Functional Dependency

A functional dependency on relation $R$ is the assertion that when two tuples agree on attributes $\{A_1, \ldots, A_n\}$, they must also agree on attribute $B$.

$\{A_1, \ldots, A_n\} \rightarrow B$, or $\{A_1, \ldots, A_n\}$ functionally determine $B$

Example

- Drinkers(name, addr, beersLiked, manf, favBeer).
- Reasonable FD’s to assert:
  1. name $\rightarrow$ addr
  2. name $\rightarrow$ favBeer
  3. beersLiked $\rightarrow$ manf

Example Data

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>beersLiked</th>
<th>manf</th>
<th>favBeer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spock</td>
<td>Wick</td>
<td>Entourage</td>
<td>End</td>
<td>Wick</td>
</tr>
<tr>
<td>Because name $\rightarrow$ addr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Because beersLiked $\rightarrow$ favBeer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FD’s With Multiple Attributes

- No need for FD’s with $> 1$ attribute on right.
  - But sometimes convenient to combine FD’s as a shorthand.
  - Example: name $\rightarrow$ addr and name $\rightarrow$ favBeer become name $\rightarrow$ addr favBeer
- $> 1$ attribute on left may be essential.
  - Example: bar beer $\rightarrow$ price

Keys of Relations

- $\{A_1, \ldots, A_n\}$ is a key of $R$ if
  - $\{A_1, \ldots, A_n\}$ functionally determines all attributes of $R$
  - No proper subset of $\{A_1, \ldots, A_n\}$ functionally determines all attributes of $R$
- Superkey
- Note:
  - A relation may have multiple keys
  - A key may have multiple attributes
  - Minimal AI = Minimum
  - ER keys $\neq$ relational keys
Example

- Consider relation Drinkers(name, addr, beersLiked, manf, favBeer).
- \{name, beersLiked\} is a superkey because together these attributes determine all the other attributes.
  - name -> addr favBeer
  - beersLiked -> manf

Example, Cont.

- \{name, beersLiked\} is a **key** because neither \{name\} nor \{beersLiked\} is a superkey.
  - name doesn’t -> manf; beersLiked doesn’t -> addr.
- In this example, there are no other keys, but lots of superkeys.
  - Any superset of \{name, beersLiked\}.

Discovering FDs and Keys

- **Obvious ones**
  - SSN, VIN, StudentID ...
- **Less obvious ones**
  - (hour,room) -> class
  - (playerID, year) -> teamID
- **Keys from ER**
  - Entity Set
  - Binary relationships
  - Multi-way relationships

Trivial Functional Dependency

- **FD**: \{A₁,…,Aₙ\} → \{B₁,…,Bₘ\}
- **FD is trivial** if all B’s are in \textbf{A}
- **FD is nontrivial** if at least one B is not in \textbf{A}
- **FD is completely nontrivial** if no B is in \textbf{A}

Armstrong’s Axioms

- **Reflexivity**
  - If \{B₁,…,Bₘ\} ⊆ \{A₁,…,Aₙ\}, then \textbf{A} → \textbf{B}
- **Transitivity**
  - If \{A₁,…,Aₙ\} → \{B₁,…,Bₘ\}, and \{B₁,…,Bₘ\} → \{C₁,…,Cₙ\}, then \textbf{A} → \textbf{C}
- **Augmentation**
  - If \{A₁,…,Aₙ\} → \{B₁,…,Bₘ\}, then \{A₁,…,Aₙ,C₁,…,Cₙ\} → \{B₁,…,Bₘ,C₁,…,Cₙ\}

Closure of Attributes

- **Given**
  - a set of attributes \textbf{A}
  - a set of functional dependencies \textbf{S}
- **Closure of \textbf{A} under \textbf{S}, \textbf{A}⁺**, is the set of all possible attributes that are functionally determined by \textbf{A} based on the functional dependencies inferable from \textbf{S}
Simple Closure Example

- R: \{A,B,C\}
- S: \{A\rightarrow B, B\rightarrow C\}
- \{A\}^+ ??
- \{B\}^+ ??
- \{C\}^+ ??

Computing A+

- Initialize \(A^+ = A\)
- Search in S for \(B\rightarrow C\) where
  - \(B \subseteq A^+\)
  - \(C \notin A^+\)
- Add C to \(A^+\)
- Repeat until nothing can be added to \(A^+\)

Computing \(A^+\) Example

- R: \{A,B,C,D,E,F\}
- S: \{AB\rightarrow C, BC\rightarrow AD, D\rightarrow E, CF \rightarrow B\}
- \{A,B\}^+ ??
- Is \{A,B\} a key ??

Correctness of the Closure Algorithm

- If \(B \subseteq A^+\), then \(A\rightarrow B\)
  - Proof by induction
- If \(A\rightarrow B\), then \(B \subseteq A^+\)
  - Proof by contradiction – if such B exists, that’s because \(A\rightarrow B\) cannot be inferred from S
    - If \(A\rightarrow B\) is inferable from S, then all relations that satisfy S also satisfy \(A\rightarrow B\)
    - A counter-example can be constructed where it satisfies S but not \(A\rightarrow B\)

Projection

- We often want to break one relation into two or more relations
  - E.g. breaks \((A,B,C,D)\) into \((A,B,C)\) and \((C,D)\)
- The resulting relations can be considered as projections of the original relation
- Given a set of FDs for the original relation, what can we say about the FDs of the projected relations?

Compute Functional Dependencies After Projection

- Let the new relation be \(R’\), compute the closures of all subset’s of \(R’\)’s attributes, and exclude the FD’s that involves the attributes that are projected out.
  - No need to compute the closures of the empty set and the full set
  - If \(A^+\) is already the set of all attributes, no need to compute the closures of A’s superset
Example

\[ R: (A, B, C, D), \text{ S: } (A \rightarrow B, B \rightarrow C, C \rightarrow D) \]
\[ R': (A, C, D) \]

\[ A \rightarrow C, A \rightarrow D, C \rightarrow D: \text{ basis} \]
\[ A \rightarrow C, C \rightarrow D: \text{ minimal basis} \]