

Spoilt for Choice: Full First-Order Hierarchical Decompositions

*This research is supported by the Marsden fund council from Government funding,
administered by the Royal Society of New Zealand.*

Sebastian Link

*Centre for Logic, Language, Information and Computation
School of Information Management
Victoria University of Wellington
New Zealand*



Relational Databases

- ▶ finite structure over a relational signature, but named perspective
- ▶ schema: finite set R of attributes (with domains), e.g.

CONFERENCE = {Conf_Name, Talk, Sight, Food}

- ▶ database: relation r over R , e.g.

Conf_Name	Talk	Sight	Food
WoLLIC09	Knots	Fuji-san	Nigiri Toro
WoLLIC09	Ontologies	O-torii	Okonomiyaki
WoLLIC09	Knots	O-torii	Nigiri Toro
WoLLIC09	Knots	O-torii	Okonomiyaki
WoLLIC09	Knots	Fuji-san	Okonomiyaki
WoLLIC09	Ontologies	Fuji-san	Nigiri Toro
WoLLIC09	Ontologies	Fuji-san	Okonomiyaki
WoLLIC09	Ontologies	O-torii	Nigiri Toro

- ▶ attribute A with $dom(A)$, attribute sets: X, X_i, Y, Y_i, Z ($\subseteq R$)
- ▶ $r[X], r \bowtie r'$

Full First-Order Hierarchical Decompositions

- ▶ FOHDs $X : [Y_1 \mid \cdots \mid Y_k]$ over R with partition $Y_1 \cdots Y_k = R - X$
for example, Conf_Name: [Talk | Sight | Food]
- ▶ $\models_r X : [Y_1 \mid \cdots \mid Y_k]$ precisely when r is decomposable into k of its projections without loss of information

$$r = (\cdots (r[XY_1] \bowtie r[XY_2]) \bowtie \cdots \bowtie r[XY_k])$$

- ▶ Conf_Name: [Talk | Sight | Food] satisfied by previous database
- ▶ independent info units (efficient updating vs. querying)

Conf_Name	Talk
WoLLIC09	Knots
WoLLIC09	Ontologies

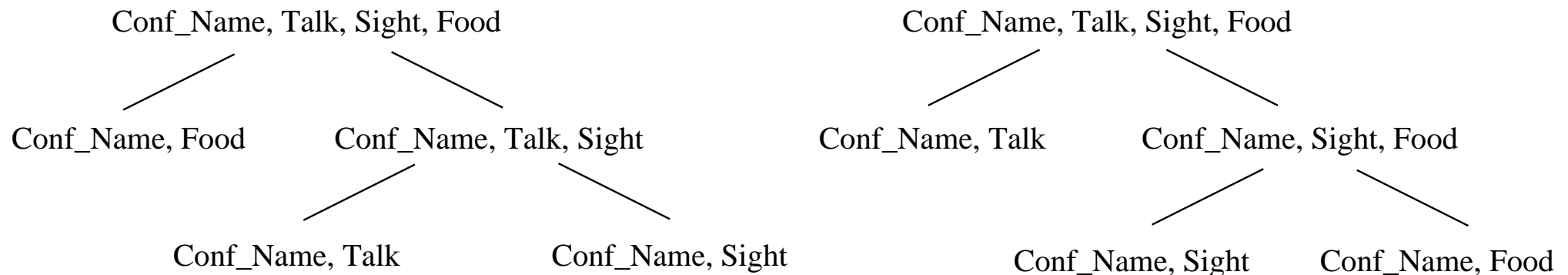
Conf_Name	Sight
WoLLIC09	Fuji-san
WoLLIC09	O-torii

Conf_Name	Food
WoLLIC09	Nigiri toro
WoLLIC09	Okonomiyaki

- ▶ normalization: decompose into info units
- ▶ de-normalization: put info units back together
- ▶ no theory based on most common types of queries and updates

Hierarchical Decompositions

- ▶ FOHDs determine stepwise decompositions dividing each unit into two new ones
- ▶ apply Conf_Name: [Food | Talk | Sight], and Conf_Name: [Talk | Sight | Food]



- ▶ FOHDs determine two things at once:
 - ↪ independent information units
 - ↪ the order of decomposition (some permutation of info units)
- ▶ in practice: database designer determines order according to some heuristic (at best!)

