

Presentación IIC3432

Data Integration: A Theoretical Perspective (Maurizio Lenzerini)

09-Abril-2007

Estructura de la presentación

Seguirá la estructura del paper de Lenzerini.

- 1 Introducción
- 2 Framework de Integración de Datos
- 3 Modelación
 - LAV
 - GAV
 - Comparación entre LAV y GAV
- 4 Procesamiento de Consultas
 - En LAV
 - En GAV
- 5 Inconsistencias entre Fuentes
- 6 Razonamiento sobre Consultas
- 7 Conclusiones

Estructura de la Presentación

- 1 Introducción
- 2 Framework de Integración de Datos
- 3 Modelación
 - LAV
 - GAV
 - Comparación entre LAV y GAV
- 4 Procesamiento de Consultas
 - En LAV
 - En GAV
- 5 Inconsistencias entre Fuentes
- 6 Razonamiento sobre Consultas
- 7 Conclusiones

El problema

- El problema de integración de datos se refiere a la combinación de los datos que se encuentran en diferentes fuentes, para entregar al usuario una vista unificada de estas.

Sistemas de unificación

Los sistemas de unificación de datos que se estudian en el paper son aquellos que utilizan un esquema global y un conjunto de fuentes.

- Las fuentes contienen los datos reales.
- Esquema global provee una vista integrada de las fuentes subyacentes.

Para modelar la relación entre esquema global y fuentes existen básicamente 2 tipos de mapeo:

- Local as View conocido como LAV.
- Global as View conocido como GAV.

Cualquiera de los métodos que usemos para realizar el mapeo implicará tener que reformular la consulta que hacemos sobre el esquema global en términos de consultas sobre las fuentes.

Estructura de la Presentación

- 1 Introducción
- 2 **Framework de Integración de Datos**
- 3 Modelación
 - LAV
 - GAV
 - Comparación entre LAV y GAV
- 4 Procesamiento de Consultas
 - En LAV
 - En GAV
- 5 Inconsistencias entre Fuentes
- 6 Razonamiento sobre Consultas
- 7 Conclusiones

Formalizamos un sistema de integración $\mathcal{I}$ como una tupla $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$

- $\mathcal{G}$ es el esquema global en un lenguaje $\mathcal{L}_{\mathcal{G}}$ sobre un alfabeto $\mathcal{A}_{\mathcal{G}}$.
- $\mathcal{S}$ el esquema de la fuente en un lenguaje $\mathcal{L}_{\mathcal{S}}$ sobre un alfabeto $\mathcal{A}_{\mathcal{S}}$.
- $\mathcal{M}$ el mapeo entre el $\mathcal{G}$ y $\mathcal{S}$. Un conjunto de aserciones del tipo:

$$q_{\mathcal{S}} \rightsquigarrow q_{\mathcal{G}}$$

$$q_{\mathcal{G}} \rightsquigarrow q_{\mathcal{S}}$$

donde $q_{\mathcal{G}}$ y $q_{\mathcal{S}}$ son consultas de igual aridad, sobre los lenguajes $\mathcal{L}_{\mathcal{M},\mathcal{G}}$ y $\mathcal{L}_{\mathcal{M},\mathcal{S}}$ respectivamente.

Una aserción del tipo $q_S \rightsquigarrow q_G$ indica que el concepto representado por la consulta q_S sobre las fuentes corresponde a aquel que representa la consulta q_G sobre el esquema global.

Una base de datos para un esquema no es más que una colección de conjuntos, un conjunto para cada uno de los símbolos del alfabeto del esquema.

Asumiremos que las bases de datos para nuestros sistemas son sobre un dominio fijo infinito Γ .

Tomemos una base de datos $\mathcal{D}$ para el esquema $\mathcal{S}$. El conjunto de bases de datos $\mathcal{B}$ que son legales en $\mathcal{I}$ respecto a $\mathcal{D}$ es:

$$\{\mathcal{B} \mid \mathcal{B} \text{ es legal respecto a } \mathcal{G} \text{ y satisface } \mathcal{M} \text{ respecto a } \mathcal{D}\}$$

- Una consulta q de aridad n es una fórmula con n variables libres.
- Si $\mathcal{D}$ es una base de datos entonces $q^{\mathcal{D}}$ es el conjunto de tuplas en $\mathcal{D}$ que satisfacen q .
- Dada $\mathcal{D}$ para $\mathcal{I}$, entonces $q^{\mathcal{I}, \mathcal{D}}$ es el conjunto de tuplas t en Γ tales que:

- $t \in q^{\mathcal{B}}$ para todas las bases de datos $\mathcal{B}$ que son legales para $\mathcal{I}$.

