Data exchange: semantics and query answering

Ronald Fagin, Phokion G.Kolaitis, Renée J.Miller,
Lucian Popa

Cristián Muñoz Urbancic

Motivación

- Transferencia de datos entre aplicaciones independientes
- Creciente demanda debido a Internet
- Múltiples esquemas (BD relacionales, DTD , XML)

Data Exchange Problem

- Configuración:
  - ✓ $(S, T, \Sigma_{st}, \Sigma_t)$
  - ✓ $\forall x(\phi(x) \rightarrow \kappa(x))$
- Problema:
  - ✓ Dado I sobre S
  - ✓ Materializar J tal que:
    Satisface Σt
    I y J satisfacen Σst

Data Exchange Problem

- Conjunto de soluciones Sol(I)
- Algoritmo para verificar si existe solución y encontrarla (Tiempo Polinomial).
- Solución Universal.

Query Answering

- Para cuales queries se pueden obtener las respuestas certeras utilizando solo la solución J

- Análisis de complejidad.

  - ✓ Dada una configuración: (S, T, Σ_{st} , Σ_t)
  - ✓ I: instancia fuente, J: solución.
  - ✓ Sea q , query sobre J.
  - ✓ Encontrar las respuestas certeras de q con respecto de I

Data Exchange

- Un esquema: R = {*R*1, . . . , *Rk*}

  - ✓ Símbolo de relación
  - ✓ Cada relación de aridad m, tiene m atributos (las columnas)
  - ✓ I sobre R, se asocia para cada símbolo en R. I(Ri)
  - ✓ Sea t, una tupla que ocurre en R, entonces esta tupla es un hecho.

Data Exchange

- Configuración:

  - ✓ **S** = {*S*1, . . . , *Sn*}
  - ✓ **T** = {*T*1, . . . , *Tm*}
  - ✓ Dos esquemas disjuntos
  - ✓ I sobre S , *source instance*
  - ✓ J sobre T, *target instance*
  - ✓ Σ_{st}: *source to target dependencies*
  - ✓ Σ_{t} : *target dependencies*
  - ✓ $\langle I, J \rangle$ sobre **S**$\cup$**T**

    K(Si) = I (Si) and K(Tj) = J(Tj)

Data Exchange

- Definición 2.1 (The data exchange problem)
  - Dada la configuración (fija):
    $(S, T, \Sigma_{st}, \Sigma_t)$
  - Encontrar $\langle I, J \rangle$ tal que :
    $\langle I, J \rangle$ cumple con Σ_{st}
    J cumple con Σ_t
  - J es solución para I. $J \in Sol(I)$

Data Exchange

- Dependencias (Σ_{st}):

$$\forall \mathbf{x}(\phi_{\mathbf{S}}(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi_{\mathbf{T}}(\mathbf{x}, \mathbf{y})),$$

- Dependencias (Σ_t):

$$\forall \mathbf{x}(\phi_{\mathbf{T}}(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi_{\mathbf{T}}(\mathbf{x}, \mathbf{y})) \qquad \forall \mathbf{x}(\phi_{\mathbf{T}}(\mathbf{x}) \rightarrow (x_1 = x_2)).$$

Data Exchange

- Ejemplo:

$$\Sigma_{st}: \begin{array}{l} P(a,b,c) \rightarrow \exists Y \exists Z\ T(a,Y,Z), \\ Q(a,b,c) \rightarrow \exists X \exists U\ T(X,b,U), \\ R(a,b,c) \rightarrow \exists V \exists W\ T(V,W,c), \end{array} \quad I = \begin{array}{l} \{P(a_0, b_0', c_0'), \\ Q(a_0'', b_0, c_0''), \\ R(a_0''', b_0''', c_0)\}. \end{array}$$

- Solución:

$$J_1 = \{T(a_0, b_0, c_0)\}, \qquad J_2 = \{T(a_0, b_0, Z_1), T(V_1, W_1, c_0)\}$$

$$J = \{T(a_0, Y_0, Z_0), T(X_0, b_0, U_0), T(V_0, W_0, c_0)\}$$

¿Cuál es la mejor solución?

