Presentación IIC3432

Data Exchange: Getting to the Core

Ronald Fagin, Phokion G.Kolaitis, Lucian Popa

28-Mayo-2007

El problema

- El problema de Data Exchange se refiere a tomar los datos en un esquema fuente (source schema) y crear una instancia del esquema destino (target schema).

- Para ello es necesario respetar las dependencias entre la fuente y el destino, llamadas Source to Target Dependencies. También deben respetarse las dependencias del destino, llamadas Target Dependencies.

¿Por qué nos interesa?

¿Por qué nos interesa?

- Últimamente la necesidad de Data Exchange ha incrementado dada la proliferación de bases de datos en la red las cuales utilizan diversos formatos.

¿Por qué nos interesa?

- Últimamente la necesidad de Data Exchange ha incrementado dada la proliferación de bases de datos en la red las cuales utilizan diversos formatos.
- Es un problema que concierne al área de e-business.

¿Por qué nos interesa?

- Últimamente la necesidad de Data Exchange ha incrementado dada la proliferación de bases de datos en la red las cuales utilizan diversos formatos.
- Es un problema que concierne al área de e-business.
- Tiene relación con el problema de Data Integration.

¿Por qué nos interesa?

- Últimamente la necesidad de Data Exchange ha incrementado dada la proliferación de bases de datos en la red las cuales utilizan diversos formatos.
- Es un problema que concierne al área de e-business.
- Tiene relación con el problema de Data Integration.
- Se usa en sistemas como Clio (de IBM).

Formalizando el problema

Podemos ver el problema de Data Exchange como una cuadrupla $(\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$

Formalizando el problema

Podemos ver el problema de Data Exchange como una cuadrupla $(\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$

- **S** Es el esquema fuente, Source.

Formalizando el problema

Podemos ver el problema de Data Exchange como una cuadrupla $(\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$

- **S** Es el esquema fuente, Source.
- **T** Es el esquema destino, Target.

Formalizando el problema

Podemos ver el problema de Data Exchange como una cuadrupla $(\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$

- **S** Es el esquema fuente, Source.
- **T** Es el esquema destino, Target.
- Σ_{st} Es el conjunto de dependencias que entregan restricciones entre **S** y **T**.

Formalizando el problema

Podemos ver el problema de Data Exchange como una cuadrupla $(\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$

- **S** Es el esquema fuente, Source.
- **T** Es el esquema destino, Target.
- Σ_{st} Es el conjunto de dependencias que entregan restricciones entre **S** y **T**.
- Σ_t Es el conjunto de dependencias que entregan restricciones de **T**.

Formalizando el problema

Podemos ver el problema de Data Exchange como una cuadrupla $(\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$

- **S** Es el esquema fuente, Source.
- **T** Es el esquema destino, Target.
- Σ_{st} Es el conjunto de dependencias que entregan restricciones entre **S** y **T**.
- Σ_t Es el conjunto de dependencias que entregan restricciones de **T**.

Dada una instancia I sobre **S** buscamos una instancia J sobre **T** tal que I junto con J satisfagan las restricciones Σ_{st}. Además J debe satisfacer Σ_t

Soluciones

- La solución del problema de Data Exchange es la instancia J sobre $\mathbf{T}$.
- Pueden existir múltiples instancias que cumplan las restricciones o sea múltiples soluciones.

Ejemplo Simple

S:

$$cine(t, s)$$

$$pelicula(t, a)$$

T:

$$cartelera(t, s)$$

$$actor(n, t)$$

Σ_{st}:

$$cine(t, s) \rightarrow cartelera(t, s)$$

$$pelicula(t, a) \rightarrow \exists S \ cartelera(t, S)$$

$$pelicula(t, a) \rightarrow \exists N \ actor(N, t)$$

$\Sigma_t = \emptyset$

...Ejemplo Simple

Para la instancia I:

$$cine(spiderman, sala1)$$

$$pelicula(spiderman, 2007)$$

$$cine(shrek, sala2)$$

$$pelicula(shrek, 2007)$$

...Ejemplo Simple

Para la instancia I:

$$cine(spiderman, sala1)$$

$$pelicula(spiderman, 2007)$$

$$cine(shrek, sala2)$$

$$pelicula(shrek, 2007)$$

Hay muchas (de hecho infinitas) soluciones, o sea J's.