The Implication Problem

- ▶ $\Sigma \cup \{\varphi\}$ set of FOHDs over R :
 $\Sigma \models \varphi$ iff $\forall r \subseteq \text{dom}(R)$: if $\models_r \Sigma$, then $\models_r \varphi$
- ▶ finite and infinite implication coincide
- ▶ given arbitrary R , $\Sigma \cup \{\varphi\}$ decide if $\Sigma \models \varphi$
- ▶ significant:
 - ↳ determines atomic info units: every info unit union of atoms
 - ↳ reveals new decompositions that best fit queries and updates
- ▶ Conf_Name: [Sight | Talk, Food] and Conf_Name: [Food | Talk, Sight]
 imply Conf_Name: [Talk | Food | Sight]
 ↳ therefore new info unit: Conf_Name, Sight, Food

An Axiomatisation for FOHDs

$\frac{X : [Y_1 \mid \cdots \mid Y_k]}{XZ : [Y_1 - Z \mid \cdots \mid Y_k - Z]}$ <p style="text-align: center;">(augmentation, \mathcal{A})</p> $\frac{X : [Y_1 \mid \cdots \mid Y_k]}{X : [Y_1 Y_2 \mid \cdots \mid Y_k]}$ <p style="text-align: center;">(merging, \mathcal{M}')</p>	$\frac{\overline{\emptyset : [R]}}{\text{(universal, } \mathcal{U})}$ $\frac{X : [X_1 \mid X_2] \quad XX_1 : [Y_1 \mid \cdots \mid Y_k]}{X : [Y_1 \mid \cdots \mid Y_k \mid X_1]}$ <p style="text-align: center;">(transitivity, \mathcal{T}')</p> $\frac{X : [Y_1 \mid \cdots \mid Y_k]}{X : [Y_{\pi(1)} \mid \cdots \mid Y_{\pi(k)}]}$ <p style="text-align: center;">(permutation, \mathcal{P})</p>
--	--

- ▶ Theorem: The set $\mathfrak{F}' = \{\mathcal{U}, \mathcal{A}, \mathcal{T}', \mathcal{M}', \mathcal{P}\}$ forms a finite axiomatisation for the implication of full first-order hierarchical decompositions.
- ▶ key to proof: the split rule

$$\frac{X : [X_1 \mid X_2] \quad X : [Y_1 \mid \cdots \mid Y_k]}{X : [Y_1 \mid \cdots \mid Y_j \cap X_i \mid Y_j - X_i \mid \cdots \mid Y_k]}$$

↪ generalisation of result by Beeri, Fagin, Howard who show binary case ($k=2$)

The Role of the Permutation Rule

- ▶ intuition of \mathcal{P} : no new information units, only change of order
- ▶ design process:
 - (i) inference engine determines atomic information units
 - (ii) database designer chooses decomposition order
- ▶ adequate axiomatisation should mimic this behaviour:
 - ↪ permutation rule only applied at final inference step
- ▶ Example:

$$\begin{array}{c}
 \frac{\text{Conf_Name} : [\text{Talk}, \text{Food} \mid \text{Sight}]}{\mathcal{P} : \text{Conf_Name} : [\text{Sight} \mid \text{Talk}, \text{Food}]} \quad \frac{\text{Conf_Name} : [\text{Food} \mid \text{Talk}, \text{Sight}]}{\mathcal{A} : \text{Conf_Name}, \text{Sight} : [\text{Food} \mid \text{Talk}]} \\
 \hline
 \mathcal{T}' : \quad \frac{\text{Conf_Name} : [\text{Food} \mid \text{Talk} \mid \text{Sight}]}{\mathcal{P} : \text{Conf_Name} : [\text{Food} \mid \text{Sight} \mid \text{Talk}]} \\
 \hline
 \mathcal{M}' : \quad \frac{\text{Conf_Name} : [\text{Food}, \text{Sight} \mid \text{Talk}]}{\mathcal{P} : \text{Conf_Name} : [\text{Talk} \mid \text{Food}, \text{Sight}]}
 \end{array}$$