Data Exchange

- Solución Universal:
  - Vars (*nulls*)
  - Const (*datos)*
  - Var ∩ Const = ϕ
  - Sea *K* una instancia sobre un esquema *R,* Vars*(K) conjunto de nulls que ocurren en las relaciones en K*

Data Exchange

- Definición 2.3:
  - ✓ *K1 y K2 instancias sobre R*
  - ✓ *R tiene valores en (Const U Vars) = CV*
  - ✓ *h:K1→K2, mapeo de CK(K1) a CV(K2)*
    - *h(c) = c , para toda constante.*
    - *R(t) de K1 → R(h(t)) de K2, para cada hecho t de K1*
  - ✓ *K1 ≅ K2 ("homomorficamente"), si existe un homorfismo en ambos sentidos. Es decir, h:K1→K2 y h':K2→K1*

Data Exchange

- Solución Universal (definición 2.4):

  Dada una configuración para el problema de data exchange:

  $(S, T, \Sigma_{st}, \Sigma_t)$

  I es las fuente, entones *J* es solución universal si para todo *J'*

  que pertenece a *Sol(I), existe un homomorfismo h: J→J'*

- Volviendo al ejemplo anterior

$$J_1 = \{T(a_0, b_0, c_0)\}.$$

$$J_2 = \{T(a_0, b_0, Z_1), T(V_1, W_1, c_0)\}$$

$$J = \{T(a_0, Y_0, Z_0), T(X_0, b_0, U_0), T(V_0, W_0, c_0)\}$$

Data Exchange

- Propiedades.

  - Sea I una fuente, J y J' soluciones universales para I. Entonces J y J' son homorficamente equivalentes.

  - *I , I' dos fuentes. J* es solución para *I, J'* es solucion para I'. Entonces ,

    $Sol(I) \subseteq Sol(I') \leftrightarrow \exists h: J' \rightarrow J.$

    Es decir Sol(I) = Sol(I') si J y J' son *homomorficamente equivalentes.*

Encontrando la solución Universal

- Existe una solución universal?
- Como encontrarla?
- Veremos dos formas de abordar este problema.

Encontrando la solución Universal

- Generación canónica de la solución universal.
  - Si termina encuentra la solución.
  - Si falla, no existe solución.
  - Caza finito puede no existir para la configuración

Encontrando la solución Universal

- Se introduce la clase de:

  weakly acyclic tgds (tuple generating dependencies Σ_{st})

  - Garantiza el termino del algoritmo en tiempo polinomial
  - Es decir , puede verificar la existencia de una solución en tiempo polinomial. Y además, encontrar dicha solución en el mismo tiempo.

Generación canónica de la Solución universal.

- Punto de partida $\langle I, \phi \rangle$
- **"Cazar"** $\langle I, \phi \rangle$, aplicando las dependencias en algún orden arbitrario.
- Aplicar las dependencias mientras sea posible.(Loop)

Generación canónica de la Solución universal.

- *Chase step* (def. 3.1): Sea *K* una instancia.

(tgd):

Sea d una tgd

$$\phi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$$

Sea h:ϕ(x) $\rightarrow$*K* un homomorfismo , tal que no existe una extensión de h a h': ϕ(x) $\wedge$ ψ(x,y) $\rightarrow$ *K,.* Entonces *d* puede ser aplicado a *K* con homomorfismo *h.*

$$K \xrightarrow{d,h} K'.$$

Generación canónica de la Solución universal.

- *Chase step* (def. 3.1): Sea *K* una instancia.

(egd):

Sea d una egd:

$$\phi(x) \rightarrow (x_1 = x_2)$$

Sea $h:\phi(x) \rightarrow K$ un homomorfismo , tal que $h(x_1) \neq h(x_2)$

1. Si h(x1) , h(x2) ∈ *Const, Fracaso*

$$K \xrightarrow{d,h} \perp$$

2. Si uno es constante, se reemplaza la constante en los nulls etiquetados. Si ambos son nulls, se reemplaza uno por el otro en todos lados.

$$K \xrightarrow{d,h} K'.$$

Generación canónica de la Solución universal.

