Appendix C: Advanced Normalization Theory

- Reasoning with MVDs
- Higher normal forms
  - Join dependencies and PJNF
  - DKNF
Let $D$ denote a set of functional and multivalued dependencies. The closure $D^+$ of $D$ is the set of all functional and multivalued dependencies logically implied by $D$.

Sound and complete inference rules for functional and multivalued dependencies:

1. **Reflexivity rule.** If $\alpha$ is a set of attributes and $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ holds.

2. **Augmentation rule.** If $\alpha \rightarrow \beta$ holds and $\gamma$ is a set of attributes, then $\gamma \alpha \rightarrow \gamma \beta$ holds.

3. **Transitivity rule.** If $\alpha \rightarrow \beta$ holds and $\gamma \alpha \rightarrow \gamma \beta$ holds, then $\alpha \rightarrow \gamma$ holds.
4. **Complementation rule.** If $\alpha \rightarrow \rightarrow \beta$ holds, then $\alpha \rightarrow \rightarrow R - \beta - \alpha$ holds.

5. **Multivalued augmentation rule.** If $\alpha \rightarrow \rightarrow \beta$ holds and $\gamma \subseteq R$ and $\delta \subseteq \gamma$, then $\gamma \alpha \rightarrow \delta \beta$ holds.

6. **Multivalued transitivity rule.** If $\alpha \rightarrow \rightarrow \beta$ holds and $\beta \rightarrow \rightarrow \gamma$ holds, then $\alpha \rightarrow \rightarrow \gamma - \beta$ holds.

7. **Replication rule.** If $\alpha \rightarrow \beta$ holds, then $\alpha \rightarrow \rightarrow \beta$.

8. **Coalescence rule.** If $\alpha \rightarrow \rightarrow \beta$ holds and $\gamma \subseteq \beta$ and there is a $\delta$ such that $\delta \subseteq R$ and $\delta \cap \beta = \emptyset$ and $\delta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$ holds.
We can simplify the computation of the closure of $D$ by using the following rules (proved using rules 1-8).

- **Multivalued union rule.** If $\alpha \rightarrow\rightarrow \beta$ holds and $\alpha \rightarrow\rightarrow \gamma$ holds, then $\alpha \rightarrow\rightarrow \beta \gamma$ holds.

- **Intersection rule.** If $\alpha \rightarrow\rightarrow \beta$ holds and $\alpha \rightarrow\rightarrow \gamma$ holds, then $\alpha \rightarrow\rightarrow \beta \cap \gamma$ holds.

- **Difference rule.** If $\alpha \rightarrow\rightarrow \beta$ holds and $\alpha \rightarrow\rightarrow \gamma$ holds, then $\alpha \rightarrow\rightarrow \beta - \gamma$ holds and $\alpha \rightarrow\rightarrow \gamma - \beta$ holds.
Example

- \( R = (A, B, C, G, H, I) \)
- \( D = \{ A \rightarrow\rightarrow B, \ B \rightarrow\rightarrow HI, \ CG \rightarrow H \} \)

- Some members of \( D^+ \):
  - \( A \rightarrow\rightarrow CGHI. \)
    - Since \( A \rightarrow\rightarrow B \), the complementation rule (4) implies that \( A \rightarrow\rightarrow R – B – A. \)
    - Since \( R – B – A = CGHI \), so \( A \rightarrow\rightarrow CGHI. \)
  - \( A \rightarrow\rightarrow HI. \)
    - Since \( A \rightarrow\rightarrow B \) and \( B \rightarrow\rightarrow HI \), the multivalued transitivity rule (6) implies that \( B \rightarrow\rightarrow HI – B. \)
    - Since \( HI – B = HI \), \( A \rightarrow\rightarrow HI. \)
Example (Cont.)

- Some members of $D^+$ (cont.):

  - $B \rightarrow H$.
    Apply the coalescence rule (8); $B \rightarrow Hl$ holds.
    Since $H \subseteq Hl$ and $CG \rightarrow H$ and $CG \cap Hl = \emptyset$, the
    coalescence rule is satisfied with $\alpha$ being $B$, $\beta$ being $Hl$, $\delta$ being $CG$,
    and $\gamma$ being $H$. We conclude that $B \rightarrow H$.