Este conjunto $q^{\mathcal{I}, \mathcal{D}}$ se denomina el conjunto de respuestas seguras respecto a $\mathcal{I}$ y a $\mathcal{D}$.

Estructura de la Presentación

- 1 Introducción
- 2 Framework de Integración de Datos
- 3 Modelación
 - LAV
 - GAV
 - Comparación entre LAV y GAV
- 4 Procesamiento de Consultas
 - En LAV
 - En GAV
- 5 Inconsistencias entre Fuentes
- 6 Razonamiento sobre Consultas
- 7 Conclusiones

Es la especificación de correspondencia entre los datos reales en las fuentes y el esquema global.

- ¿Como se hace el mapeo?
 - se define $\mathcal{S}$ en términos de $\mathcal{G} \rightarrow \text{LAV}$
 - se define $\mathcal{G}$ en términos de $\mathcal{S} \rightarrow \text{GAV}$
 - una mezcla de los 2 anteriores $\rightarrow \text{GLAV}$
 - mapeo entre fuentes sin esquema global $\rightarrow \text{P2P}$

Se basa en la idea de que el contenido de cada fuente s debe ser caracterizado en términos de una vista $q_{\mathcal{G}}$ sobre $\mathcal{G}$.

El mapeo $\mathcal{M}$ asocia cada elemento s de la fuente $\mathcal{S}$ a una consulta $q_{\mathcal{G}}$. Luego, las aserciones de $\mathcal{M}$ son del tipo:

$$s \rightsquigarrow q_{\mathcal{G}}$$

Noten que en LAV se favorece la extensibilidad del sistema, pues para agregar una nueva fuente basta con agregar el conjunto de aserciones.

Extensión

Se han propuesto 3:

- $as(s) = sound$: su extensión provee un subconjunto de tuplas que satisface la la vista q_G

$$s^D \subseteq q_G^B \quad \text{Default}$$

- $as(s) = complete$: su extensión provee un superconjunto de tuplas que satisface la la vista q_G

$$s^D \supseteq q_G^B$$

- $as(s) = exact$: su extensión provee exactamente el conjunto de tuplas que satisface la la vista q_G

$$s^D = q_G^B$$

- Information manifold.
- DWQ.
- Píxel.

Se basa en la idea de que el contenido de cada elemento g del esquema global, debe ser caracterizado en términos de una vista q_S sobre las fuentes.

El mapeo $\mathcal{M}$ asocia a cada elemento g en $\mathcal{G}$ una consulta q_f sobre la fuente. Luego, las aserciones de $\mathcal{M}$ son del tipo:

$$g \rightsquigarrow q_s$$

Noten que en GAV se favorece el procesamiento de consultas, pues le dice al sistema como utilizar las fuentes para poder obtener los datos.

Extensiones de las vistas

Se han propuesto 3:

- $as(g) = sound$: su provee un superconjunto de tuplas que satisface la la vista q_S

$$q_S^D \subseteq g^B$$

- $as(g) = complete$: su extension provee un subconjunto de tuplas que satisface la la vista q_S

$$q_S^D \supseteq g^B$$

- $as(g) = exact$: su extension provee exactamente el conjunto de tuplas que satisface la la vista q_S

$$g^B = q_S^D \quad \text{Default}$$

Sistemas que utilizan GAV

- Carnot.
- SIMS.
- Tsimmis.
- IBIS.

- En términos de calidad: la de LAV depende de la caracterización de la fuentes, mientras que en GAV depende de que tan bien se han compilado las fuentes en el esquema global.
- En términos de Extensibilidad: Es muy simple en LAV basta con agregar las aserciones, pero en GAV implica rehacer el esquema global.
- En terminos del procesamiento de consultas: en LAV necesita más razonamiento pues se debe replantear la consulta en terminos de las fuentes pero la información que entregan las aserciones son exactamente al revés(fuentes en términos de vista global), en GAV es más sencillo pues la información de las aserciones es exactamente la necesitada.

