Propiedades de Clausura para Operadores de Mapeos

Jorge Pérez
PUC Chile
jperez@ing.puc.cl

Marcelo Arenas
PUC Chile
marenas@ing.puc.cl

Cristian Riveros
PUC Chile
crj@ing.puc.cl

RESUMEN

Una pregunta natural que surge en el contexto de operadores de mapeos entre esquemas, es si existe algún formalismo de especificación de mapeos que sea cerrado bajo cierto operador. Dados mapeos especificados en un lenguaje $\mathcal{L}$ y una operación entre ellos, ¿puede el resultado de la operación siempre expresarse en $\mathcal{L}$? La composición y el inverso son dos operadores de mapeos entre esquemas que han recibido recientemente considerable atención. A pesar de que el problema de clausura ha sido estudiado con cierto detalle para el operador de composición, poca investigación se ha conducido para estudiar el mismo problema en el caso del inverso. Atacar este problema es el objetivo principal de nuestra investigación. En este documento resumimos parte de nuestro proyecto presentando nuestros resultados preliminares, el trabajo en curso y los resultados esperados.

1. INTRODUCCIÓN

Un mapeo entre esquemas es una especificación de alto nivel que describe la relación entre los datos estructurados bajo dos esquemas de bases de datos independientes. Estas especificaciones cumplen un rol esencial en diversas tareas de intercambio e integración de datos.

Existen muchas maneras de especificar un mapeo entre esquemas. En el caso relacional generalmente se usan fórmulas en alguna lógica tomando como vocabulario al conjunto de tablas (o relaciones) de ambos esquemas. Por ejemplo, considere un esquema $\mathbf{S}_1$ con una relación $\texttt{Emp}(\texttt{nom}, \texttt{vive\_en}, \texttt{trabaja\_en})$ que almacena nombres de empleados y el lugar donde el empleado vive y trabaja. Considere además un esquema $\mathbf{S}_2$ con una relación $\texttt{Bus}(\texttt{nombre}, \texttt{destino})$ pensada para almacenar nombres de empleados que deben tomar el bus para llegar a su lugar de trabajo (destino). Una posible forma de relacionar estos esquemas es con la siguiente fórmula de lógica de primer orden:

$$\forall x \forall z (\exists y (\texttt{Emp}(x, y, z) \wedge y \neq z) \rightarrow \texttt{Bus}(x, z)). \quad (1)$$

La anterior fórmula esencialmente especifica que si un empleado vive en un lugar distinto al lugar donde trabaja, entonces debe tomar el bus para llegar a su trabajo.

La mayoría de los estudios que se encuentran en la literatura acerca de mapeos entre esquemas, abordan problemas que tienen que ver con cómo utilizar estos mapeos para llevar a cabo tareas de intercambio e integración de datos. Sin embargo, Bernstein [2] propone que para dar una solución integral a los problemas de integración de datos, se necesita desarrollar una teoría básica que no sólo diga cómo utilizar los mapeos para realizar ciertas tareas particulares, si no también cómo poder operar estos mapeos para reutilizarlos más allá de las tareas específicas para las que fueron creados. La idea es considerar los mapeos como *objetos* dignos de estudio por sí mismos, y estudiar operaciones de tipo algebraico sobre ellos. Considere el siguiente ejemplo de manipulación de mapeos de esquemas. Sea $\mathbf{S}_3$ un esquema con una relación $\texttt{Asignacion}(\texttt{nom}, \texttt{monto})$ que almacena los nombres de los empleados que tienen una asignación de movilización y el monto de ésta. Considere la siguiente fórmula que relaciona $\mathbf{S}_2$ y $\mathbf{S}_3$:

$$\forall x (\exists y \, \texttt{Bus}(x, y) \rightarrow \exists u \, \texttt{Asignacion}(x, u)). \quad (2)$$