...Ejemplo Simple

Algunas soluciones:

J1:

$cartelera(spiderman, sala1)$ $actor(N1, spiderman)$
$cartelera(shrek, sala2)$ $actor(N2, shrek)$

J2:

$cartelera(spiderman, sala1)$ $actor(N, spiderman)$
$cartelera(shrek, sala2)$ $actor(N, shrek)$

J3:

$cartelera(spiderman, sala1)$ $actor(N2, shrek)$
$cartelera(shrek, sala2)$ $actor(N3, spiderman)$
$actor(N1, spiderman)$ $actor(N4, shrek)$

Soluciones Universales

De las múltiples soluciones podemos distinguir un conjunto particular con las siguientes propiedades:

- Son homomorficamente equivalentes (existen homomorfismos entre ellas).
- Son las más generales entre las soluciones.

Soluciones Universales en el Ejemplo

J1:

$cartelera(spiderman, sala1)$ $actor(N1, spiderman)$
$cartelera(shrek, sala2)$ $actor(N2, shrek)$

J2:

$cartelera(spiderman, sala1)$ $actor(N, spiderman)$
$cartelera(shrek, sala2)$ $actor(N, shrek)$

J3:

$cartelera(spiderman, sala1)$ $actor(N2, shrek)$
$cartelera(shrek, sala2)$ $actor(N3, spiderman)$
$actor(N1, spiderman)$ $actor(N4, shrek)$

Soluciones Universales en el Ejemplo

J1: Solución Universal

$$cartelera(spiderman, sala1) \quad actor(N1, spiderman)$$
$$cartelera(shrek, sala2) \quad actor(N2, shrek)$$

J2:

$$cartelera(spiderman, sala1) \quad actor(N, spiderman)$$
$$cartelera(shrek, sala2) \quad actor(N, shrek)$$

J3:

$$cartelera(spiderman, sala1) \quad actor(N2, shrek)$$
$$cartelera(shrek, sala2) \quad actor(N3, spiderman)$$
$$actor(N1, spiderman) \quad actor(N4, shrek)$$

Soluciones Universales en el Ejemplo

J1: Solución Universal

$$cartelera(spiderman, sala1) \quad actor(N1, spiderman)$$
$$cartelera(shrek, sala2) \quad actor(N2, shrek)$$

J2: No es Solución Universal

$$cartelera(spiderman, sala1) \quad actor(N, spiderman)$$
$$cartelera(shrek, sala2) \quad actor(N, shrek)$$

J3:

$$cartelera(spiderman, sala1) \quad actor(N2, shrek)$$
$$cartelera(shrek, sala2) \quad actor(N3, spiderman)$$
$$actor(N1, spiderman) \quad actor(N4, shrek)$$

Soluciones Universales en el Ejemplo

J1: Solución Universal

$$cartelera(spiderman, sala1) \quad actor(N1, spiderman)$$
$$cartelera(shrek, sala2) \quad actor(N2, shrek)$$

J2: No es Solución Universal

$$cartelera(spiderman, sala1) \quad actor(N, spiderman)$$
$$cartelera(shrek, sala2) \quad actor(N, shrek)$$

J3: Solución Universal

$$cartelera(spiderman, sala1) \quad actor(N2, shrek)$$
$$cartelera(shrek, sala2) \quad actor(N3, spiderman)$$
$$actor(N1, spiderman) \quad actor(N4, shrek)$$

Cores y Soluciones Universales

- Es natural preguntarse cual de estas soluciones universales es la mejor.
- Se propone utilizar el criterio de *minimalidad*.
- Las soluciones universales mínimas (isomórficas) son llamadas Core.

Cores y Soluciones Universales

- Es natural preguntarse cual de estas soluciones universales es la mejor.
- Se propone utilizar el criterio de *minimalidad*.
- Las soluciones universales mínimas (isomórficas) son llamadas Core.

Core:

dadas 2 estructuras A y C, C es Core para A si existe un homomorfismo desde A a C, pero no entre A y cualquier subestructura propia de C.

El Core en el Ejemplo Anterior

En el ejemplo la solución J1 es el Core, solución universal minimal.

J1:

$$\begin{array}{ll} cartelera(spiderman, sala1) & actor(N1, spiderman) \\ cartelera(shrek, sala2) & actor(N2, shrek) \end{array}$$

El Core en el Ejemplo Anterior

En el ejemplo la solución J1 es el Core, solución universal minimal.

J1: ¡Core!

$$cartelera(spiderman, sala1) \quad actor(N1, spiderman)$$
$$cartelera(shrek, sala2) \quad actor(N2, shrek)$$

Buscando la mejor Solución

Nos interesa por tanto encontrar la mejor solución para el caso de Data Exchange, esta viene dada por el Core de las soluciones universales.

Buscando la mejor Solución

Nos interesa por tanto encontrar la mejor solución para el caso de Data Exchange, esta viene dada por el Core de las soluciones universales.

¿Que tan difícil es encontrarlo?

