NORMALIZACIÓN

Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación
Base de Datos

Autor: Alejandro Eidelsztein

DEFINICIÓN DE DEPENDENCIA FUNCIONAL

Decimos que *vale* $X \dashrightarrow Y$ en R

si para toda r se verifica que

si $t_1(X) = t_2(X)$ entonces necesariamente $t_1(Y) = t_2(Y)$

Decimos que

X *determina funcionalmente* Y

o que

Y *es determinado funcionalmente* por X

R, esquema de relación
r, instancia de R
X, conjunto de atributos de R, lado izquierdo.
Y, conjunto de atributos de R, lado derecho
X e Y no tienen que ser necesariamente disjuntos
t_1 y t_2 dos tuplas cualesquiera de r

REGLAS DE INFERENCIA

1) **Reflexividad: Si Y $\subseteq$ X entonces X --> Y**

2) **Aumento: Para cualquier W, si X --> Y entonces XW --> WY**

3) **Transitividad: Si X --> Y e Y --> Z entonces X -->Z**

4) Unión: Si X --> Y y X --> Z entonces X --> YZ

5) Pseudotransitividad: Para cualquier W, Si X --> Y e YW --> Z entonces XW --> Z

6) Descomposición: Si X --> YZ entonces X --> Y y X --> Z

NOTA:
Las reglas 1,2 y 3 son las llamadas *axiomas de Armstrong*.
Las reglas 4, 5 y 6 son las llamadas *adicionales* y se demuestran a partir de 1,2 y 3.

CLAUSURA DE UN CONJUTO DE DEPENDENCIAS FUNCIONALES

Es el conjunto de todas las dependencias funcionales que pueden inferirse a partir de F aplicando las reglas de inferencia, o sea:

$$F^+ = \{X \dashrightarrow Y \,/\, F \models X \dashrightarrow Y\}$$

CLAUSURA DE UN CONJUNTO DE ATRIBUTOS

Dado el conjunto de atributos X, X^+ con respecto a F, es el conjunto de todos los atributos A tal que X --> A, o sea:

$$X^+ = \{A \in R \ / \ F \models X \rightarrow A\}$$

Algoritmo:
Una forma de calcular X^+ es computar una secuencia de conjuntos de atributos X^0, X^1, … aplicando las siguientes reglas:

1) X^0 es X
2) X^{i+1} es X^i Unión {el conjunto de atributos A tal que hay alguna dependencia funcional Y --> Z en F, A está en Z e $Y \subseteq X^i$ }
   Aplicamos repetidas veces la regla (2) hasta que $X^i = X^{i+1}$

NOTA: Como $X^0 \subseteq \ldots \subseteq X^i \subseteq R$, y R es finito, eventualmente llegaremos a que $X^i = X^{i+1}$, que es la parada del algoritmo.

PROPIEDAD DE LA CLASURA DE UN CONJUNTO DE ATRIBUTOS

Dados F y X --> Y, luego F $\models$ X --> Y Sii Y $\subseteq$ X^+

Si X^+ = R entonces X es **SUPERCLAVE** de R

EQUIVALENCIA DE CONJUNTOS DE DEPENDENCIAS FUNCIONALES

Decimos que dos conjuntos de dependencias funcionales F y G sobre R son equivalentes ($F \equiv G$) si

$$F^+ = G^+$$

O en forma equivalente si:

$$F \models G \quad y \quad G \models F$$

(si F cubre a G y G cubre a F)

CUBRIMIENTO MINIMAL

Dado F buscamos un F_m tal que $F_m \equiv F$ y además F_m tiene:

1) Todo lado derecho tiene un único atributo (regla de descomposición)

2) Todo lado izquierdo es reducido (no tiene atributos redundantes),
   $B \subset X$ es redundante para X-->A si $A \in (X - \{B\})^+$

3) No contiene dependencias funcionales redundantes (en general las que se obtienen por transitividad, X--> A es redundante si $(F - \{X\text{-->}A\}) \equiv F$)

NOTA:
Puede haber varios cubrimientos minimales para un mismo F

ATRIBUTOS PRIMOS

Decimos que un atributo es PRIMO si es miembro de ALGUNA clave candidata

Si no, decimos que es NO PRIMO

FORMA NORMAL DE BOYCE-CODD (FNBC):

R está en FNBC si para TODA dependencia funcional NO TRIVIAL X--> A sobre R,

X es una superclave de R

O en forma equivalente:

R está en FNBC si para TODA dependencia funcional X-->Y en F^+, o bien $Y \subseteq X$ o X es una superclave de R.

NOTA:

Todo esquema de dos atributos está en FNBC.

TERCERA FORMA NORMAL (3FN):

R esta en 3FN si para TODA dependencia funcional NO TRIVIAL X--> A sobre R,

1) X es superclave de R

o

2) A es primo

O en forma equivalente:

R está en 3FN si para TODA dependencia funcional NO TRIVIAL X-->Y sobre R, o bien X es una superclave de R o Y es un subconjunto de alguna clave de R.

SEGUNDA FORMA NORMAL (2FN):

R esta en 2FN si TODO atributo no primo A en R NO es *parcialmente dependiente* de ALGUNA clave de R.

