1. Enunciado

Sea R un esquema relacional. Sea $p = (R_1, R_2)$ una descomposición, y sea F un conjunto de dependencias funcionales. Es cierto que si

$$R_1 \cap R_2 \to R_2 - R_1$$

entonces no hay pérdida de dependencias funcionales?

Definiciones

La definición de proyección de un conjunto de dep. func. F sobre una partición $R = A_1 ... A_n$es

$$\pi_F(R) = \left\{ X \to Y \mid (X \to Y \in F^+) \wedge (XY \subseteq R) \right\}$$

La definición de que una descomposición tiene pérdida de dep. func. es:

$$\pi_{R1}(F) \cup \pi_{R2}(F) \models F$$

i.e. F incl. en la clausura de la derecha

Resolución

No, la afirmación es falsa. Veamos un contraejemplo.

Sea R el esquema relacional CEP, donde C es Ciudad, E es Estado y P es CódigoPostal.

Sea $F = \{CE \to P, P \to C\}$

Sea $p = (R_1, R_2)$, con $R_1 = EP$ y $R_2 = CP$

Veamos que vale $R_1 \cap R_2 \to R_2 - R_1$

$$R_1 \cap R_2 = EP \cap CP = P$$

$$R_2 - R_1 = C$$

Claramente, $F \models P \to C$ (en particular, $P \to C \in F$)

Resta ver que la descomposicón pierde dependecias funcionales.

La proyección de F sobre EP son solo dependencias triviales (es decir, que se derivan del axioma de reflexibidad).

La proyección de F sobre CP es $P \to C$ mas dependencias funcionales triviales.

Como $\pi_{R1}(F) \cup \pi_{R2}(F) \not\models F$, pues $CE \to P$ *no* es una implicación lógica de $\pi_{R1}(F) \cup \pi_{R2}(F)$, con lo cual la descomposición no preserva dep. func.

2. Enunciado

Sea R un esquema relacional. Sea $p = (R_1, R_2)$ una descomposición y sea F un conjunto de dependencias funcionales. Demuestre que

$$R_1 \cap R_2 \to R_1 - R_2$$

si y solo si la descomposicón es sin pérdida por junta.

Resolución

=>) sppj $=> R_1 \cap R_2 \to R_1 - R_2$

Usando el algoritmo para testear si una descomp. es sppj, tenemos la siguiente tabla inicial

......

Al finalizar el algoritmo, nos queda una fila $a_! \dots a_n$

......

Si la fila con a's es la primera, intercambiamos los nombres de R_1 y R_2 y listo.

Sea A un atributo cualquiera tq $A \in R_1 - R_2$

Para la fila R_2, columna A, el valor inicial es b, y luego de j iteraciones del algoritmo, se convierte en a.

Luego, existe una dep. func. X\rightarrow Y tq:

- $X \subseteq R_1 \cap R_2$ y
- $A \subseteq Y$

Entonces $A \in X^+ \subseteq (R_1 \cap R_2)^+$

Como A es un atributo de R_1-R_2 vale que $R_1 \cap R_2 \to R_1 - R_2$

<=) $R_1 \cap R_2 \to R_1 - R_2 =>$sppj

Como $R_1 \cap R_2$ vale a en las dos filas, y $R_1 \cap R_2 \to R_1 - R_2$, por el lema el algoritmo pone a en $R_1 - R_2$, por lo que queda una fila $a_1 \dots a_n$ (ver tabla inicial), y la descomposición es sppj.

Lema: si

1. Si $X = a$ en las dos filas
2. $\vdash X \to Y$

entonces el algoritmo pone $Y = a$ en las dos filas en alguna iteración.

Demostración: por inducción en la longitud de la demostración de $X \to Y$ por los axiomas de Armstrong.

Caso base: longitud = 1

- Si $X \to Y \in F$, en una iteración el algoritmo iguala las dos filas con a.
- $Y \subsetneq X$(reflexividad), en cuyo caso el algoritmo inicializa las dos filas con a.

Paso inductivo:

Hipotesis: Para toda demostración de $X \to Y$ con longitud $< n$, si $X = a$ en las 2 filas, entonces el algoritmo eventualmente pone Y en a en las 2 filas.

Tésis: q.v.q. si tengo una demostración de longitud n y ambas filas tienen a en X, entonces el algorimto pone Y en a en las 2 filas.

- Transitividad

  ...
  i) $X \to Z$
  ...
  j) $Z \to Y$
  ...
  n) $X \to Y$ (por transitividad de i y j)

  X vale a en las dos filas, por HI, el algoritmo pone Z en a en las dos filas.

  Como Z vale a en las dos filas, por HI, el algoritmo pone Y en a en las dos filas.

- Aumentación

  i) $V \to W$
  ...
  n) $VZ \to WZ$ (uso $X = VZ$ y $Y = WZ$)

  $V \to W$ y las dos filas valen a en V, por HI el algoritmo pone W en a.

  Si las dos filas valen a en X, como $Z \subsetneq X$ ya tiene sus atributos en a

  Luego, $WZ = Y$ tiene a en las dos filas

□