Horn Clauses and Functional Dependencies in Complex-value Databases

Sven Hartmann, Sebastian Link

Information Science Research Centre, Dept of Information Systems
Massey University, New Zealand

This research is supported by Marsden Funding, Royal Society of New Zealand.
Relational FDs and Horn Clauses

- consider \texttt{LECTURE} = \{Class, Lecturer, Time, Room\} with \( \Sigma \) of FDs:
  - Class \( \rightarrow \) Lecturer, and Class, Time \( \rightarrow \) Room, and
  - Lecturer, Time \( \rightarrow \) Class, and Room, Time \( \rightarrow \) Class
- FD \( \sigma \): Class, Lecturer, Room \( \rightarrow \) Time is not implied by \( \Sigma \)
  - \{(Databases, H. Simpson, 2:30pm, 3.12), (Databases, H. Simpson, 4:30pm, 3.12)\}
- \textit{Fagin’77}: FD implication is equivalent to Horn clause implication
- view attributes as propositional variables and FDs as Horn clauses
- define \( \theta \) by \( \theta(V) = \text{true} \) iff \( V \in \{\text{Class, Lecturer, Room}\} \)
- \( \theta \) satisfies all Horn clauses in \( \Sigma \) but violates \( \sigma \)
Complex-value Databases

- schema: \texttt{SHOP(Customer, Bag\langle ITEM(Article, Price)\rangle, Discount)}
- instance:
  (Homer, \langle(Donut, 1.5), (Donut, 1.5), (Chocolate, 2), (Chocolate, 2)\rangle, 0)
  (Bart, \langle(Donut, 2), (Donut, 2), (Chocolate, 1.5), (Chocolate, 1.5)\rangle, 1)
- customers with the same bag of items should receive the same discount

\texttt{SHOP(Bag\langle ITEM(Article, Price)\rangle)} \rightarrow \texttt{SHOP(Discount)}

- fundamental problem:
  What other dependencies are implied by those specified?
- major objective:
  Find logical characterisation for implication of dependencies in complex-value databases
Database Schemata: Nested Attributes

- capture characteristics of objects in target database by attributes

\[ N := A \mid \lambda \mid L(N, \ldots, N) \mid L[N] \mid L\{N\} \mid L\langle N \rangle \]

- examples:
  - \text{SHOP}(\text{Customer}, \text{BAG}\langle \text{ITEM}(\text{Article}, \text{Price}), \text{Discount} \rangle)
  - \text{SOCCER}\{\text{MATCH}(\text{Winner}, \text{Loser})\}
  - \text{NUMBERS}[\text{REPRESENTATION}(\text{Prime})]
  - \text{LECTURE}(\text{Class}, \text{Lecturer}, \text{Time}, \text{Room})
Database Instances: Domain Assignment

- extend $dom$ from flat to nested attributes ($dom(\lambda) = \{\text{ok}\}$)
- examples for nested tuples:
  - \text{SHOP}(\text{Customer, Bag}\langle \text{Item} \langle \text{Article, Price} \rangle, \text{Discount} \rangle):
    - (Homer, \langle (\text{Donut, 1.5}), (\text{Donut, 1.5}), (\text{Chocolate, 2}), (\text{Chocolate, 2}) \rangle, 0)
    - (Bart, \langle (\text{Donut, 2}), (\text{Donut, 2}), (\text{Chocolate, 1.5}), (\text{Chocolate, 1.5}) \rangle, 1)
  - \text{SOCCEER}\{\text{MATCH}(\text{Winner, Loser})\}:
    - \{(\text{Denmark, Sweden}), (\text{New Zealand, Australia})\}
    - \{(\text{Mexico, USA}), (\text{Brazil, Argentina}), (\text{Brazil, USA})\}

- RDM: single application of record constructor
- Nested Relational Data Model: record and set constructor
- Object-oriented Data Models: record, set, multiset and list constructor
Subschemata: Subattributes

- recursively replacing attributes by \( \lambda \) gives different layers of info:

- some subattributes of

  \( \text{SHOP}(\text{Customer}, \text{Bag}<\text{Item}(\text{Article, Price}), \text{Discount}) \):

  - \( \text{SHOP}(\lambda, \text{Bag}<\text{Item}(\text{Article, Price}), \text{Discount}) \)
  - \( \text{SHOP}(\text{Customer}, \text{Bag}<\text{Item}(\lambda, \lambda), \text{Discount}) \)
  - \( \text{SHOP}(\lambda, \text{Bag}<\text{Item}(\text{Article, } \lambda), \lambda) \)
  - \( \text{SHOP}(\text{Customer}, \lambda, \text{Discount}) \)

- formally:
  define subattribute relation \( \leq \) on nested attributes (partial order)

- \( \leq \) induces Brouwerian algebra \((\text{Sub}(N), \leq, \sqcup, \sqcap, \forall, \lambda_N)\) on set \( \text{Sub}(N) \) of subattributes on \( N \)
Database Transformations: Projection Function

