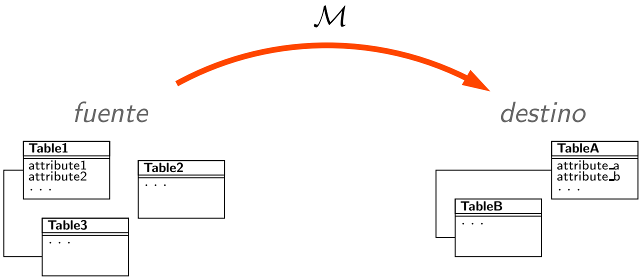
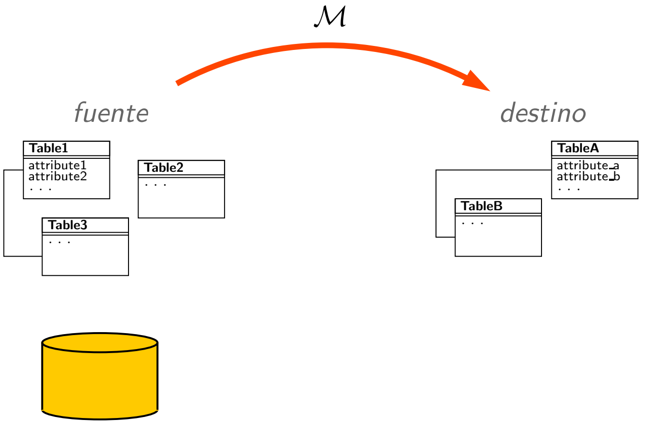
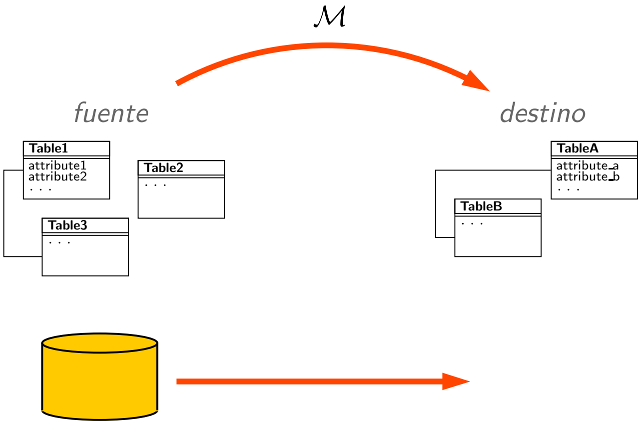
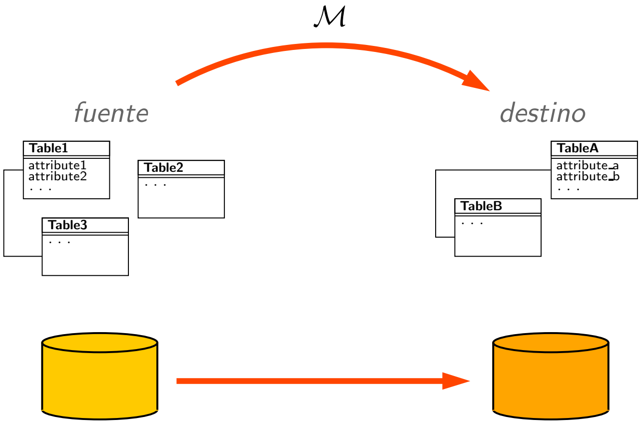
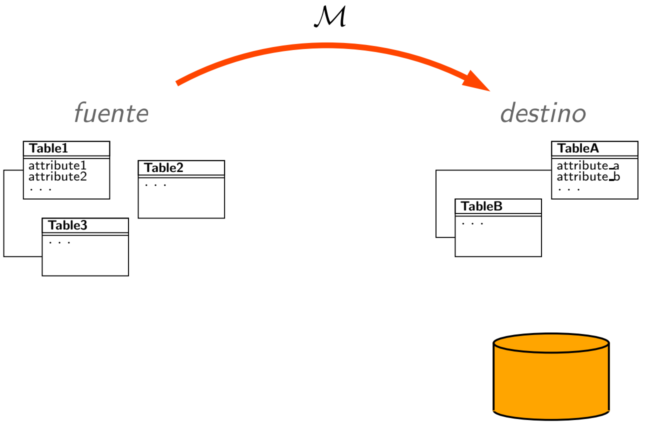

