Tema 5

El cálculo relacional

IDBIS – DECSAI

Definición formal: ideas básicas

- Un lenguaje de Cálculo de Predicados se define para describir un “mundo”. Debe tener símbolos y frases.
- Este lenguaje debe incluir un alfabeto donde haya:
  - Símbolos para describir los **objetos del mundo** (constantes)
    - Juan, José, ...Seat,..... Rojo,....etc, GR-150-A....
  - Símbolos para describir **funciones** que nos dan unos objetos en función de otros:
    - Color, Padre, Madre, Propietario etc...
  - Símbolos para describir **variables**: x, y, z,
  - Símbolos de predicados que describan **relaciones** entre objetos:
    - Casados, Conduce, Prefiere.
  - Los predicados describen relaciones binarias, ternarias etc...

- El lenguaje además de símbolos tendrá que generar "frases" (fórmulas, expresiones...) por ello debe tener
  - Símbolos de puntuación: ( ) , . ; [ ]
  - Conectores: $\wedge$, $\vee$, $\neg$, $\rightarrow$
  - Cuantificadores: $\forall$, $\exists$

El Cálculo de predicados como elemento de representación de información

- Definición formal: Un lenguaje de Cálculo de Predicados se define como L=(S, W) donde
  - S es un conjunto de símbolos incluyendo:
    - Constantes
    - Funciones
    - Variables
    - Predicados
    - Símbolos adicionales
  - W un conjunto de frases "correctas" o fórmulas bien formadas (Well formed formulae) o "wff"
  - W se define de forma recursiva:
    - Un átomo a se define cómo:
      - Un símbolo de constante (Jose) ó
      - Un símbolo de variable (x) ó
      - f(a) donde f es un símbolo de función y a un átomo: Padre(José), Color(x)

El Cálculo de predicados como elemento de representación de información

- Una formula atómica se define como $p(a_1,a_2,..a_n)$ donde:
  - p es un símbolo de predicado n-ario
  - $a_1,a_2,....a_n$ son átomos

- Toda formula atómica es wff ($\in$ W)

- Si $f_1,f_2 \in W$ entonces:
  - $f_1 \vee f_2 \in W$; $f_1 \wedge f_2 \in W$; $\neg f_1 \in W$; $f_1 \rightarrow f_2 \in W$

- Si $f_1(x) \in W$ entonces:
  - $\forall$ x $f_1(x) \in W$; $\exists x\, f_1(x) \in W$

- Algunos ejemplos de wffs:
  - casados(Juan,Ana), casados(padre(x),madre(x)),prefiere(Ana,Honda,rojo)
  - conduce(Juan,GR-150-A)$\wedge\neg$conduce(propietario(GR-150-A),GR-150-A)
  - $\forall$x coche(x)$\rightarrow$prefiere(propietario(x),marca(x),color(x))
  - $\exists$y (persona(y) $\wedge\neg$casados(padre(y),madre(y)))

- Toda variable en una wff que no esté cuantificada se denomina variable libre
- En caso contrario se denomina variable ligada

El Cálculo de predicados como elemento de representación de información

- Ejemplos:
  - **casados(Juan,Ana)** será cierta si (Juan,Ana)$\in$E(casados)

  donde E(casados) es la extensión del predicado casados (equivale a la instancia de la relación casados)
  - **prefiere(Ana,Honda,rojo)** es cierta si (Ana,Honda,rojo)$\in$E(prefiere)
  - **conduce(Juan,GR-150-A)$\wedge\neg$conduce(propietario(GR-150-A),GR-150-A)** sera cierta si conduce(Juan,GR-150-A) es cierta y conduce(propietario(GR-150-A),GR-150-A) es falsa
  - **$\exists$y (persona(y) $\wedge\neg$casados(padre(y),madre(y)))** será cierta si podemos encontrar una constante c tal que
    - (persona(c) $\wedge\neg$casados(padre(c),madre(c)))
  - **x | casados(padre(x),madre(x))** define un conjunto de constantes

Analogías intuitivas entre el calculo de predicados y el modelo relacional

- Ideas básicas:
  - Un lenguaje de calculo de predicados y un modelo relacional son estructuras formales para describir la realidad: ambas pueden identificarse.
  - Una instancia de una base de datos se identificaría entonces con una interpretación de su lenguaje asociado.
  - Las reglas de integridad serían wffs y la interpretación debería ser un modelo para ellas.
  - Las consultas se generarían mediante wffs con variables libres. Los conjuntos de constantes que las hacen ciertas serán la solución de la consulta.

El calculo relacional orientado a tuplas

- Definición de una consulta:
  - Consideremos una base de datos con relaciones $R(A_1,...A_n)$, $S(B_1,...B_m)$ etc... y le asociamos un lenguaje de Cálculo de Predicados.
  - Supongamos que R_x, R_y ....S_x, S_y.. etc.. son variables que toman valores en R, S etc.. Son variables tupla.
  - Una consulta en C.R. orientado a tuplas (lenguaje QUEL) tiene la forma:

Select $R_x.A_i$, $R_x.A_j$ ..., $R_y.A_h$, $S_x.B_l$,... Where $wff(R_x,R_y,S_x...)$

- $R_x.A_i$, $R_x.A_j$ ..., $R_y.A_h$,$S_x.B_l$,... se denomina "lista objetivo" .
- $wff(R_x,R_y,S_x...)$ es una fórmula cuyas variables libres aparecen en la lista objetivo.
- La particularización de la lista objetivo para las tuplas que hacen cierta esta fórmula nos da la solución a la consulta.

