Factorized databases III

Jakub Zavodny (University of Oxford, UK)

DATA ANALYSIS AND MODELING LAB
Palacky University, Olomouc, Czech Republic

INVESTMENTS IN EDUCATION DEVELOPMENT

Factorised Databases 3.

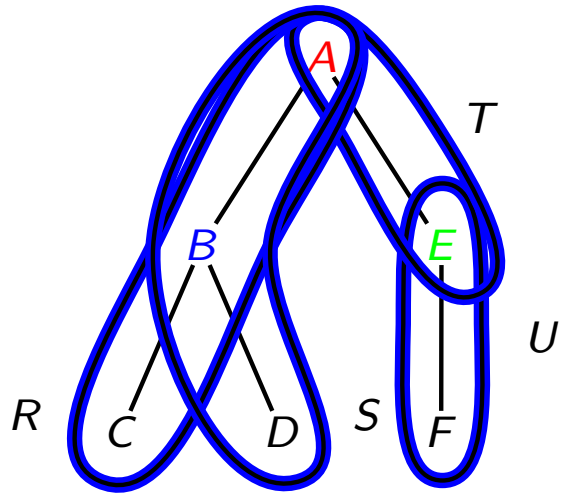
Recent Developments: Factorisations with Pointers

Jakub Závodný, University of Oxford

based on joint work with PhD supervisor Dan Olteanu

Factorisations

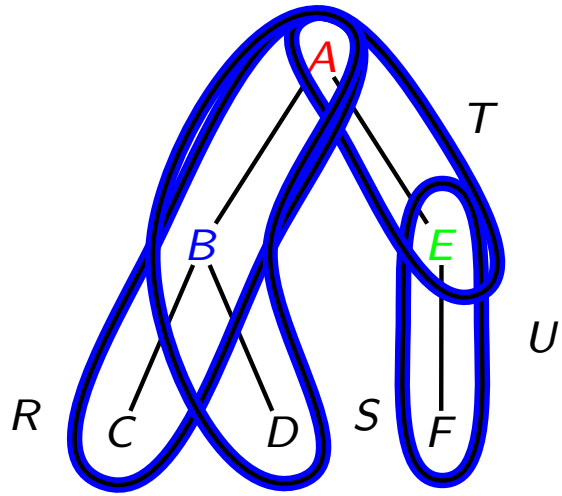
$$Q(A, B, C, D, E, F) = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, E) \bowtie U(E, F)$$



$$\bigcup_{a \in A} \left(\langle a \rangle \times \bigcup_{b \in B} \left(\langle b \rangle \times \left(\bigcup_{c \in C} \langle c \rangle \right) \times \left(\bigcup_{d \in D} \langle d \rangle \right) \right) \times \bigcup_{e \in E} \left(\langle e \rangle \times \left(\bigcup_{f \in F} \langle f \rangle \right) \right) \right)$$

Repeating Subexpressions in Factorisations

$$Q(A, B, C, D, E, F) = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, E) \bowtie U(E, F)$$



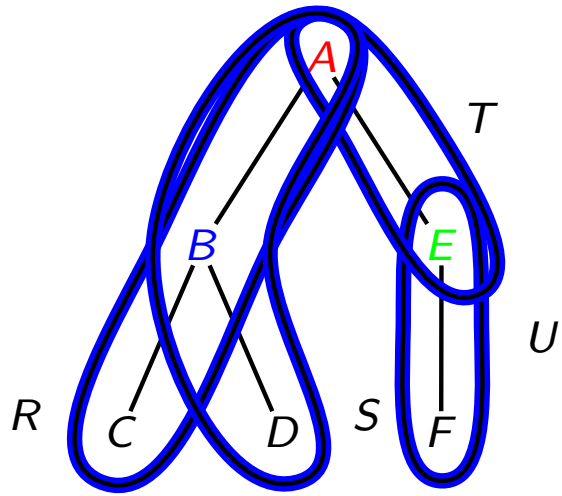
$$\bigcup_{a \in A} \left(\langle a \rangle \times \bigcup_{b \in B} \left(\langle b \rangle \times \left(\bigcup_{c \in C} \langle c \rangle \right) \times \left(\bigcup_{d \in D} \langle d \rangle \right) \right) \times \bigcup_{e \in E} \left(\langle e \rangle \times \left(\bigcup_{f \in F} \langle f \rangle \right) \right) \right)$$