$\mathcal{M}$
fuente
destino
Table1
attribute1
attribute2
...
Table2
...
Table3
...
TableA
attribute_a
attribute_b
...
TableB
...



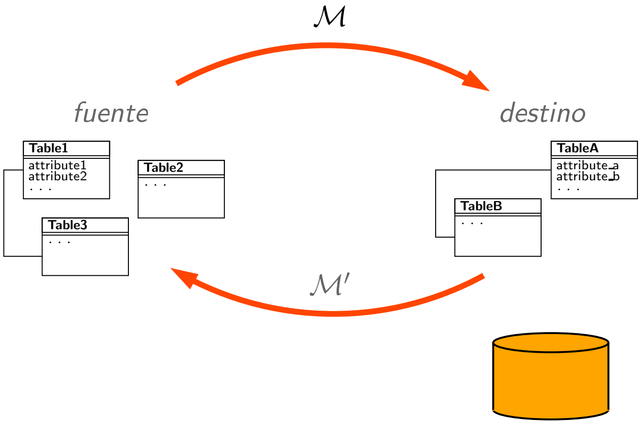
$\mathcal{M}$
fuente
destino
Table1
attribute1
attribute2
Table2
Table3
TableA
attribute_a
attribute_b
TableB


$\mathcal{M}$
fuente
destino
Table1
attribute1
attribute2
. . .
Table2
. . .
Table3
. . .
TableA
attribute_a
attribute_b
. . .
TableB
. . .

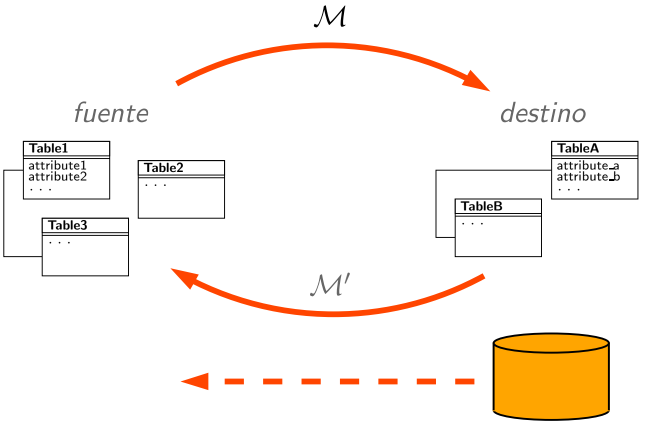

$\mathcal{M}$
fuente
destino
Table1
attribute1
attribute2
. . .
Table2
. . .
Table3
. . .
TableA
attribute_a
attribute_b
. . .
TableB
. . .



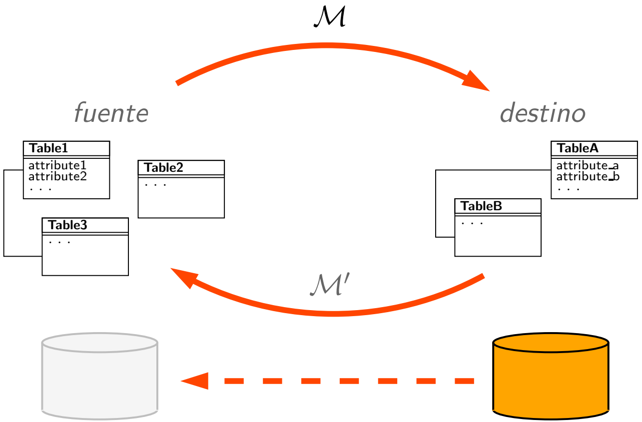
$\mathcal{M}$
fuente
destino
Table1
attribute1
attribute2
...
Table2
...
Table3
...
TableA
attribute_a
attribute_b
...
TableB
...