First Confirmation: An Adequate Axiomatisation

$\frac{X : [Y_1 \mid \cdots \mid Y_k]}{XZ : [Y_1 - Z \mid \cdots \mid Y_k - Z]}$ <p style="text-align: center;">(augmentation, \mathcal{A})</p> $\frac{X : [Y_1 \mid \cdots \mid Y_k]}{X : [Y_1 \mid \cdots \mid Y_i Y_j \mid \cdots \mid Y_k]}$ <p style="text-align: center;">(merging, \mathcal{M})</p>	$\frac{\overline{\emptyset : [R]}}{\text{(universal, } \mathcal{U})}$ $\frac{X : [X_1 \mid X_2] \quad XX_i : [Y_1 \mid \cdots \mid Y_k]}{X : [Y_1 \mid \cdots \mid Y_k \mid X_i]}$ <p style="text-align: center;">(transitivity, \mathcal{T})</p> $\frac{X : [Y_1 \mid \cdots \mid Y_k]}{X : [Y_{\pi(1)} \mid \cdots \mid Y_{\pi(k)}]}$ <p style="text-align: center;">(permutation, \mathcal{P})</p>
--	---

► Theorem:

For every inference γ from Σ by \mathfrak{F} there is an inference ξ from Σ by \mathfrak{F} with:

- γ and ξ infer the same full first-order hierarchical decomposition,
- in ξ the permutation rule \mathcal{P} is applied at the very last step only. □

• Corollary:

For all $X : [Y_1 \mid \cdots \mid Y_k]$ on R we have: $X : [Y_1 \mid \cdots \mid Y_k] \in \Sigma_{\mathfrak{F}}^+$ iff there is some permutation π on $\{1, \dots, k\}$ such that $X : [Y_{\pi(1)} \mid \cdots \mid Y_{\pi(k)}] \in \Sigma_{\{\mathcal{U}, \mathcal{A}, \mathcal{T}, \mathcal{M}\}}^+$.

Adquate Inference of Previous Example

- ▶ Recall inference with \mathcal{F}' :

$$\begin{array}{c}
 \frac{\text{Conf_Name} : [\text{Talk}, \text{Food} \mid \text{Sight}]}{\mathcal{P} : \text{Conf_Name} : [\text{Sight} \mid \text{Talk}, \text{Food}]} \quad \frac{\text{Conf_Name} : [\text{Food} \mid \text{Talk}, \text{Sight}]}{\mathcal{A} : \text{Conf_Name}, \text{Sight} : [\text{Food} \mid \text{Talk}]} \\
 \mathcal{T}' : \quad \frac{\text{Conf_Name} : [\text{Food} \mid \text{Talk} \mid \text{Sight}]}{\mathcal{P} : \text{Conf_Name} : [\text{Food} \mid \text{Sight} \mid \text{Talk}]} \\
 \mathcal{M}' : \quad \frac{\text{Conf_Name} : [\text{Food}, \text{Sight} \mid \text{Talk}]}{\mathcal{P} : \text{Conf_Name} : [\text{Talk} \mid \text{Food}, \text{Sight}]}
 \end{array}$$

- ▶ An inference with \mathcal{F} :

$$\begin{array}{c}
 \text{Conf_Name} : [\text{Talk}, \text{Food} \mid \text{Sight}] \quad \frac{\text{Conf_Name} : [\text{Food} \mid \text{Talk}, \text{Sight}]}{\text{Conf_Name}, \text{Sight} : [\text{Food} \mid \text{Talk}]} \\
 \mathcal{T} : \quad \frac{\text{Conf_Name} : [\text{Food} \mid \text{Talk} \mid \text{Sight}]}{\mathcal{M} : \text{Conf_Name} : [\text{Food}, \text{Sight} \mid \text{Talk}]} \\
 \mathcal{P} : \quad \frac{\text{Conf_Name} : [\text{Talk} \mid \text{Food}, \text{Sight}]}{\text{Conf_Name} : [\text{Talk} \mid \text{Food}, \text{Sight}]}
 \end{array}$$