- *Chase* (def. 3.2):

Sea Σ un conjunto de tdgs y egds, y *K una instancia*

1. A *chase sequence of K with* Σ es una secuencia de pasos (chase step), finita o infinita.

$$K_i \xrightarrow{d_i, h_i} K_{i+1},$$

2. Una *finite chase of K with* Σ *es una secuencia* finita de pasos (chase steps) .

   a. Porque falló para algún paso m (falló) $K_m = \perp$
   b. No hay dependencia ni homomorfismo que que se pueda aplicar que se pueda aplicar.(éxito)

Generación canónica de la Solución universal.

- Teormea 3.3

  Asumiendo que Σ_{st} consiste de tgds y Σ_t consiste de tgds y egds.

1. Si J es resultado de un caso finito, J es solución universal.
2. Si existe un caso de fallo, entonces no existe solución

Lema 3.4:

Si $K_1 \xrightarrow{d,h} K_2$ distinto de “fallo” . Sea *K* una instancia que satisface d y existe un homomorfismo $h_1 : K_1 \to K$ Entonces, existe un homomorfismo h_2 de K_2 a K.

Generación canónica de la Solución universal.

- Ejemplo:

$$\Sigma_{st} = \{\, \texttt{DeptEmp}(d, n, e) \rightarrow \exists M(\texttt{Dept}(d, M, n) \wedge \texttt{Emp}(e, d)) \,\}$$

$$\Sigma_{t} = \{\, \texttt{Dept}(d, m, n) \rightarrow \exists D\ \texttt{Emp}(m, D),$$

$$\texttt{Emp}(e, d) \rightarrow \exists M \exists N\ \texttt{Dept}(d, M, N) \,\}.$$

I={DeptEmp(CS, Mary, E003)}

Generación canónica de la Solución universal.

- En la primera iteración, se genera:

$$J_1 = \{\texttt{Dept}(CS, M, Mary), \texttt{Emp}(E003, CS)\},$$

- M es etiquetada como null, codifica el mrg_id desconocido.

- Quedan dependencias Σ_t. Por lo tanto la búsqueda no se detiene.

$$\Sigma_t = \{\, \texttt{Dept}(d, m, n) \rightarrow \exists D\, \texttt{Emp}(m, D),$$
$$\texttt{Emp}(e, d) \rightarrow \exists M \exists N\, \texttt{Dept}(d, M, N)\,\}$$

Generación canónica de la Solución universal.

- La primera dependencia de Σ_t

$$\texttt{Dept}(d, m, n) \rightarrow \exists D\ \texttt{Emp}(m, D)$$

requiere que M aparezca en EMP

- Por lo que genera EMP(M,D)
- Ahora se aplica la segunda dependencia.

$$\texttt{Emp}(e, d) \rightarrow \exists M \exists N\ \texttt{Dept}(d, M, N)$$

- Esta genera Dept(D,M',N'). Asi sucesivamente por lo que claramente cae en un loop infinito.

$$J = \{\texttt{Dept}(CS, M, Mary), \texttt{Emp}(E003, CS), \texttt{Emp}(M, D), \texttt{Dept}(D, M', N'), \ldots\}.$$

Generación canónica de la Solución universal.

- Por lo tanto, no existe un caza finito
- Para satisfacer todas las dependencias, se requeriría una instancia *J* infinita.
- Algoritmo no es capaz de encontrar una solución
- Sin embargo, si existen soluciones finitas:

$$J' = \{\texttt{Dept}(CS, E003, Mary), \texttt{Emp}(E003, CS)\},$$
$$J'' = \{\texttt{Dept}(CS, M, Mary), \texttt{Emp}(E003, CS), \texttt{Emp}(M, CS)\}.$$

Cual es el problema de estas soluciones?

- No son Soluciones universales. (No definen el espacio de soluciones.)