- Subattributes represent at most as much info as their superattributes
- Formally: for $M \leq N$ there is projection $\pi^N_M : \text{dom}(N) \rightarrow \text{dom}(M)$
- $N = \text{SHOP}($Customer,BAG\langle ITEM(Article,Price)\rangle,Discount) with\n  \begin{align*}
  t &= (\text{Bart},\langle\text{(Donut,2)},(\text{Donut,2}),\text{(Chocolate,1.5)},(\text{Chocolate,1.5})\rangle,1) \\
  M &= \text{SHOP}($Customer,BAG\langle ITEM(\lambda,Price)\rangle,Discount) \notag \\
  \pi^N_M(t) &= (\text{Bart},\langle\text{ok,2)},(\text{ok,2}),\text{(ok,1.5)},(\text{ok,1.5})\rangle,1) \notag \\
  M &= \text{SHOP}(\lambda,BAG\langle ITEM(\lambda,\lambda)\rangle,Discount) \notag \\
  \pi^N_M(t) &= (\text{ok},\langle\text{(ok,ok)},(\text{ok,ok}),\text{(ok,ok)},(\text{ok,ok})\rangle,1)
  \end{align*}
Identifying Nested Data Elements

- to store tuples in relational database we store their values on attributes
- $a \in L$ of lattice $(L, \sqsubseteq, \sqcup, \sqcap, 0)$ is join-irreducible iff $a \neq 0$ and if $a = b \sqcup c$ holds for any $b, c \in L$, then $a = b$ or $a = c$
- let $\mathcal{B}(N)$ denote the join-irreducibles of $(\text{Sub}(N), \leq, \sqcup, \sqcap, \lambda_N)$
- What subattributes identify nested data elements in presence of type constructors?

- **Bag**\langle**ITEM**(Article, Price)\rangle:
  
  $\langle(\text{Donut}, 1.5), (\text{Donut}, 1.5), (\text{Chocolate}, 2), (\text{Chocolate}, 2)\rangle$
  $\langle(\text{Donut}, 2), (\text{Donut}, 2), (\text{Chocolate}, 1.5), (\text{Chocolate}, 1.5)\rangle$

- join-irreducibles are sufficient in presence of records and lists
- what is needed in presence of sets or multisets?
Extended Join-Irreducibles

- $X, Y \in Sub(N)$ reconcilable iff one of the following holds:
  - $Y \leq X$ or $X \leq Y$,
  - $N = L(N_1, \ldots, N_k), X = L(X_1, \ldots, X_k), Y = L(Y_1, \ldots, Y_k)$ where $X_i$ and $Y_i$ are reconcilable for all $i = 1, \ldots, k$,
  - $N = L[N'], X = L[X'], Y = L[Y']$ where $X'$ and $Y'$ reconcilable
- $\text{SHOP}(\lambda, \text{Bag}\langle \text{Item}(\text{Article}, \lambda) \rangle, \lambda)$, $\text{SHOP}(\lambda, \text{Bag}\langle \text{Item}(\lambda, \text{Price}) \rangle, \lambda)$
- extended join-irreducibles form smallest $\mathcal{E}(N) \subseteq Sub(N)$ such that
  (i) $\mathcal{B}(N) \subseteq \mathcal{E}(N)$, and
  (ii) for all $X, Y \in \mathcal{E}(N)$ which are not reconcilable also $X \sqcup Y \in \mathcal{E}(N)$
Functional Dependencies

- a functional dependency on nested attribute $N$ is
  $$\mathcal{X} \rightarrow \mathcal{Y} \quad \text{with } \leq \text{-antichains } \mathcal{X}, \mathcal{Y} \subseteq \mathcal{E}(N)$$

- $r \subseteq \text{Dom}(N)$ satisfies $\mathcal{X} \rightarrow \mathcal{Y}$ on $N$ ($\models_r \mathcal{X} \rightarrow \mathcal{Y}$) iff $\forall t_1, t_2 \in r$:
  $$\pi_X^N(t_1) = \pi_X^N(t_2) \quad \forall X \in \mathcal{X} \quad \text{implies} \quad \pi_Y^N(t_1) = \pi_Y^N(t_2) \quad \forall Y \in \mathcal{Y}$$

- $\text{SHOP}(\text{Bag}<\text{item}(\text{Article,Price})>) \rightarrow \text{SHOP}(\text{Discount})$

- $\text{SHOP}(\text{Bag}<\text{item}(\text{Article})>)$, $\text{SHOP}(\text{Bag}<\text{item}(\text{Price})>) \rightarrow \text{SHOP}(\text{Discount})$

- implication: $\Sigma \models \tau$ iff $\models_r \tau$ if $\models_r \sigma$ for all $\sigma \in \Sigma$ and any (finite) $r$

- $\Sigma \models_2 \tau$ iff $\models_r \tau$ if $\models_r \sigma$ for all $\sigma \in \Sigma$ and any 2-tuple instance $r$
Associating Horn Clauses and FDs

