Algorithm for Finding a Canonical Cover of a Set of FDs

- Write F as a set of dependencies where each has a single attribute on the right hand side.
- Eliminate trivial dependencies.
- Eliminate redundant dependencies (implied by other dependencies).
- Combine dependencies with the same left hand side.

Example

Find a canonical cover for the dependencies in our library database:

(1) Start with the following closure of the various attributes we found earlier:

borrower_id → borrower_id, last_name, first_name
call_number → call_number, title, author
call_number, copy_number → call_number copy_number, title, accession_number,
borrower_id, date_due, author, last_name, first_name
accession_number → accession_number, call_number, copy_number, title, borrower_id,
date_due, author, last_name, first_name

(2) Rewrite with a single attribute on the right hand side of each

borrower_id → borrower_id
borrower_id → last_name
borrower_id → first_name
call_number → call_number
call_number → title
call_number → author
call_number, copy_number → call_number
call_number, copy_number → copy_number
call_number, copy_number → title
call_number, copy_number → accession_number
call_number, copy_number → borrower_id
call_number, copy_number → date_due
call_number, copy_number → author
call_number, copy_number → last_name
call_number, copy_number → first_name
accession_number → accession_number
accession_number →call_number
accession_number →copy_number
accession_number →title
accession_number →borrower_id
accession_number →date_due
accession_number →author
accession_number →last_name
accession_number →first_name

(3) Now eliminate the trivial dependencies

(Cross out “reflexive” FDs on list)

(4) There are dependencies in this list that are implied by other dependencies in the list, and so should be eliminated.

call_number, copy_number → title
call_number, copy_number → author
(Since the same RHS appears with only call_number on the LHS)

accession_number → title
accession_number → author
(These are implied by the transitive rule given that accession_number → call_number and call_number determines these).

call_number, copy_number → last_name
call_number, copy_number → first_name
(These are implied by the transitive rule given that call_number, copy_number → borrower_id and borrower_id determines these)

accession_number → last_name
accession_number → first_name
(These are implied by the transitive rule given that accession_number → borrower_id and borrower_id determines these)

Either one of the following, but not both:

call_number, copy_number → borrower_id
call_number, copy_number → date_due

or

accession_number → borrower_id
accession_number → date_due

(Either set is implied by the transitive rule from the other set given accession_number → call_number, copy_number or call_number, copy_number → accession_number.)

Assume we keep the ones with call_number, copy_number on the LHS.

(5) Result after eliminating redundant dependencies:

borrower_id → last_name
borrower_id → first_name
call_number → title
call_number → author
call_number, copy_number → accession_number
call_number, copy_number → borrower_id
call_number, copy_number → date_due
accession_number → call_number
accession_number → copy_number

(6) Rewrite in canonical form by combining dependencies with the same left-hand side:

borrower_id → last_name, first_name
call_number → title, author
call_number, copy_number → accession_number, borrower_id, date_due
accession_number → call_number, copy_number

Note: for any given set of FD's, the canonical cover is not necessarily unique - there may be more than one set of FD's that satisfies the requirement.

Example: For the above, we could have kept

accession_number → borrower_id, date_due

and dropped

call_number, copy_number → borrower_id, date_due.