  - $A \rightarrow CG$.
    $A \rightarrow CGHl$ and $A \rightarrow Hl$.
    By the difference rule, $A \rightarrow CGHl - Hl$.
    Since $CGHl - Hl = CG$, $A \rightarrow CG$. 
Normalization Using Join Dependencies

- Join dependencies constrain the set of legal relations over a schema \( R \) to those relations for which a given decomposition is a lossless-join decomposition.

- Let \( R \) be a relation schema and \( R_1, R_2, ..., R_n \) be a decomposition of \( R \). If \( R = R_1 \cup R_2 \cup ... \cup R_n \), we say that a relation \( r(R) \) satisfies the join dependency \(*((R_1, R_2, ..., R_n))\) if:

  \[
  r = \Pi_{R_1}(r) \Pi_{R_2}(r) \ldots \Pi_{R_n}(r)
  \]

  A join dependency is trivial if one of the \( R_i \) is \( R \) itself.

- A join dependency \(*((R_1, R_2))\) is equivalent to the multivalued dependency \( R_1 \cap R_2 \rightarrow R_2 \). Conversely, \( \alpha \rightarrow \beta \) is equivalent to \(*((\alpha \cup (R - \beta), \alpha \cup \beta))\)

- However, there are join dependencies that are not equivalent to any multivalued dependency.
A relation schema $R$ is in PJNF with respect to a set $D$ of functional, multivalued, and join dependencies if for all join dependencies in $D^+$ of the form

$$*(R_1, R_2, ..., R_n)$$

where each $R_i \subseteq R$

and $R = R_1 \cup R_2 \cup ... \cup R_n$

at least one of the following holds:

- $*(R_1, R_2, ..., R_n)$ is a trivial join dependency.
- Every $R_i$ is a superkey for $R$.

Since every multivalued dependency is also a join dependency, every PJNF schema is also in 4NF.
Example

- Consider `Loan-info-schema = (branch-name, customer-name, loan-number, amount)`.
- Each loan has one or more customers, is in one or more branches and has a loan amount; these relationships are independent, hence we have the join dependency
- `*(=(loan-number, branch-name), (loan-number, customer-name), (loan-number, amount))`
- `Loan-info-schema` is not in PJNF with respect to the set of dependencies containing the above join dependency. To put `Loan-info-schema` into PJNF, we must decompose it into the three schemas specified by the join dependency:
  - `(loan-number, branch-name)`
  - `(loan-number, customer-name)`
  - `(loan-number, amount)`
Domain-Key Normal Form (DKNY)

- **Domain declaration.** Let A be an attribute, and let \( \text{dom} \) be a set of values. The domain declaration \( A \subseteq \text{dom} \) requires that the A value of all tuples be values in \( \text{dom} \).

- **Key declaration.** Let \( R \) be a relation schema with \( K \subseteq R \). The key declaration \( \text{key} (K) \) requires that \( K \) be a superkey for schema \( R \) (\( K \rightarrow R \)). All key declarations are functional dependencies but not all functional dependencies are key declarations.

- **General constraint.** A general constraint is a predicate on the set of all relations on a given schema.

- Let \( D \) be a set of domain constraints and let \( K \) be a set of key constraints for a relation schema \( R \). Let \( G \) denote the general constraints for \( R \). Schema \( R \) is in DKNF if \( D \cup K \) logically imply \( G \).
Example

- Accounts whose account-number begins with the digit 9 are special high-interest accounts with a minimum balance of 2500.
- General constraint: ``If the first digit of \( t[\text{account-number}] \) is 9, then \( t[\text{balance}] \geq 2500.\)"
- DKNF design:
  
  - Regular-acct-schema = (branch-name, account-number, balance)
  - Special-acct-schema = (branch-name, account-number, balance)

- Domain constraints for \{Special-acct-schema\} require that for each account:
  - The account number begins with 9.
  - The balance is greater than 2500.
Let $R = (A_1, A_2, ..., A_n)$ be a relation schema. Let $\text{dom}(A_i)$ denote the domain of attribute $A_i$, and let all these domains be infinite. Then all domain constraints $D$ are of the form $A_i \subseteq \text{dom}(A_i)$.

Let the general constraints be a set $G$ of functional, multivalued, or join dependencies. If $F$ is the set of functional dependencies in $G$, let the set $K$ of key constraints be those nontrivial functional dependencies in $F^+$ of the form $\alpha \rightarrow R$.

Schema $R$ is in PJNF if and only if it is in DKNF with respect to $D$, $K$, and $G$. 