- one of Fagin’s ideas: interpret attributes as propositional variables
- interpret extended join-irreducibles as variables via $\psi : \mathcal{E}(N) \rightarrow \mathcal{V}$
- $\sigma = \{X_1, \ldots, X_n\} \rightarrow \{Y_1, \ldots, Y_m\}$ gives set $\Phi(\sigma)$ of Horn clauses

$$\bigwedge_{i=1}^{n} \psi(X_i) \Rightarrow \psi(Y_1), \ldots, \bigwedge_{i=1}^{n} \psi(X_i) \Rightarrow \psi(Y_m)$$

- Horn clauses can also encode the structure of $N$

$$\Pi_N = \{\psi(U) \Rightarrow \psi(V) \mid U, V \in \mathcal{E}(N), U \text{ covers } V\}$$
The Equivalence

**Theorem.** Let $N$ be a nested attribute, $\Sigma$ a set of FDs and $\sigma$ a single FD on $N$. Let $\Pi_N$ denote the Horn clauses which encode the structure of $N$, and $\Pi$ denote the corresponding set of Horn clauses for $\Sigma$. Then

(i) $\Sigma$ implies $\sigma$,

(ii) $\Sigma$ implies $\sigma$ in the world of two-tuple instances, and

(iii) $\Pi \cup \Pi_N$ logically implies $\pi$ for all $\pi \in \Phi(\sigma)$

are equivalent.

- this extends a well-known result by *Fagin (1977)*, where
  - only single application of record constructor allowed,
  - join-irreducibles form anti-chain, and
  - join-irreducibles (attributes) suffice
A simple Example

• bijection $\psi$:

\[
\begin{align*}
\text{SHOP}(\text{Customer}) & \leftrightarrow V_1, \\
\text{SHOP}(\text{Bag}(\text{Item}(\text{Article, Price}))) & \leftrightarrow V_2, \\
\text{SHOP}(\text{Bag}(\text{Item}(\text{Article}))) & \leftrightarrow V_3, \\
\text{SHOP}(\text{Bag}(\text{Item}(\text{Price}))) & \leftrightarrow V_4, \\
\text{SHOP}(\text{Bag}(\text{Item}(\lambda, \lambda))) & \leftrightarrow V_5, \\
\text{SHOP}(\text{Discount}) & \leftrightarrow V_6
\end{align*}
\]

• $\text{SHOP}(\text{Bag}(\text{Item}(\text{Article, Price} )))$ → $\text{SHOP}(\text{Discount})$ doesn’t imply $\text{SHOP}(\text{Bag}(\text{Item}(\text{Article} )))$. $\text{SHOP}(\text{Bag}(\text{Item}(\text{Price} )))$ → $\text{SHOP}(\text{Discount})$

  \begin{align*}
  \text{(Homer, \{(Donut, 1.5), (Donut, 1.5), (Chocolate, 2), (Chocolate, 2)\}, 0)} \\
  \text{(Bart, \{(Donut, 2), (Donut, 2), (Chocolate, 1.5), (Chocolate, 1.5)\}, 1)}
  \end{align*}

• $\{V_2 \Rightarrow V_6, V_2 \Rightarrow V_3, V_2 \Rightarrow V_4, V_3 \Rightarrow V_5, V_4 \Rightarrow V_5\}$ doesn’t imply $V_3 \land V_4 \Rightarrow V_6$

  $\theta(V_i) = true$ iff $i \in \{3, 4, 5\}$
Applications

- re-using relational database design tools:

\[ R_N = \mathcal{E}(N) \quad \text{and} \quad \Sigma' = \Sigma \cup \{ X \rightarrow Y \mid X \text{ covers } Y \text{ in } \mathcal{E}(N) \}\]

- upper bounds for implication problem

\[ \Sigma \models \mathcal{X} \rightarrow \mathcal{Y} \text{ decidable in time } O(n) \text{ where } n \text{ denotes the total number of extended join-irreducibles occurring in } \Sigma \]

- apply tools from logic, e.g. first-literal unit resolution

- introduce *Boolean dependencies* \( Bd(N) \) on nested attributes \( N \):
  - \( \mathcal{E}(N) \subseteq Bd(N) \),
  - \( X \in Bd(N) \text{ implies } \neg X \in Bd(N) \),
  - \( X, Y \in Bd(N) \text{ implies } (X \land Y), (X \lor Y), (X \Rightarrow Y) \in Bd(N) \)
Conclusion and Future Work

- framework of nested attributes allows to capture data models by including corresponding type constructors
- theory of Brouwerian algebras can be used to extend many achievements from relational databases
- allows to study direct impact of type constructor on design problem without considering peculiarities of specific data model
- study different classes of dependencies in different combinations of constructors
- increase expressiveness by studying embedded dependencies (allowing several Brouwerian algebras simultaneously)
- normal forms