Esta fórmula indica que si un empleado debe tomar el bus para llegar a su trabajo, entonces debe tener una asignación de movilización. Suponga ahora que se quiere relacionar los datos estructurados bajo $\mathbf{S}_1$ con los datos estructurados bajo $\mathbf{S}_3$ para una aplicación de intercambio de datos. El primer problema que surge es que no existe una especificación para tal intercambio. Sin embargo, si consideramos las fórmulas (1) y (2) podemos *inferir* que la fórmula

$$\forall x (\exists y \exists z (\texttt{Emp}(x, y, z) \wedge y \neq z) \rightarrow \exists u \, \texttt{Asignacion}(x, u)) \quad (3)$$

es un mapeo válido entre los esquemas $\mathbf{S}_1$ y $\mathbf{S}_3$. Intuitivamente lo que estamos haciendo es *derivar* desde un mapeo entre $\mathbf{S}_1$ y $\mathbf{S}_2$, y otro mapeo entre $\mathbf{S}_2$ y $\mathbf{S}_3$, un nuevo mapeo, esta vez entre $\mathbf{S}_1$ y $\mathbf{S}_3$. Esta operación se conoce como *composición* de mapeos y ha recibido recientemente considerable atención [4, 6].

Con el estudio de operadores entre mapeos de esquemas surgen muchas interrogantes. Lo primero es que no es para nada claro qué significa operar especificaciones lógicas, o sea, se debe estudiar cuál es la *semántica* correcta para las operaciones. Otra interrogante es cuáles operaciones tienen sentido y para qué aplicaciones. Cuando fijamos una operación particular y le damos una semántica precisa, surgen otras preguntas. Para el caso de la composición, por ejemplo, podemos preguntarnos si dados mapeos especificados en un formalismo lógico $\mathcal{L}$, ¿puede su composición siempre ser expresada en $\mathcal{L}$? Si la respuesta a esta pregunta es negativa otra interrogante que surge es ¿existe algún formalismo lógico que sea *cerrado* bajo composición? Estas preguntas se pueden aplicar a cualquier otro operador entre mapeos de esquemas que se proponga. Por ejemplo, recientemente se han comenzado a estudiar distintas nociones de *inverso* de mapeos [3, 5, 1]. A pesar de que la interrogante acerca de formalismos cerrados bajo composición ha sido estudiada con cierto detalle [4], no existen trabajos que aborden las mismas preguntas para el caso del inverso de un mapeo.

El primer objetivo de nuestro proyecto es estudiar nociones de inversos de mapeos y la interrogante de si existe algún formalismo lógico que sea cerrado bajo inversión. Un segundo objetivo, relacionado pero mucho mas ambicioso, es desarrollar buenas nociones de inverso y composición que admitan la existencia de algún formalismo lógico que sea simultáneamente cerrado bajo ambas operaciones.

1.1 Notación

Un mapeo $\mathcal{M}$ entre dos esquemas $\mathbf{S}$ y $\mathbf{T}$, es simplemente un conjunto de pares de instancias (I, J) donde I es una instancia en $\mathbf{S}$ y J es una instancia en $\mathbf{T}$. Dado un mapeo $\mathcal{M}$ de $\mathbf{S}$ a $\mathbf{T}$ y un mapeo $\mathcal{M}'$ de $\mathbf{T}$ a $\mathbf{R}$, la *composición* $\mathcal{M} \circ \mathcal{M}'$ es el conjunto de instancias (I, K) donde I es una instancia en $\mathbf{S}$ y K es una instancia en $\mathbf{R}$, tales que existe una instancia J de $\mathbf{T}$ con $(I, J) \in \mathcal{M}$ y $(J, K) \in \mathcal{M}'$.

Dado un conjunto de fórmulas Σ en alguna lógica, decimos que Σ *especifica* a $\mathcal{M}$ si los pares de instancias en $\mathcal{M}$ son exactamente

aquellos que satisfacen Σ. Nos concentraremos en un tipo especial de fórmulas para especificar mapeos. Dados dos lenguajes de consulta $\mathcal{L}_1$ y $\mathcal{L}_2$, una dependencia $\mathcal{L}_1 \rightarrow \mathcal{L}_2$ de un esquema $\mathbf{S}$ a un esquema $\mathbf{T}$, es una fórmula lógica $\forall \bar{x}(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$ donde $\varphi(\bar{x})$ es una consulta en $\mathcal{L}_1$ sobre $\mathbf{S}$, y $\varphi(\bar{x})$ es una consulta en $\mathcal{L}_2$ sobre $\mathbf{T}$. A $\varphi(\bar{x})$ se le llama *antecedente* y a $\psi(\bar{x})$ *consecuente*. Las dependencias más usadas como mapeos de esquemas son las especificadas por dependencias CQ→CQ donde CQ es la clase de las consultas conjuntivas. Por ejemplo las fórmulas (2) y (3) son ejemplos de dependencias CQ→CQ. La fórmula (1) es una dependencia $\text{CQ}^{\neq}$→CQ donde $\text{CQ}^{\neq}$ es la clase de consultas conjuntivas con desigualdades.

