road-network is not always a good idea!

$4/3$

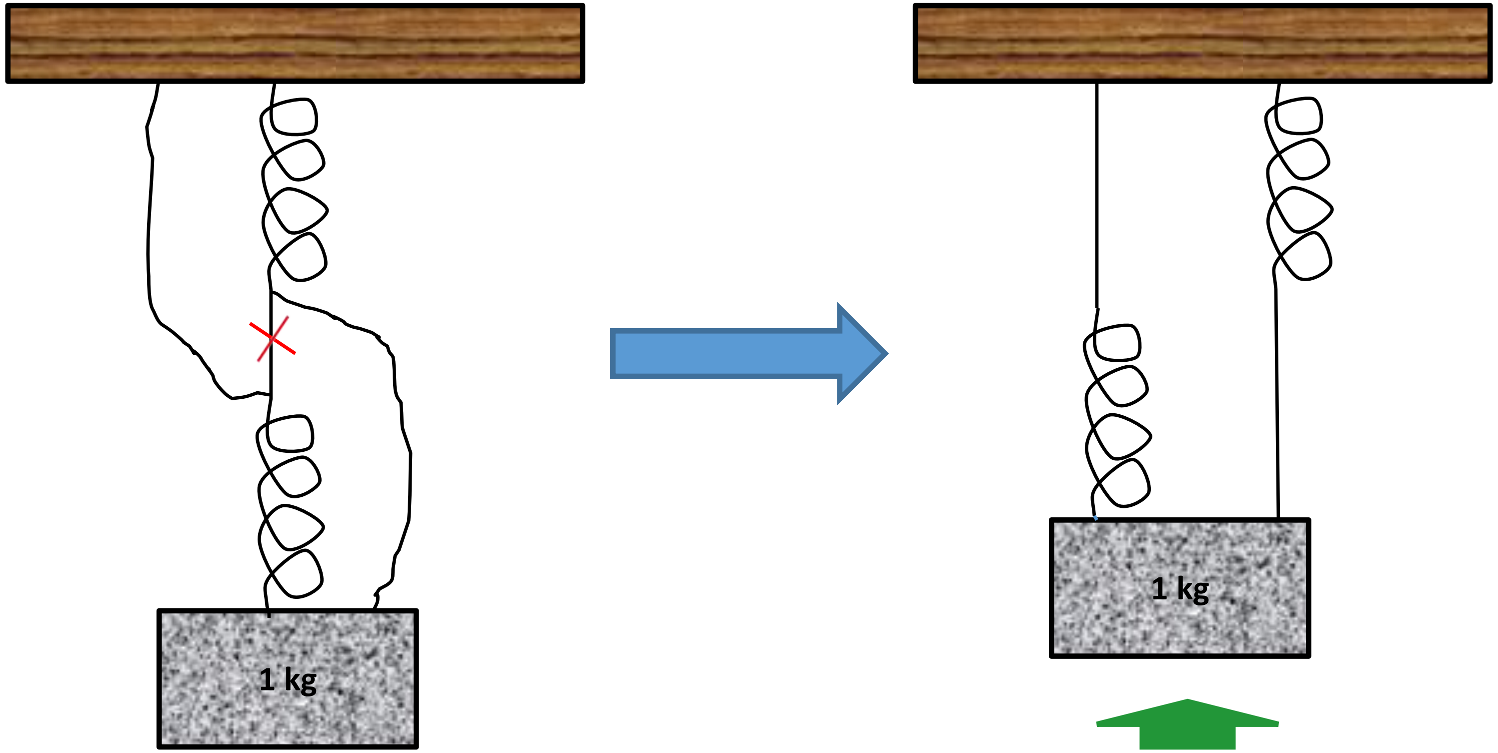
Braess's paradox

In the RHS network there exists a traffic pattern where all players have delay 1.5 hours.

$$\mathrm{PoA} = \frac{\text{performance of system in worst Nash equilibrium}}{\text{optimal performance if drivers did not decide on their own}}$$

Price of Anarchy: measures the loss in system performance due to free-will

A physical representation of the Braess Paradox



Source: Balázs Sziklai

Pointers to related work

- Czerwinski, Martens, Niewerth, Parys. *Optimizing Tree Patterns for Querying Graph- and Tree-Structured Data*. SIGMOD Record 2017. https://sigmodrecord.org/publications/sigmodRecord/1703/pdfs/06_Optimizing_RH_Czerwinski.pdf
- Czerwinski, Martens, Niewerth, Parys. Minimization of Tree Patterns. JACM 2018. https://doi.org/10.1145/3180281
- Braess. *Über ein Paradoxon aus der Verkehrsplanung*. Unternehmensforschung 1968. https://doi.org/10.1007/BF01918335
- Roughgarden, Tardos. *How bad is selfish routing?* JACM 2002. https://doi.org/10.1145/506147.506153