The entire expression $\left(\bigcup_{f \in F} \langle f \rangle \right)$ depends only on the preceding $\langle e \rangle$.

Given $\langle e \rangle$, the expression $\left(\bigcup_{f \in F} \langle f \rangle \right)$ is same for each $\langle a \rangle$.

Using Pointers in Factorisations

$$Q(A, B, C, D, E, F) = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, E) \bowtie U(E, F)$$



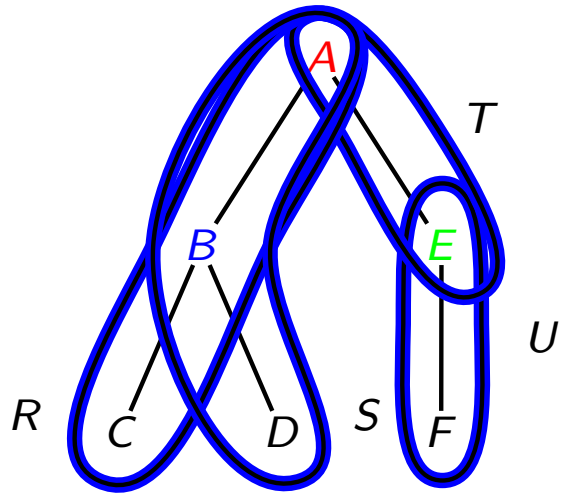
Store the expression $\left(\bigcup_{f \in F} \langle f \rangle\right)$ once for each $\langle e \rangle$ and use pointers!

$$\bigcup_{a \in A} \left(\langle a \rangle \times \bigcup_{b \in B} \left(\langle b \rangle \times \left(\bigcup_{c \in C} \langle c \rangle \right) \times \left(\bigcup_{d \in D} \langle d \rangle \right) \right) \times \bigcup_{e \in E} \left(\langle e \rangle \times *U_e \right) \right)$$

$$\left\{ U_e = \bigcup_{f \in F_e} \langle f \rangle \right\}$$

Using Pointers in Factorisations

$$Q(A, B, C, D, E, F) = R(A, B, C) \bowtie S(A, B, D) \bowtie T(A, E) \bowtie U(E, F)$$



$$\bigcup_{a \in A} \left(\langle a \rangle \times \bigcup_{b \in B} \left(\langle b \rangle \times \left(\bigcup_{c \in C} \langle c \rangle \right) \times \left(\bigcup_{d \in D} \langle d \rangle \right) \right) \times \bigcup_{e \in E} \left(\langle e \rangle \times * U_e \right) \right)$$

$$\left\{ U_e = \bigcup_{f \in F_e} \langle f \rangle \right\}$$

Total size of factorisation with pointers is $O(|\mathbf{D}|)$.