Tema 5 Cálculo relacional y las BD deductivas

El calculo relacional orientado a tuplas

- Ejemplo:
  - Modelo relacional
    - S(S#,nombres,ciudad,status)
    - P(P#,tipop,peso,color,ciudad)
    - J(J#,nombre,ciudad,director,presupuesto)
    - SPJ(S#,P#,J#,cantidad,fecha)
  - Lenguaje
    - Constantes: s_1,s_2,..., p_1,p_2..., Madrid,...., 24, 25 ,..., etc.....
    - Variables:
      - Range S_x, S_y ... in S, Range P_x, P_y, ...in P
    - Funciones: S#, nombres,....P#,......
  - Consultas
    - Select S_x.S#,S_x.nombres,S_x.ciudad where S_x.status=25
    - Select S_x.S#,S_x.nombres where $\exists SPJ_y(SPJ_y.s\#=S_x.S\# \wedge SPJ_y.p\#=`p_1' \wedge SPJ_y.cantidad>=200)$

Ejemplos

Trabajadores (id_trabajador, nombre, trf_hr, tipo_de_oficio, id_supv)
Edificios (id_edificio, dir_edificio, tipo, nivel_calidad, categoria)
Asignaciones (id_trabajador, id_edificio, fecha_inicio, num_dias)
Oficios (tipo_de_oficio, prima, horas_por_sem)

Tema 5 Cálculo relacional y las BD deductivas

Ejemplos

- Encontrar los datos de aquellos trabajadores que son electricistas:

```
RANGE Tx IN Trabajadores
SELECT Tx.*
WHERE Tx.tipo_de_oficio='Electricista'
```

$\sigma_{\text{tipo\_de\_oficio='Electricista'}}$ TRABAJADORES

- Encontrar el nombre de aquellos trabajadores que son electricistas:

```
RANGE Tx IN Trabajadores
SELECT Tx.nombre
WHERE Tx.tipo_de_oficio='Electricista'
```

$\Pi_{\text{nombre}}(\sigma_{\text{tipo\_de\_oficio='Electricista'}}$ TRABAJADORES)

- Encontrar el número de horas semanales que trabaja cada trabajador:

RANGE Tx IN Trabajadores
RANGE Ox IN Oficios
SELECT Tx.nombre, Ox.horas_por_sem
WHERE Tx.tipo_de_oficio=Ox.tipo_de_oficio

$\Pi_{nombre,horas\_por\_sem}$ (TRABAJADORES|X|OFICIOS)

Ejemplos

- Encontrar los nombres de trabajadores que han trabajado tanto en el edificio 312 como en el edificio 460:

RANGE Tx IN Trabajadores
RANGE Ax, Ay IN Asignaciones
SELECT Tx.nombre
WHERE (∃Ax,Ay ((Ax.id_trabajador=Tx.id_trabajador) ∧ (Ay.id_trabajador=Tx.id_trabajador) ∧ (Ax.id_edificio=312) ∧ (Ay.id_edificio=460))

$$\Pi_{nombre}\ (\text{TRABAJADORES}\,|X|\,((\Pi_{id\_trabajador}\ \sigma_{id\_edificio=312}\ \text{ASIGNACIONES}) \cap (\Pi_{id\_trabajador}\ \sigma_{id\_edificio=460}\ \text{ASIGNACIONES}))$$

Ejemplos

- Encontrar los nombres de trabajadores que han trabajado o en el edificio 312 o en el edificio 460:

RANGE Tx IN Trabajadores

RANGE Ax IN Asignaciones

SELECT Tx.nombre

WHERE $\exists$Ax ((Ax.id_trabajador=Tx.id_trabajador) $\wedge$ ((Ax.id_edificio=312) $\vee$ (Ax.id_edificio=460)))

$\Pi_{nombre}(\text{TRABAJADORES}|X|\sigma_{id\_edificio=312 \vee id\_edificio=460}\ \text{ASIGNACIONES})$

Ejemplos

- Encontrar el nombre de aquellos trabajadores que no han trabajado en el edificio 312:

RANGE Tx IN Trabajadores

RANGE Ax IN Asignaciones

SELECT Tx.nombre

WHERE ¬(∃Ax ((Ax.id_trabajador=Tx.id_trabajador) ∧ (Ax.id_edificio=312)))

Π_{nombre} (TRABAJADORES |X| ($\Pi_{id_trabajador}$TRABAJADORES - $\Pi_{id\_trabajador}$($\sigma_{id\_edificio=312}$ ASIGNACIONES)))

Ejemplos

- Encontrar parejas de trabajadores que tengan el mismo oficio:

RANGE Tx, Ty IN Trabajadores
SELECT Tx.nombre, Ty.nombre
WHERE (Tx.tipo_de_oficio=Ty.tipo_de_oficio) $\wedge$ Tx.nombre<Ty.nombre)}

A:= $\Pi_{\text{nombre, tipo\_de\_oficio}}$ TRABAJADORES
B:= $\Pi_{\text{nombre, tipo\_de\_oficio}}$ TRABAJADORES
$\Pi_{\text{A.nombre, B.nombre}}$ ($\sigma_{\text{A.tipo\_de\_oficio=B.tipo\_de\_oficio} \wedge \text{A.nombre<B.nombre}}$ (AXB))

- Encontrar aquellos edificios en los que han trabajado todos los trabajadores de la empresa:

RANGE Tx IN Trabajadores
RANGE Ex IN Edificios
RANGE Ax IN Asignaciones
SELECT Ex.*
WHERE $\forall$Tx ($\exists$Ax ((Ax.id_trabajador=Tx.id_trabajador) $\wedge$ (Ax.id_edificio=Ex.id_edificio)))

$$(\Pi_{id\_edificio,\ id\_trabajador}\ \text{ASIGNACIONES})$$
$$\div$$
$$(\Pi_{id\_trabajador}\ \text{TRABAJADORES})$$