Es un enfoque mixto, entre LAV y GAV. Básicamente constituido por aserciones del tipo:

$$q_S \rightsquigarrow q_G$$

Estructura de la Presentación

- 1 Introducción
- 2 Framework de Integración de Datos
- 3 Modelación
 - LAV
 - GAV
 - Comparación entre LAV y GAV
- 4 Procesamiento de Consultas**
 - En LAV
 - En GAV
- 5 Inconsistencias entre Fuentes
- 6 Razonamiento sobre Consultas
- 7 Conclusiones

- Sólo poseemos información parcial de la base de datos global.
- Puede haber más de una base de datos que sea legal para el sistema.

Ejemplo

Tomemos un sistema $\mathcal{I}$, con $\mathcal{G}$ que contiene una relación binaria *couple* y dos constantes *Ann* y *Bill*, también tenemos dos fuentes, *female* y *male*. Las aserciones en LAV son:

$$female(f) \rightsquigarrow \{f, m \mid couple(f, m)\}$$

$$male(m) \rightsquigarrow \{f, m \mid couple(f, m)\}$$

La base de datos $\mathcal{D}$ tiene $female^{\mathcal{D}} = \{Ann\}$ y $male^{\mathcal{D}} = \{Bill\}$.

Tomemos $Q = \{x, y \mid couple(x, y)\}$

...Ejemplo

- si $as(s)=\text{sound}$: sólo podemos saber que existe una pareja que contiene a Ann y otra que contiene a Bill. La consulta Q es vacía pues $s^{\mathcal{D}}$ nos retorna solo un subconjunto de las respuestas.
- si $as(s)=\text{exact}$: podemos saber que existe una pareja que contiene a Ann y otra que contiene a Bill, pero como $s^{\mathcal{D}}$ nos retorna todas las respuestas a la consulta entonces retornará (Ann, Bill).

Existen 2 maneras de enfrentar este problema:

- view based query rewriting:
 - El objetivo es reescribir la consulta en términos de las fuentes.
 - El problema es que la consulta puede no existir, en estos casos se buscará la reescritura que la capture de la mejor forma posible.
- view based query answering:
 - No solo se nos entregan las aserciones, sino también las extensiones de las vistas.
 - El objetivo es computar las tuplas t de manera que el conocimiento de las extensiones nos permita la implicación lógica de que t es la respuesta a la consulta.

- El procesamiento de las consultas se basa en una estrategia de “desenrollado”.
- Teniendo la consulta sobre el alfabeto $\mathcal{A}_G$, basta con reemplazar cada elemento con su correspondiente q_S . y evaluar la consulta resultante.

El problema

El problema se presenta cuando el language permite restricciones de integridad y las vistas son *sound*. En estos casos la respuesta a la consulta podría ser incompleta.

Ejemplo

Tomamos $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$

- $\mathcal{G}$ constuido por las relaciones

employee(*Ecode*, *Ename*, *Ecity*)

company(*Ccode*, *Cname*)

employed(*Ecode*, *Ccode*)

- Las restricciones de integridad siguientes:

$\text{key}(\text{employee}) = \{Ecode\}$

$\text{key}(\text{company}) = \{Ccode\}$

$\text{employed}[Ecode] \subseteq \text{employee}[Ecode]$

$\text{employed}[Ccode] \subseteq \text{company}[Ccode]$

...Ejemplo

Las fuentes son las siguientes:

- s_1^D

12	calvin	rome	21
15	alice	hong kong	24

- s_2^D

AF	hotdog corp.
BN	banana ltd.