Caza en tiempo polinomial

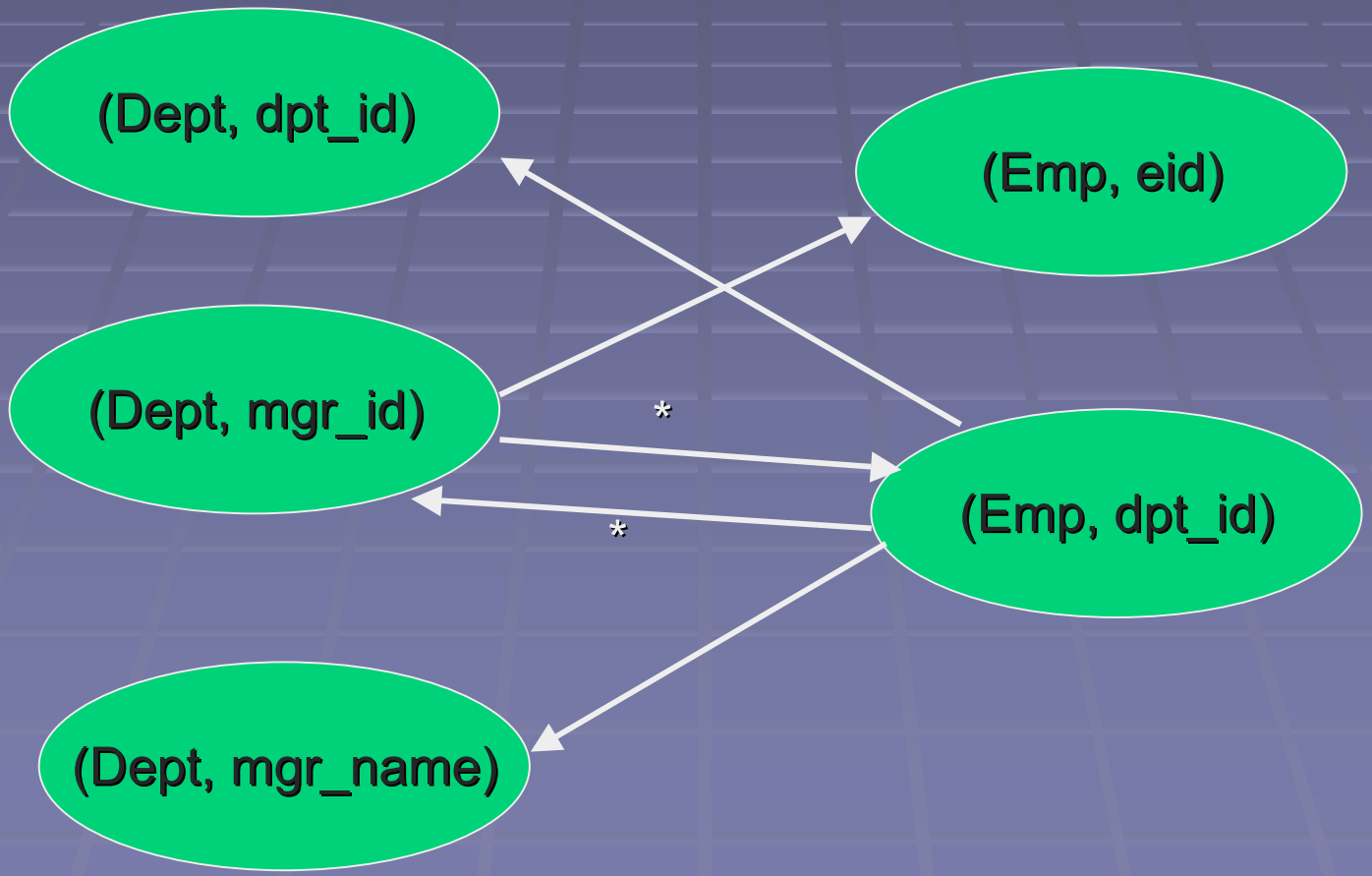
- Weakly acyclic tgds (def 3.7) .
- Construir el grafo de dependencias.
  - Por cada para (R, A), crear un nodo
  - Para cada tgd $\phi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ y para cada x que ocurre en ψ:
    - Por cada ocurrencia de x en ϕ en la posición (R, A_i)=NR:
      - ☞ Por cada ocurrencia de x en ψ *en la posición (S, B_j)=NS, agregar un arco de NR a NS.*
      - ☞ Por cada variable existencial cuantificada y por cada ocurrencia de y en ψ *en la posición (T, C_k)=NT agregar un arco (*) de NR a NT.*