Core Identification

Es el siguiente problema de decisión:

> *Dada 2 estructuras* $\mathbf{A}$ *y* $\mathbf{B}$ *sobre un esquema* $\mathbf{R}$, *tal que* $\mathbf{B}$ *es una subestructura de* $\mathbf{A}$, *¿es core*$(\mathbf{A}) = \mathbf{B}$*?*

Complejidad de Core Identification

Complejidad de Core Identification

- Primero recordemos las clases de complejidad.

Complejidad de Core Identification

- Primero recordemos las clases de complejidad.
- Es fácil demostrar que es un problema NP-Hard, reduciendo 3-colorabilidad.

Complejidad de Core Identification

- Primero recordemos las clases de complejidad.
- Es fácil demostrar que es un problema NP-Hard, reduciendo 3-colorabilidad.
- Se demuestra que es un problema DP-Completo.

Complejidad de Core Identification

Para demostrar que es DP-Completo

Complejidad de Core Identification

Para demostrar que es DP-Completo

- Se demuestra que está en DP

Complejidad de Core Identification

Para demostrar que es DP-Completo

- Se demuestra que está en DP
  - Homomorfismo está en NP

Complejidad de Core Identification

Para demostrar que es DP-Completo

- Se demuestra que está en DP
  - Homomorfismo está en NP
  - Core Recognition está en Co-NP

Complejidad de Core Identification

Para demostrar que es DP-Completo

- Se demuestra que está en DP
  - Homomorfismo está en NP
  - Core Recognition está en Co-NP
- Demostrar que es DP-Hard, reduciendo 3-colorabilidad/no 3-colorabilidad.

Complejidad de Core Identification para Data Exchange

Estos resultados no son directamente aplicables en el caso de Data Exchange, pues para este problema se calcula el Core de una solución universal y no de una instancia particular.

Complejidad de Core Identification para Data Exchange

Estos resultados no son directamente aplicables en el caso de Data Exchange, pues para este problema se calcula el Core de una solución universal y no de una instancia particular.

Puede existir un algoritmo polinomial

Complejidad de Core Identification para Data Exchange

Estos resultados no son directamente aplicables en el caso de Data Exchange, pues para este problema se calcula el Core de una solución universal y no de una instancia particular.

Puede existir un algoritmo polinomial ¿existe?

Algoritmos Polinomiales

Se presentan 2 algoritmos polinomiales para los casos particulares en que Σ_{st} es un conjunto de tgds y Σ_t es un conjunto de egds.

- Un algoritmo Greedy (codicioso, avaro, voraz, etc).
- Algoritmo de Blocks.

Ambos algoritmos utilizan el procedimiento Chase para encontrar una solución pre-universal canónica.

Algoritmo Greedy

- Usar Chase sobre I con Σ_{st} para obtener solución pre-universal canónica J.
- Usar Chase sobre J con Σ_t si Chase falla, retornar falla, sino J' será la solucion Universal canónica.
- Inicializar J^* como J'.
- Mientras haya un *fact* $R(t)$ en J^* tal que $\langle I, J^* - \{R(t)\}\rangle$ satisface Σ_{st} entonces $J^* = J^* - \{R(t)\}$
- Retornar J^*.

Algoritmo de Blocks

- Usar Chase sobre I con Σ_{st} para obtener solución pre-universal canónica J.
- Computar los Blocks de J y hacer Chase de J usando Σ_t para obtener la solucion canónica J'. Si Chase falla, retornar falla, sino inicializar J'' con J'.
- Chequear si existe un endomorfismo h (homomorfismo que retorna una subinstancia) local en J, util (no uno a uno), de J'', si no existe detenerse y dar J'' como resultado.
- $J'' = h(J'')$ y repetir el paso anterior.

Otros Resultados

- No se puede obtener el Core usando solo Chase.
- El Core entrega la mejor aproximación correspondiente a las certain answers.
  - $certain(q, I) = \bigcap \{q(J) : J \text{ es solución de } I\}$
  - $u - certain(q, I) = \bigcap \{q(J) : J \text{ es solución universal de } I\}$

Conclusiones

- Se identifica la mejor solución para el problema, el Core de las soluciones Universales.
  - Ocupa el menor espacio, pues es minimal.
  - Entrega la mejor aproximación a las Certain Answers.
- A pesar de que el Core Identification es DP-Completo existen algoritmos polinomiales en el caso estudiado.

Trabajos Propuestos

- ¿Hay un algoritmo más eficiente para encontrar el Core de las Soluciones Universales?
- Ampliar los resultados a otros tipos de Dependencias.
- El trabajo solo considera Data Exchange sobre esquemas Relacionales, se propone ver hasta que punto es esto extendible a otros tipos de esquemas (XML).

Fin

Fin