Dado un mapeo $\mathcal{M}$ entre $\mathbf{S}$ y $\mathbf{T}$, una consulta Q sobre $\mathbf{T}$, y una instancia I de $\mathbf{S}$, se define el conjunto de las *respuestas certeras* de Q sobre I con respecto a $\mathcal{M}$, denotado por $\text{certain}_{\mathcal{M}}(Q, I)$ como

$$\text{certain}_{\mathcal{M}}(Q, I) = \bigcap_{(I,J)\in\mathcal{M}} Q(J).$$

2. NOCIONES DE INVERSA

Desde el punto de vista más abstracto, dado un mapeo $\mathcal{M}$ de $\mathbf{S}$ a $\mathbf{T}$, un *inverso* de $\mathcal{M}$ debiera ser un mapeo $\mathcal{M}'$ de $\mathbf{T}$ a $\mathbf{S}$ que *revierta* la aplicación de $\mathcal{M}$. Por ejemplo, si $\mathcal{M}$ es usado para intercambiar información, la aplicación de $\mathcal{M}'$ debiera, bajo algún criterio, recuperar la información intercambiada.

En [3], Fagin propone la primera noción de inversa para mapeos de esquema, centrando su estudio en mapeos especificados por dependencias CQ→CQ. La noción de inversa propuesta en [3] resulta ser muy restrictiva; la mayor parte de los mapeos CQ→CQ no poseen inversa. En vista de esta limitación se han propuesto nociones que relajan a la inversa de Fagin. En [5], Fagin et al. proponen la noción de *quasi-inversa*. A pesar de que la noción de quasi-inversa relaja estrictamente a la noción de inversa, aún existen mapeos muy simples especificados por dependencias CQ→CQ que no poseen quasi-inversa. En [1], Arenas et al. proponen la noción de *recovery-máximo* de un mapeo. Una característica importante de la noción de recovery-máximo, es que todos los mapeos CQ→CQ poseen un recovery-máximo.

Hemos mencionado que las dependencias CQ→CQ son las más usadas en la práctica. Por esto sería ideal que el resultado de operar mapeos dados por dependencias CQ→CQ fuera expresable en el mismo lenguaje, o en una extensión minimal de este lenguaje que mantuviera sus buenas propiedades. Un punto en contra de las nociones de quasi-inversa y recovery-máximo, es que para ser especificadas ambas necesitan agregar disyunciones a los consecuentes de las dependencias. Manejar dependencias con disyunciones en los consecuentes es impráctico y computacionalmente complejo. Por su parte la noción de inversa, a pesar de necesitar un lenguaje mucho más manejable para ser especificada (sólo se necesita agregar desigualdades a los antecedentes), es demasiado restrictiva lo que la hace poco aplicable en la práctica. Esto motiva el estudio de nuevas nociones que aminoren las desventajas mencionadas y cuenten con propiedades de clausura.

2.1 Inversas basadas en respuestas certeras

En este trabajo proponemos nuevas nociones de inversas basadas en respuestas certeras. La intuición de nuestras definiciones es la siguiente. Considere una consulta Q una instancia I y un mapeo $\mathcal{M}$. Buscamos un mapeo $\mathcal{M}'$ tal que, si se intercambia la información de I usando $\mathcal{M}$ y se retorna usando $\mathcal{M}'$, el resultado de consultar Q sea lo más cercano posible al resultado de consultar Q directamente sobre la instancia I inicial. El ideal es cuando estos resultados son iguales.

DEFINICIÓN 2.1. *Sea $\mathcal{C}$ una clase de consultas y $\mathcal{M}$ un mapeo. $\mathcal{M}'$ es una $\mathcal{C}$-inversa de $\mathcal{M}$ si* $\text{certain}_{\mathcal{M}\circ\mathcal{M}'}(Q, I) = Q(I)$ *para toda I y toda $Q \in \mathcal{C}$.*

Cuando este ideal no es posible, nos interesa que al menos la información recurperada sea sana, o sea, que no se recupere más información que la inicial. Adicionalmente, queremos recuperar la mayor cantidad de información sana que sea posible. Estas nociones se formalizan en la siguiente definición