- s_3^D

12	AF
16	BN

...Ejemplo

las aserciones son:

$$employee \rightsquigarrow: \{x, y, z | s_1(x, y, z, w)\}$$

$$company \rightsquigarrow: \{x, y | s_2(x, y)\}$$

$$employee \rightsquigarrow: \{x, w | s_3(x, w)\}$$

...Ejemplo

Tomemos la consulta:

$$\{x \mid \text{employee}(x, y, z), \text{employed}(x, w)\}$$

- si reemplazáramos las aserciones sobre la consulta tendríamos que la consulta expresada en términos de las fuentes es $\{x \mid s_1^{\mathcal{D}}(x, y, z, w_1), s_3^{\mathcal{D}}(x, w_2)\}$ y por lo tanto la respuesta sería $\{12\}$.
¡Error!
- Si nos fijamos en las restricciones vemos que $\text{employed}[Ecode] \subseteq \text{employee}[Ecode]$ esta nos permite saber que 16 de la tabla $s_3^{\mathcal{D}}$ es también un *Ecode*, luego la respuesta correcta sería $\{12, 16\}$.

Estructura de la Presentación

- 1 Introducción
- 2 Framework de Integración de Datos
- 3 Modelación
 - LAV
 - GAV
 - Comparación entre LAV y GAV
- 4 Procesamiento de Consultas
 - En LAV
 - En GAV
- 5 Inconsistencias entre Fuentes**
- 6 Razonamiento sobre Consultas
- 7 Conclusiones

La formalización del sistema que teníamos estaba basada una interpretación en lógica de primer orden de las aserciones en el mapeo. Lamentablemente esto no sirve para trabajar con las inconsistencias entre las fuentes.

En la práctica esto se soluciona limpiando y transformando los datos que es toman desde las fuentes.

¿Que se necesita?

Se necesita un sistema que permita hacer consultas aun cuando los datos en las fuentes sean incoherentes respecto a las restricciones de integridad. Se presentan 2 posibilidades:

- Caracterizar el sistema sólo en terminos de las bases de datos globales que satisfacen las restricciones de integridad y aproximar de la mejor manera posible la satisfacción de las aserciones.
- Definir un orden entre las bases de datos globales $\mathcal{B}_1$ es mejor que $\mathcal{B}_2$, $\mathcal{B}_1 \gg_D \mathcal{B}_2$, si hay al menos una aserción para $\mathcal{B}_1$ que satisface el mapeo *sound* mejor que $\mathcal{B}_2$. Y en ninguna aserción pasa la contrario. Y tomar el $\mathcal{B}$ legal con respecto a $\mathcal{G}$ como la mejor de las bases de datos.

Estructura de la Presentación

- 1 Introducción
- 2 Framework de Integración de Datos
- 3 Modelación
 - LAV
 - GAV
 - Comparación entre LAV y GAV
- 4 Procesamiento de Consultas
 - En LAV
 - En GAV
- 5 Inconsistencias entre Fuentes
- 6 Razonamiento sobre Consultas
- 7 Conclusiones

Una consulta está contenida en otra relativo al conjunto de fuentes, si para cada extensión de las vistas las respuestas seguras a la primera consulta son idénticas a las respuestas seguras de la segunda.

Está demostrado que el problema de chequear el contenimiento relativo de una en el caso de consultas conjuntivas y vistas es Π_2^P Completo.

Un conjunto de vistas es Lossless con respecto a una consulta si, no importando la base de datos, se puede contestar la consulta con tan solo el contenido de las vistas.

Hay al menos 2 versiones de Lossless:

- Bajo vistas Sound: las vistas pueden contener información suficiente, pero la incompletitud de las fuentes pueden impedir obtener todas las respuestas que la consulta obtendría de la base de datos.
 - PSPACE-complete respecto a las vistas
 - EXSPACE-complete respecto a la consulta
- Bajo vistas Exactas: las vistas contienen la información suficiente para obtener la respuesta a la consulta.

Estructura de la Presentación

- 1 Introducción
- 2 Framework de Integración de Datos
- 3 Modelación
 - LAV
 - GAV
 - Comparación entre LAV y GAV
- 4 Procesamiento de Consultas
 - En LAV
 - En GAV
- 5 Inconsistencias entre Fuentes
- 6 Razonamiento sobre Consultas
- 7 Conclusiones

Hay muchos problemas abiertos (al menos habían):

- Manejo de inconsistencias.
- Incorporación de nociones de calidad y limpieza de datos.
- Construcción de un esquema global apropiado.
- Como optimizar la evaluación de consultas.

entre otros varios.