Second Confirmation: Order-invariant Hierarchical Decompositions

- ▶ OIHDs $X : \{Y_1, \dots, Y_k\}$ over R with partition $Y_1 \cdots Y_k = R - X$
 - ↪ notion of satisfaction as for FOHDs (join associative)
 - ↪ set notation captures all potential permutations
 - ↪ only focus is on inferring all atomic information units
- ▶ Theorem: The set $\mathcal{D} = \{\mathcal{U}, \mathcal{A}, \mathcal{M}, \mathcal{T}\}$ forms a finite axiomatisation for the implication of OIHDs.

$\frac{X : \{Y_1, \dots, Y_k\}}{XZ : \{Y_1 - Z, \dots, Y_k - Z\}}$ <p style="text-align: center;">(augmentation, \mathcal{A})</p> $\frac{}{\emptyset : \{R\}}$ <p style="text-align: center;">(universal axiom, \mathcal{U})</p>	$\frac{X : \{X_1, X_2\} \quad XX_i : \{Y_1, \dots, Y_k\}}{X : \{Y_1, \dots, Y_k, X_i\}}$ <p style="text-align: center;">(transitivity, \mathcal{T})</p> $\frac{X : \{Y_1, \dots, Y_k\}}{X : \{Y_1, \dots, Y_i Y_j, \dots, Y_k\}}$ <p style="text-align: center;">(merging, \mathcal{M})</p>
--	---

Correspondence to Logic

- ▶ data dependencies are certain first-order formulae
 \hookrightarrow the OIHD $A : \{B, C, D\}$ over relation schema R expressed by:

$$\forall a, b, c, d, b', c', d', b'', c'', d'' \\ ((R(a, b, c, d) \wedge R(a, b', c', d') \wedge R(a, b'', c'', d'')) \Rightarrow R(a, b, c', d'')).$$

- ▶ for R let $\mathcal{V}_R = \{V_A : A \in R\}$ denote set of variables
- ▶ for an OIHD $\varphi = X : \{Y_1, \dots, Y_k\}$, let φ' denote:

$$\left(\bigwedge_{A \in X} V_A \right) \Rightarrow \left(\bigvee_{i=1}^k \left(\bigwedge_{B \in Y_i} V_B \right) \right)$$

- ▶ for Σ let $\Sigma' = \{\sigma' : \sigma \in \Sigma\}$
- ▶ Theorem[Sagiv, Fagin, Stott Parker, Delobel]:
 Let R be some relation schema, and let $\Sigma \cup \{\varphi\}$ be a set of binary OIHDs over R . Then $\Sigma \models \varphi$ if and only if $\Sigma' \models \varphi'$

Failed Correspondence

- ▶ for arbitrary OIHDs, i.e. where $k > 2$, the correspondence fails
- ▶ Let $R_k = \{A, B_1, \dots, B_k\}$, and let
 - ↳ $\Sigma_k = \{\sigma_k\}$ where $\sigma_k = A : \{\{B_1, \dots, B_{k-1}\}, B_k\}$, and
 - ↳ $\varphi_k = A : \{B_1, \dots, B_k\}$
- ▶ For $k > 2$: Σ_k does not imply φ_k
 - ↳ $\{(a, b_1, \dots, b_{k-1}, b_k), (a, b'_1, \dots, b'_{k-1}, b_k)\}$
- ▶ Then $\mathcal{V}_{R_k} = \{V_A, V_{B_1}, \dots, V_{B_k}\}$, and
 - ↳ $\Sigma'_k = \{V_A \Rightarrow ((V_{B_1} \wedge \dots \wedge V_{B_{k-1}}) \vee V_{B_k})\}$, and
 - ↳ $\varphi'_k = V_A \Rightarrow (V_{B_1} \vee \dots \vee V_{B_k})$
- ▶ however: Σ'_k does imply φ'_k

Future Work?



Questions?