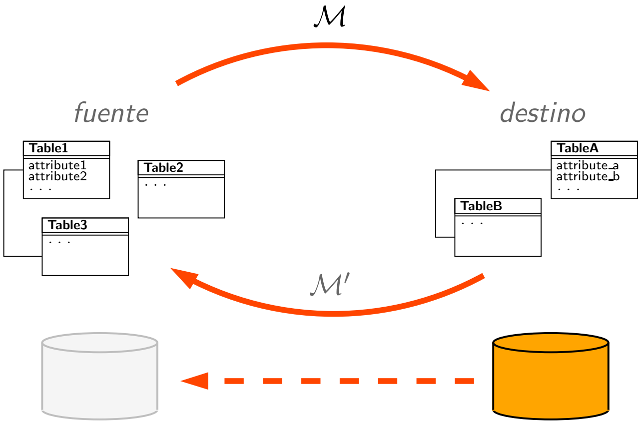


$\mathcal{M}$
fuente
destino
Table1
attribute1
attribute2
. . .
Table2
. . .
Table3
. . .
TableA
attribute_a
attribute_b
. . .
TableB
. . .
$\mathcal{M}'$



$\mathcal{M}$
fuente
destino
Table1
attribute1
attribute2
. . .
Table2
. . .
Table3
. . .
TableA
attribute_a
attribute_b
. . .
TableB
. . .
$\mathcal{M}'$


$\mathcal{M}$
fuente
destino
Table1
attribute1
attribute2
. . .
Table2
. . .
Table3
. . .
TableA
attribute_a
attribute_b
. . .
TableB
. . .
$\mathcal{M}'$



Si $\mathcal{M}$ se especifica en un lenguaje **L**

¿Puede $\mathcal{M}'$ especificarse también en **L**?

Las propuestas actuales no resuelven el problema.

Considerando los lenguajes de mapeo más usados en la práctica:

Las propuestas actuales no resuelven el problema.

Considerando los lenguajes de mapeo más usados en la práctica:

- Inversa (Fagin 06): muy pocos mapeos tienen inversa

Las propuestas actuales no resuelven el problema.

Considerando los lenguajes de mapeo más usados en la práctica:

- Inversa (Fagin 06): muy pocos mapeos tienen inversa
- Quasi-inversa (Fagin et al. 07): no siempre existe, y no admite lenguaje cerrado

Las propuestas actuales no resuelven el problema.

Considerando los lenguajes de mapeo más usados en la práctica:

- Inversa (Fagin 06): muy pocos mapeos tienen inversa
- Quasi-inversa (Fagin et al. 07): no siempre existe, y no admite lenguaje cerrado
- Maximum-Recovery (Arenas, Pérez, Riveros 08): siempre existe, pero no admite lenguaje cerrado

Las propuestas actuales no resuelven el problema.

Considerando los lenguajes de mapeo más usados en la práctica:

- Inversa (Fagin 06): muy pocos mapeos tienen inversa
- Quasi-inversa (Fagin et al. 07): no siempre existe, y no admite lenguaje cerrado
- Maximum-Recovery (Arenas, Pérez, Riveros 08): siempre existe, pero no admite lenguaje cerrado

¿Existe una noción de inverso
que admita un lenguaje **L** cerrado bajo inversión?