Flat vs. Factorisation vs. Factorisation with Pointers

For a conjunctive query Q,

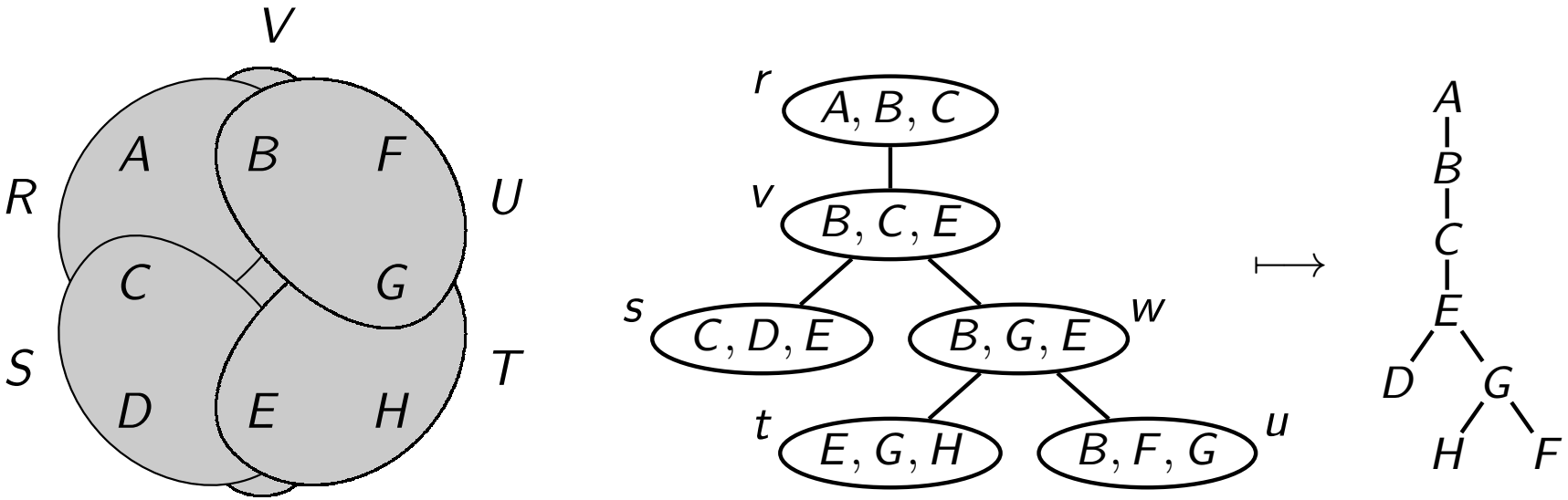
- For any $\mathbf{D}$, $|Q(\mathbf{D})|$ is $O(|\mathbf{D}|^{\rho^*(Q)})$. [AGM'08]
- For any $\mathbf{D}$, $Q(\mathbf{D})$ has factorisation of size $O(|\mathbf{D}|^{s(Q)})$. [OZ'11]
- For any $\mathbf{D}$, $Q(\mathbf{D})$ has factorisation with pointers of size $O(|\mathbf{D}|^{s^\uparrow(Q)})$. [OZ'13]

- $\rho^*(Q)$ = fractional edge cover number of the **entire query**.
- $s(Q)$ = fractional edge cover number of **root-to-leaf paths in best f-tree**.
- $s^\uparrow(Q)$ = fractional edge cover number of **dependency paths in best f-tree**.

$$1 \leq s^\uparrow(Q) \leq s(Q) \leq \rho^*(Q) \leq |Q|$$

(Hyper)Tree Decompositions

F-trees are closely related to tree decompositions and path decompositions.



$s^{\uparrow}(Q) =$ fractional hypertree width of the hypergraph of Q

$s(Q) \geq$ fractional hyperpath width of the hypergraph of Q

Road Map of Query Decomposition Parameters

- $\rho^*(Q)$: size bounds for results of Q.
- $s(Q)$: size bounds for factorisations.
- $s^{\uparrow}(Q)$: size bounds for factorisations with pointers.
- $\mathrm{fhw}(Q)$: fractional hypertree width
- $\mathrm{fhpw}(Q)$: fractional hyperpath width

$$1 \leq \underbrace{s^{\uparrow}(Q) = \mathrm{fhw}(Q) \leq \mathrm{fhpw}(Q) \leq s(Q)}_{\text{factor } O(\log |Q|)} \leq \rho^*(Q) \leq |Q|$$

Each $\leq$ can express a gap of any size as permitted by other inequalities.

