Finding Schema Normal Forms

Tom Kelliher, CS 318

Apr. 3, 2002

1 Administrivia

Announcements

Assignment

Read 9.1–3.

From Last Time

Normal forms; properties of decompositions.

Outline

1. Preliminaries: minimal covers.
2. 3NF synthesis.
3. Finding BCNF schemas.

Coming Up

Triggers.

2 Preliminaries

2.1 Minimal Cover

1. Sets of FDs aren't unique. It'd be nice to have a canonical set.

   Minimal covers come close.

2. Let $\mathcal{F}$ be a set of FDs. The set of FDs $\mathcal{G}$ is a minimal cover of $\mathcal{F}$ if:

   (a) $\mathcal{G}$ is equivalent to $\mathcal{F}$.

   (b) All FDs in $\mathcal{G}$ have the form $\overline{X} \to A$ where A is a singleton attribute.

   (c) It is not possible to make $\mathcal{G}$ smaller by either:

      i. Deleting an FD from $\mathcal{G}$.

      ii. Deleting an attribute from the LHS of one of $\mathcal{G}$'s FDs.

3. Algorithm for finding a minimal cover of $\mathcal{F}$:

   (a) Compute $\mathcal{G} := \mathcal{F}$ where the FDs in $\mathcal{G}$ have singleton RHSs.

   Note: Just apply FD decomposition.

   (b) Remove redundant attributes from the RHSs of FDs of $\mathcal{G}$.

   Note on computation: To check to see if $B \in \overline{X}$ in $\overline{X} \to A$ of $\mathcal{G}$ is redundant, check whether or not $(\overline{X} - B) \to A$ is entailed by $\mathcal{G}$.

   (c) Remove redundant FDs from $\mathcal{G}$.

   Note on computation: To check to see if an FD of $\mathcal{G}$ is redundant, create $\mathcal{G}'$ which is just $\mathcal{G}$ without the candidate redundant FD. Check whether or not the candidate FD is entailed by $\mathcal{G}'$.

   The resulting $\mathcal{G}$ is a minimal cover of $\mathcal{F}$.

4. Example. Find a minimal cover of:

$$
\begin{aligned}
ABH &\rightarrow C \\
A &\rightarrow D \\
C &\rightarrow E \\
BGH &\rightarrow F \\
F &\rightarrow AD \\
E &\rightarrow F \\
BH &\rightarrow E
\end{aligned}
$$

3 Synthesizing 3NF Schemas

1. Algorithm works by collecting individual attributes into groups, forming relations, as opposed to decomposing relations.

2. Algorithm. Given $\mathbf{R} = (\overline{R}; \mathcal{F})$

   (a) Find a minimal cover, $\mathcal{G}$, of $\mathcal{F}$.

   (b) Partition $\mathcal{G}$ into sets $\mathcal{G}_i$ where the FDs in a $\mathcal{G}_i$ all have the same LHS. Minimize $i = 1, \ldots, n$.

   (c) For each $\mathcal{G}_i$, the corresponding relation schema is $\mathbf{R_i} = (\overline{R_i}; \mathcal{G}_i)$ where $\overline{R_i}$ contains all the attributes mentioned in $\mathcal{G}_i$.

   (d) If any of the $\overline{R_i}$ is a superkey of R, we are done. (Check $(\overline{R_i})^+_{\mathcal{F}} = \overline{R}$.)

   Otherwise, let $\overline{R_0}$ be a key of $\mathbf{R}$. Add $\mathbf{R_0} = (\overline{R_0}; \mathcal{G}_0)$ to the schema set.

   Why is the resulting schema

   (a) in 3NF?

   (b) dependency preserving?

   (c) lossless?

3. It would appear the LHSs of FDs of any of the relations form superkeys of their respective relations. Why aren't the schemas necessarily in BCNF?

   Consider the following schema $\mathbf{R} = (\overline{R}; \mathcal{F})$: $\overline{R} = ABCDE$ and $\mathcal{F}$ contains

$$\begin{aligned} ABC &\rightarrow D \\ ABC &\rightarrow E \\ B &\rightarrow D \end{aligned}$$

   This will synthesize into two relations. The third FD actually will apply to both relations.

4 Finding BCNF Schemas

1. A good overall strategy: synthesize a 3NF schema. If it is also in BCNF, we are done. Otherwise, perform BCNF decomposition on the 3NF.

   Recall, the 3NF will be dependency preserving. A dependency preserving BCNF schema is our first choice.

2. Algorithm. Given $\mathbf{R} = (\overline{R}; \mathcal{F})$.

   Decomposition := $\mathbf{R}$
   while there is a schema $\mathbf{S} = (\overline{S}; \mathcal{F}')$ in Decomposition that is not in BCNF **do**
   /* Assume $\overline{X} \rightarrow \overline{Y}$ in $\mathcal{F}'$ violates BCNF. */
   Replace $\mathbf{S}$ in Decomposition with schemas $\mathbf{S_1} = (\overline{XY}; \mathcal{F}_1)$ and
   $\mathbf{S_2} = ((\overline{S} - \overline{Y}) \cup \overline{X}; \mathcal{F}_2)$ where $\mathcal{F}_1$ and $\mathcal{F}_2$ are all the FDs of $\mathcal{F}'$ that involve only attributes from their respective schemas
   end

   Questions:

   (a) After an iteration, why will $\overline{X} \rightarrow \overline{Y}$ no longer violate BCNF?

   (b) Why might this not be dependency preserving?

   (c) Why is this lossless?

3. Example. Find a BCNF decomposition of $\mathbf{R} = (\overline{R}; \mathcal{F}$ where $\overline{R} = ABCDEFGH$ and $\mathcal{F}$ contains:

$$ABH \rightarrow C$$

$$
\begin{aligned}
A &\rightarrow DE \\
BGH &\rightarrow F \\
F &\rightarrow ADH \\
BH &\rightarrow GE
\end{aligned}
$$