Propiedades de Clausura para Operadores de Mapeos

Jorge Pérez

Alumno de Doctorado
Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

trabajo conjunto con Marcelo Arenas, Cristian Riveros, Juan Reutter

Contribuciones

Conceptualmente

- Una nueva noción de inversa basada en *consultas conjuntivas*,

  CQ-Maximum Recovery

  que admite un lenguaje cerrado

Contribuciones

Conceptualmente

- Una nueva noción de inversa basada en *consultas conjuntivas*,

  CQ-Maximum Recovery

  que admite un lenguaje cerrado

Tecnicamente

- Condiciones necesarias y suficiente para existencia
- Algoritmo para computar una inversa
- Demostración de que admite un lenguaje cerrado
- Demostración de que consultas conjuntivas *maximal*

Consultas conjuntivas **CQ** (`SELECT-FROM-WHERE` de SQL)

Ejemplo consulta conjuntiva (**CQ**)

Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).

Consultas conjuntivas **CQ** (`SELECT-FROM-WHERE` de SQL)

Ejemplo consulta conjuntiva (**CQ**)

Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).

¿nombres de alumnos y sus carreras?

Consultas conjuntivas **CQ** (SELECT-FROM-WHERE de SQL)

Ejemplo consulta conjuntiva (**CQ**)

Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).

¿nombres de alumnos y sus carreras?

```
SELECT Alumno.nombre, Carrera.nombre
FROM Alumno, Carrera
WHERE Alumno.carrera_id = Carrera.id
```

Consultas conjuntivas **CQ** (`SELECT-FROM-WHERE` de SQL)

Ejemplo consulta conjuntiva (**CQ**)

Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).

¿nombres de alumnos y sus carreras?

```
SELECT Alumno.nombre, Carrera.nombre
FROM Alumno, Carrera
WHERE Alumno.carrera_id = Carrera.id
```

$$Q(x, y) \quad = \quad \exists U \;\; \text{Alumno}(x, U) \wedge \text{Carrera}(U, y)$$

Consultas conjuntivas **CQ** (`SELECT-FROM-WHERE` de SQL)

Ejemplo: consulta conjuntiva con desigualdades ($\mathbf{CQ}^{\neq}$)

Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).

Consultas conjuntivas **CQ** (`SELECT-FROM-WHERE` de SQL)

Ejemplo: consulta conjuntiva con desigualdades ($\mathbf{CQ}^{\neq}$)

Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).

¿nombres de alumnos que cursan dos carreras y sus carreras?

Consultas conjuntivas **CQ** (`SELECT-FROM-WHERE` de SQL)

Ejemplo: consulta conjuntiva con desigualdades ($\mathbf{CQ}^{\neq}$)

Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).

¿nombres de alumnos que cursan dos carreras y sus carreras?

$$\begin{aligned} Q(x,y,z) \;&=\; \exists U,V\;\; \text{Alumno}(x,U) \wedge \text{Alumno}(x,V) \wedge \\ &\qquad\qquad \text{Carrera}(U,y) \wedge \text{Carrera}(V,z) \wedge y \neq z \end{aligned}$$

Mapeos entre esquemas

Ejemplo

fuente: Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).
destino: Estudiante(**id**,**nombre**,**nom_carrera**), Depto(**id**, **carrera**)

Mapeos entre esquemas

Ejemplo

fuente: Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).
destino: Estudiante(**id**,**nombre**,**nom_carrera**), Depto(**id**, **carrera**)

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow \exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

Mapeos entre esquemas

Ejemplo

fuente: Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).
destino: Estudiante(**id**,**nombre**,**nom_carrera**), Depto(**id**, **carrera**)

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow \exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

En general

Un mapeo **CQ$^{\neq}$-to-CQ** es un conjunto de fórmulas

Mapeos entre esquemas

Ejemplo

fuente: Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).
destino: Estudiante(**id**,**nombre**,**nom_carrera**), Depto(**id**, **carrera**)

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow \exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

En general

Un mapeo **CQ$^{\neq}$-to-CQ** es un conjunto de fórmulas

$$\varphi(x_1, x_2, \ldots, x_n) \rightarrow \psi(x_1, x_2, \ldots, x_n)$$

Mapeos entre esquemas

Ejemplo

fuente: Alumno(**nombre**, **carrera_id**), Carrera(**id**,**nombre**).
destino: Estudiante(**id**,**nombre**,**nom_carrera**), Depto(**id**, **carrera**)