Factorised Databases 3.

Future Directions: Instance-based Factorisation

Jakub Závodný, University of Oxford

based on joint work with PhD supervisor Dan Olteanu

Instance-based Factorisation

All previous work was about joins / query results.

Predictable structure $\Rightarrow$

- factorisable using f-trees.
- nice size bounds.
- allows for fast querying.

What about general relations?

Given a relation R, find a *good* factorisation.

- over f-tree?
- as small as possible?
- allowing for fast querying?

Instance-based Factorisation

chain	country
Tesco	Czech Republic
Tesco	Slovakia
Tesco	Hungary
Tesco	Poland
Tesco	UK
Tesco	Spain
Lidl	Czech Republic
Lidl	Slovakia
Lidl	Hungary
Lidl	Poland
Lidl	UK
Aldi	Hungary
Aldi	Poland
Aldi	UK
Aldi	Spain

$$
\begin{aligned}
\mapsto \quad & (\langle \mathit{Tesco} \rangle \cup \langle \mathit{Lidl} \rangle) \times (\langle \mathit{CzechRep} \rangle \cup \langle \mathit{Slovakia} \rangle) \cup \\
& (\langle \mathit{Tesco} \rangle \cup \langle \mathit{Lidl} \rangle \cup \langle \mathit{Aldi} \rangle) \times (\langle \mathit{Hungary} \rangle \cup \langle \mathit{Poland} \rangle \cup \langle \mathit{UK} \rangle) \cup \\
& (\langle \mathit{Tesco} \rangle \cup \langle \mathit{Aldi} \rangle) \times \langle \mathit{Spain} \rangle
\end{aligned}
$$

Instance-based Factorisation

Smallest possible number of products $\equiv$ biclique cover.

	CzechRep	Slovakia	Hungary	Poland	Spain	UK
Tesco	✓	✓	✓	✓	✓	✓
Lidl	✓	✓	✓	✓		✓
Aldi			✓	✓	✓	✓

$$(\langle \mathit{Tesco} \rangle \cup \langle \mathit{Lidl} \rangle) \times (\langle \mathit{CzechRep} \rangle \cup \langle \mathit{Slovakia} \rangle) \cup$$
$$(\langle \mathit{Tesco} \rangle \cup \langle \mathit{Lidl} \rangle \cup \langle \mathit{Aldi} \rangle) \times (\langle \mathit{Hungary} \rangle \cup \langle \mathit{Poland} \rangle \cup \langle \mathit{UK} \rangle) \cup$$
$$(\langle \mathit{Tesco} \rangle \cup \langle \mathit{Aldi} \rangle) \times \langle \mathit{Spain} \rangle$$

Biclique cover is NP-hard.

Instance-based Factorisation

	CzechRep	Slovakia	Hungary	Poland	Spain	UK
Tesco	✓	✓	✓	✓	✓	✓
Lidl	✓	✓	✓	✓		✓
Aldi			✓	✓	✓	✓

$$(\langle \mathit{Tesco} \rangle \cup \langle \mathit{Lidl} \rangle) \times (\langle \mathit{CzechRep} \rangle \cup \langle \mathit{Slovakia} \rangle) \cup$$
$$(\langle \mathit{Tesco} \rangle \cup \langle \mathit{Lidl} \rangle \cup \langle \mathit{Aldi} \rangle) \times (\langle \mathit{Hungary} \rangle \cup \langle \mathit{Poland} \rangle \cup \langle \mathit{UK} \rangle) \cup$$
$$(\langle \mathit{Tesco} \rangle \cup \langle \mathit{Aldi} \rangle) \times \langle \mathit{Spain} \rangle$$

Heuristics:

- Find large, maximal bicliques (formal concepts).
- Remove and factorise the rest.

\+ Adaptations for higher dimensions.
\+ P-time factorisation algorithm for exact products.

Thank you!