DEFINICIÓN 2.2. *Sea $\mathcal{C}$ una clase de consultas y $\mathcal{M}$ un mapeo.*

1. *$\mathcal{M}'$ es un $\mathcal{C}$-recovery de $\mathcal{M}$ si para toda instancia I y consulta $Q \in \mathcal{C}$ se tiene que* $\text{certain}_{\mathcal{M}\circ\mathcal{M}'}(Q, I) \subseteq Q(I)$.
2. *$\mathcal{M}'$ es un $\mathcal{C}$-recovery-máximo de $\mathcal{M}$ si $\mathcal{M}'$ es un $\mathcal{C}$-recovery de $\mathcal{M}$ y para todo otro $\mathcal{M}''$ $\mathcal{C}$-recovery de $\mathcal{M}$ se tiene que* $\text{certain}_{\mathcal{M}\circ\mathcal{M}''}(Q, I) \subseteq \text{certain}_{\mathcal{M}\circ\mathcal{M}'}(Q, I)$ *para toda instancia I y consulta $Q \in \mathcal{C}$.*

3. RESULTADOS PRELIMINARES

En lo que sigue centraremos nuestro estudio en el caso en que se usa CQ como la clase de consultas para invertir. La primera pregunta que surge considerando la noción de CQ-inversa es qué tan restrictiva es. El siguiente resultado muestra que la noción de CQ-inversa es tan restrictiva como la noción de inversa introducida por Fagin [3].

TEOREMA 3.1. *Sea $\mathcal{M}$ un mapeo especificado por dependencias* CQ→CQ*. $\mathcal{M}$ tiene una* CQ*-inversa si y sólo si $\mathcal{M}$ tiene una inversa de Fagin [3].*

Afortunadamente, la noción de CQ-recovery-máximo es una relajación estricta de la noción de CQ-inversa.

TEOREMA 3.2. *Todo mapeo especificado por dependencias* CQ→CQ *tiene un* CQ*-recovery-máximo.*

3.1 Trabajo en curso

Nuestro objetivo principal es poder estudiar propiedades de clausura para esta nueva noción de inverso. Desafortunadamente, nuestro primer resultado al respecto es negativo; CQ→CQ no es un lenguaje cerrado bajo esta nueva noción.

TEOREMA 3.3. *Existe un mapeo especificado por dependencias* CQ→CQ *que no tiene un* CQ*-recovery-máximo especificado por dependencias* CQ→CQ*.*

Nuestra investigación hasta el momento nos permite conjeturar que existe una clase de dependencias $\mathcal{L}$→CQ que es suficiente para especificar el CQ-recovery-máximo de mapeos dados por dependencias CQ→CQ. Es decir, a pesar de que en el antecedente se puede necesitar más que CQ, no se necesita más que CQ en el consecuente de las dependencias. Esta sería una diferencia radical con las nociones de quasi-inversa [5] y recovery-máximo [1] ya que ambas necesitan disyunción. El siguiente paso en nuestra investigación sería demostrar que este nuevo lenguaje $\mathcal{L}$→CQ es *cerrado* bajo CQ-recovery-máximo.

Finalizamos este resumen planteando la conjetura que hemos adoptado como hipótesis de nuestra investigación en curso.

CONJETURA 3.4. *Existe un lenguaje $\mathcal{L}$ (que es una extensión* razonable *de* CQ*), tal que todo mapeo especificado por dependencias $\mathcal{L}$→CQ tiene un* CQ*-recovery-máximo especificado por dependencias $\mathcal{L}$→CQ.*

4. REFERENCIAS

[1] M. Arenas, J. Pérez, and C. Riveros. The recovery of a schema mapping: Bringing exchanged data back. In *PODS*, pages 13–22, 2008.

[2] P. Bernstein. Applying model management to classical meta data problems. In *CIDR*, 2003.

[3] R. Fagin. Inverting schema mappings. *TODS*, 32(4), 2007.

[4] R. Fagin, P. G. Kolaitis, L. Popa, and W.-C. Tan. Composing schema mappings: Second-order dependencies to the rescue. *TODS*, 30(4):994–1055, 2005.

[5] R. Fagin, P. G. Kolaitis, L. Popa, and W.-C. Tan. Quasi-inverses of schema mappings. In *PODS*, pages 123–132, 2007.

[6] A. Nash, P. A. Bernstein, S. Melnik. Composition of mappings given by embedded dependencies. In *PODS*, pages 172–183, 2005.