Entonces Σ es acíclico si el grafico generado no posee ciclos que tengan arcos especiales (*)

Caza en tiempo polinomial

- Ejemplo

$$\Sigma_t = \{\, \mathtt{Dept}(d, m, n) \rightarrow \exists D\, \mathtt{Emp}(m, D),$$
$$\mathtt{Emp}(e, d) \rightarrow \exists M \exists N\, \mathtt{Dept}(d, M, N) \,\}$$



Caza en tiempo polinomial

- Teorema 3.9

  Sea Σ la unión de un conjunto de tgds débilmente acíclicas con un conjunto de egds. Entonces, existe un polinomio en el tamaño de la instancia K que acota el largo de todas las secuencias de caza (chase sequence).

- Corolario

  La existencia de una solución puede ser verificada en tiempo polinomial. Si existe, puede ser también producida en tiempo polinomial.

Query Answering

- Se adopta la noción de “certain answers” para la semántica de “query answering”.
- Definicion 4.1

  Dado $(S, T, \Sigma_{st}, \Sigma_t)$

  - Sea q una query k-aria con $k \geq 0$, sobre el esquema *T y* la fuente *I. Certain(q,I)* es el conjunto de todas las tuplas de constantes de *I ,* tal que para cada solución $J \in Sol(I)$, $t \in q(J)$.
  - En particular sea q una consulta de aridad k=0. Entonces, *true* denota el conjunto con una 0-aria tupla, y *false* es un conjunto vacío. Entonces, *certain(q,I)=true,* significa que *q(J)* es true para cada $J \in Sol(I)$, *y certain(q,I)=false* significa que existe una solución $J \in Sol(I)$ *tal que q(J) es false.*

Query Answering

- Certain answers, implica computar sobre todo el universo de soluciones, que puede ser infinito.

- Por lo que es necesario identificar situaciones en las cuales las certain answers de una consulta puedan ser computadas evaluando *q* en una solución fija (una instancia en el espacio de soluciones).

Query Answering

- Proposición 4.2: Dado,

$$(S, T, \Sigma_{st}, \Sigma_t)$$

- Sea q un disyunción de consultas conjuntivas

  $q(\mathbf{x}) = q1(x) \vee q2(x) \vee q3(x) \vee \ldots.. qn(x)$

  I la fuente y *J* la solución universal, entonces ***certain(q,I)=q(J)***

- Se cumple en ambos sentidos.
- Corolario:

  Para cada fuente I, el conjunto de certain answers, puede ser computada en tiempo polinomial.

Query Answering

- Misma configuración anterior, pero ahora las consultas conjuntivas tienen desigualdades.
- Ninguna solución universal puede ser usada par obtener las certain answers de q con respecto a *I, certain(q,I)*.
- Sin embargo, se puede reescribir q* con desigualdades, tal que las *certain(q,I)* pueden ser obtenidas evaluando q* en la solución universal canónica.

Query Answering

- Proposición 4.4:
- Sea S el símbolo de una relación binaria en la fuente y T en el objetivo
- Sea

$$S(x, y) \rightarrow \exists z(T(x, z) \wedge T(z, y))$$

una dependencia que pertenece a Σ_{st}

- Sea la consulta q= $\exists x \exists y(T(x, y) \wedge (x \neq y))$
- Entonces:
  - *∃ I tal que certain(q,I)=false , pero q(J)=true ∀ J ∈ Sol(I)*
  - *Sea q*=* $\exists x \exists y \exists z(T(x, z) \wedge T(z, y) \wedge (x \neq y))$ *, si J es la solución universal de I, entonces certain(q,I)=q*(J)*

Complejidad Query Answering

- Definicion 5.1:

  Dado (S, T, Σ_{st} , Σ_t)

  - Sea q una consulta k-aria sobre T. Dada *I* sobre *S* y una k-tupla t de constantes de *I*. *t* $\in$ *certain(q,I)?*
  - Sea q , la consulta booleana sobre *T*. Dada *I* sobre *S* ,*certain(q,I) = true?*
  - Sea *C* una clase de complejidad y *Q* una clase de consultas sobre *T*.
    - Computar las certain answers de querys en *Q* está en *C* si $\forall$ q $\in$ *Q, computar q* $\in$ C.
    - Computar las certain answers de querys en *Q es C-complete* si está en C, *y hay al menos una q* $\in$ *Q tal que las certain answers de q es un problema C-completo*

Complejidad Query Answering

- Dados los estudios y resultados en sistemas de integración de datos *LAV, con vistas "sound" y definido por consultas conjuntivas.*
- Podemos ver LAV como un caso particular de (S, T, Σ_{st} , Σ_t).
  - Donde $\Sigma_{st} = \phi$
  - Cada dependencia (source to target) es de la forma $S_i(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi_{\mathbf{T}}(\mathbf{x}, \mathbf{y})$ donde S_i es una relación del esquema fuente S, y ψ_t son una conjunción de fórmulas atómicas sobre el esquema objetivo *T.*

Esta configuración en adelante la llamaremos *LAV*

Complejidad Query Answering

- Computar certain answers en LAV de consultas con desigualdades es coNP.
- Σ_{st} , Σ_{t} conjunto acíclico de dependencias.
- q, es la unión de consultas conjuntivas con desigualdades.
- Entonces *(Teorema 5.2), computar las "certain answers" de q es coNP.*

Complejidad Query Answering

- En el caso que Σ_t no tiene variables existenciales cuantificables, es fácil de probar utilizando “small model property”. Es decir, encontrar un testigo.
- En el caso más general, que contenga variables existenciales cuantificadas. Se utiliza disjunctive chase.
- Para decidir si $t \in certain(q,I)$, sustituimos t en q.
- Así obtenemos una query q booleana, por lo que el problema se reduce a ver si $certain(q,I)=true$

Complejidad Query Answering

- q = q1 v q2
- q1 es la disyunción de un conjunto *C* de consultas conjuntivas sin desigualdad.
- q2 es la disyunción de un conjunto *C'* de consultas conjuntivas con al menos una desigualdad.
- Cada elemento de *C'* es de la forma

$$\exists \mathbf{x} \left(\phi(\mathbf{x}) \wedge \left(\bigwedge_i (x_i^1 \neq x_i^2) \right) \right)$$

- Entonces es fácil ver que la negación de q2 es equivalente a la conjunción de un conjunto de fórmulas *E* de la forma.

$$\forall \mathbf{x} \left(\phi(\mathbf{x}) \rightarrow \left(\bigvee_i (x_i^1 = x_i^2) \right) \right)$$

(disjunctive egds)

Complejidad Query Answering

Lema 5.3: Las siguientes expresiones son equivalentes

- *Certain(q,I)=false*
- *Existe una solución J* para I tal que J* satisface E y no satisface ninguna de las consultas conjuntivas de C (NP)*

Complejidad Query Answering

- Definición 5.4 (disjunctive chase step)
  - Sea *K* una instancia y *e* una *degd.*

$$\phi(\mathbf{x}) \rightarrow ((x_1^1 = x_1^2) \vee \cdots \vee (x_l^1 = x_l^2))$$

  - Denotamos $e_1, \ldots\ldots, e_l$. Que serán los egds asociados a *e*

$$\phi(\mathbf{x}) \rightarrow (x_1^1 = x_1^2), \ldots, \phi(\mathbf{x}) \rightarrow (x_l^1 = x_l^2),$$

  - Sea h:ϕ(x) $\rightarrow$*K* un homomorfismo , tal que:

$$h(x_1^1) \neq h(x_1^2), \ldots, h(x_l^1) \neq h(x_l^2).$$

  - Entonces decimos que *e* puede ser aplicado a *K* con un homomorfismo *h.*

Complejidad Query Answering

- Utilizamos la definición 3.1 para cada e_i , i=1……..l.