O en forma equivalente:

R está en 2FN si TODO atributo NO primo A en R es *totalmente dependiente* de TODAS las claves de R.

NOTA:
Decimos que X--> Y es dependencia funcional *parcial* (o que Y *depende parcialmente* de X) si para algún subconjunto $Z \subset X$, se verifica que Z --> Y

Si no, decimos que la dependencia funcional es *total* (o que Y *depende totalmente* de X)

PRIMERA FORMA NORMAL (1FN):

R está en 1FN si todos sus atributos son atómicos.

En el modelo relacional todo esquema R está en 1FN por definición.

NOTA:

Si R cumple con FNBC, entonces también cumple con 3FN.

Si R cumple con 3FN, entonces también cumple con 2FN.

Si R cumple con 2FN, entonces también cumple con 1FN.

DESCOMPOSICIÓN DE UN ESQUEMA DE RELACIÓN:

Para solucionar los problemas de redundancia de información y las anomalías de actualización descomponemos el esquema inicial en subesquemas.

Se debe tratar que la *descomposición* cumpla con las siguientes propiedades:

SIN PÉRDIDA DE INFORMACIÓN (SPI): También llamada *Lossless Join*, significa que al *juntar* la información contenida en los subesquemas se obtiene la información original.

Siempre es posible cumplirla.

SIN PÉRDIDA DE DEPENDENCIAS FUNCIONALES (SPDF): Significa que todas las dependencias originales se pueden obtener a partir de las dependencias que se *proyectaron* sobre los subesquemas.

No siempre es posible cumplirla para la FNBC.

ALGORITMO PARA OBTENER UNA DESCOMPOSICIÓN EN 3FN SPI Y SPDF

Este algoritmo obtiene una descomposición ρ de R por SÍNTESIS a partir de una COBERTURA MINIMAL de F.
El ρ resultante es SPI y SPDF.

1) Cada dependencia funcional se convierte en un esquema

2) Unificar los que provienen de dependencias que tienen igual lado izquierdo

3) Si ninguno de los esquemas resultantes contiene una clave se agrega uno con los atributos de alguna clave

4) Eliminar esquemas redundantes: Si alguno de los esquemas resultantes esta contenido totalmente en otro, eliminarlo.

PROPIEDAD DE DESCOMPOSICIÓN BINARIA:

Decimos que una descomposición ρ de R, $\rho = (R_1,R_2)$ es SIN PÉRDIDA DE INFORMACIÓN (SPI) con respecto a un conjunto de dependencias funcionales F, Sii

1) La dependencia funcional $(R_1 \cap R_2) \dashrightarrow (R_1 - R_2)$ está en F^+

o

2) La dependencia funcional $(R_1 \cap R_2) \dashrightarrow (R_2 - R_1)$ está en F^+

O sea que si la intersección de los atributos de R_1 y R_2 forman una *superclave* para uno de los dos esquemas (*), entonces ρ es SIN PÉRDIDA DE INFORMACIÓN (SPI)

(*) Observar que también se cumple: $(R_1 \cap R_2) \dashrightarrow R_1$ o $(R_1 \cap R_2) \dashrightarrow R_2$

PROYECCIÓN DE UN CONJUNTO DE DEPENDENCIAS FUNCIONALES

Dados R, y $\rho = (R_1, R_2, \ldots, R_k)$ decimos que la proyección de F sobre un conjunto de atributos Z (Π_Z (F)) es el conjunto de dependencias funcionales

$$X \dashrightarrow Y \in F^+ \text{ tal que } XY \subseteq Z$$

Observar que esto involucra no solamente a las dependencias de F sino a todas las de F^+

ρ preserva F, si la unión de todas las dependencias funcionales en Π_{Ri} (F) para i= 1, 2, …., k implican F , o sea:

$$F^+ = (U_{i=1,\,k}\ \Pi_{Ri}(F))^+$$

NOTA:
ρ podría ser SPI y SPDF (o ninguna de las dos) o SPI pero no SPDF (o viceversa)

ALGORITMO PARA OBTENER UNA DESCOMPOSICIÓN EN FNBC SPI:

Este algoritmo obtiene una descomposición ρ de R, partiendo R aplicando la propiedad de DESCOMPOSICIÓN BINARIA. El ρ resultante es siempre SPI, pero no siempre se puede lograr que también sea SPDF.

Si hay una dependencia funcional X --> Y que viola FNBC se parte R en R1 y R2 de la siguiente forma:

R

X --> Y

R1= XY   R2= R-Y

Y así sucesivamente, hasta que todos los subesquemas finales queden en FNBC.

NOTA: Vemos que esta descomposición es SPI porque aplica la regla de descomposición binaria.

ALGORITMO PARA OBTENER UNA DESCOMPOSICIÓN EN FNBC SPI (Continuación):

En cada paso para cada subesquema -de más de dos atributos- obtenido debemos:

1) Proyectar F (involucra verificar para toda dependencia en F^+)
2) Calcular las claves
3) Verificar en qué FN se encuentra

NOTA:

Se pueden obtener varias descomposiciones diferentes dependiendo del orden en que se vayan tomando las dependencias