- Entonces para cada e_i se distinguen los mismos dos casos.
  - K_1,……K_l son $\perp$, entonces el caso es un “failure”, y lo denotamos como $K \xrightarrow{e,h} \perp$
  - En otro caso, sean K_{i1},…….K_{ip} elementos de {K_1,……K_l} que no son $\perp$. Entonces el resultado de aplicar *e* a *K* con *h* es el conjunto {K_{i1},……K_{ip} }

$$K \xrightarrow{e,h} \{K_{i_1}, \ldots, K_{i_p}\}.$$

Complejidad Query Answering

- Definición 5.5*: Disjunctive Chase*
- *Sea Σ* un conjunto de tgds y egds. Y sea *E con conjunto de degds, y sea K una instancia.*
- Un árbol de caza de *K con* Σ U *E, es tal que:*
  - *K es la raiz*
  - Por cada nodo Kj , { Kj1,……Kjr } son sus hijos, por lo que debe existir d ∈ Σ U *E y* un homomorfismo *h tal que:*

$$K_j \xrightarrow{d,h} \{K_{j_1}, \ldots, K_{j_r}\}$$

Finite disjunctive chase es un árbol caza finito. Esto quiere decir que:

a. Km=⊥
b. No existe d ∈ Σ U *E y* un homomorfismo *h tal que* d pueda ser aplicado a Km con h.

Complejidad Query Answering

- Al igual que con *chase,* puede no existir un *finite chase.*
- Sin embargo, al igual que para el teorema 3.9, se puede exigir que las tgds sea acíclicas.
- Bajo esta condición se puede probar la siguiente proposición.

Complejidad Query Answering

- Propsición 5.6
  - Sea Σ la union de tgds acíclicos con egds.
  - Sea *E un* conjunto de degds.
  - Sea *K una instancia.*
  - Cada árbol de caza de *K con* Σ U *E es finito.* Mas aún, la profundidad del árbol está acotado por un polinomio en el tamaño de *K.*

Complejidad Query Answering

- Proposición 5.7
- Asumiendo la configuración anterior y un conjunto C de consultas conjuntivas booleanas.
- Sea *I la instancia fuente.*
- Las siguientes afirmaciones son equivalentes.
  - Existe una solución *J** para *I* tal que *J** satisface *E,* y no satisface ninguna de las consultas booleanas.
  - Existe una solución universal J para I, existe una "*finite disjunctive chase"* de J con Σ U E, y existe una hoja $J^* \neq \perp$ de T tal que J* no satisface ninguna de las consultas en C

Complejidad Query Answering

- Lema 5.8:
- Sea $K \xrightarrow{e,h} \{K_{i_1}, \ldots, K_{i_p}\}$ un disjunctive chase step no fallido.
- Sea K* una instancia tal que existe un homomorfismo de K a K*.
- Entonces $\exists\, j \in \{1, \ldots\ldots\ldots, l\}$ tal que g: $K_j \rightarrow K$ es un homomorfismo.

Complejidad Query Answering

- En una configuración LAV, computar respuestas certeras de consultas conjuntivas booleanas con seis o más desigualdades es coNP-completo.(Teorema 5.9)

- En una configuración LAV, computar respuestas certeras de consultas conjuntivas booleanas con 2 desigualdades es coNP-completo, sin importar donde estén ubicadas estas desigualdades. (Teorema 5.11)

Query Answering

- Dada una configuración LAV
- Σt , es la unión de tgds acíclicos y egds.
- Sea q la unión de consultas conjuntivas con a lo mas una desigualdad.
- Entonces, existe un algoritmo que de tiempo polinomial que computa las respuestas certeras de q respecto d I

Query Answering

- Teromea 5.14
- Dado una configuración LAV
- Una consulta booleana con una desigualdad para la cual no existe q* de primer orden tal que:
  - Para cada Instancia *I, existe una solución universal tal que certain(q,I)=q*(J).*