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow \exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

En general

Un mapeo **CQ$^{\neq}$-to-CQ** es un conjunto de fórmulas

$$\varphi(x_1, x_2, \ldots, x_n) \rightarrow \psi(x_1, x_2, \ldots, x_n)$$

- $\varphi(x_1, x_2, \ldots, x_n)$ es una consulta **CQ$^{\neq}$**
- $\psi(x_1, x_2, \ldots, x_n)$ es una consulta **CQ**

Intercambio de datos

Ejemplo

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow$$
$$\exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

Intercambio de datos

Ejemplo

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow$$
$$\exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

Alumno

nombre	carrera_id
Valesca	1
Guillermo	1
Rosa	1
Jose	3

Carrera

id	nombre
1	Computación
2	Química
3	Mecánica

Intercambio de datos

Ejemplo

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow$$
$$\exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

Alumno

nombre	**carrera_id**
Valesca	1
Guillermo	1
Rosa	1
Jose	3

Carrera

id	**nombre**
1	Computación
2	Química
3	Mecánica

$\Downarrow$

Intercambio de datos

Ejemplo

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow$$
$$\exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

Alumno

nombre	carrera_id
Valesca	1
Guillermo	1
Rosa	1
Jose	3

Carrera

id	nombre
1	Computación
2	Química
3	Mecánica

$\Downarrow$

Estudiante

id	nombre	nombre_carrera
I1	Valesca	Computación
I2	Guillermo	Computación
I3	Rosa	Computación
I4	Jose	Mecánica

Depto

id	nombre
D1	Computación
D2	Mecánica

Intercambio de datos

Ejemplo

$$\text{Alumno}(x, y) \wedge \text{Carrera}(y, z) \rightarrow$$
$$\exists I \exists D.\ \text{Estudiante}(I, x, z) \wedge \text{Depto}(D, z).$$

Alumno

nombre	carrera_id
Valesca	1
Guillermo	1
Rosa	1
Jose	3

Carrera

id	nombre
1	Computación
2	Química
3	Mecánica

$\Downarrow$

Estudiante

id	nombre	nombre_carrera
I1	Valesca	Computación
I2	Guillermo	Computación
I3	Rosa	Computación
I4	Jose	Mecánica

Depto

id	nombre
D1	Computación
D2	Mecánica

Solución *canónica*

CQ-maximum recovery

Sean

- $\mathcal{M}$ un mapeo entre un esquema fuente y uno destino
- D_1 una BD en el esquema fuente
- D_2 la BD destino obtenida usando $\mathcal{M}$ desde D_1

CQ-maximum recovery

Sean

- $\mathcal{M}$ un mapeo entre un esquema fuente y uno destino
- D_1 una BD en el esquema fuente
- D_2 la BD destino obtenida usando $\mathcal{M}$ desde D_1

$\mathcal{M}'$ es un **CQ**-maximum recovery de $\mathcal{M}$, si para toda Q en **CQ**

CQ-maximum recovery

Sean

- $\mathcal{M}$ un mapeo entre un esquema fuente y uno destino
- D_1 una BD en el esquema fuente
- D_2 la BD destino obtenida usando $\mathcal{M}$ desde D_1

$\mathcal{M}'$ es un **CQ**-maximum recovery de $\mathcal{M}$, si para toda Q en **CQ**

Si D_3 es la BD obtenida usando $\mathcal{M}'$ desde D_2

$$Q(D_3) \subseteq Q(D_1)$$

CQ-maximum recovery

Sean

- $\mathcal{M}$ un mapeo entre un esquema fuente y uno destino
- D_1 una BD en el esquema fuente
- D_2 la BD destino obtenida usando $\mathcal{M}$ desde D_1

$\mathcal{M}'$ es un **CQ**-maximum recovery de $\mathcal{M}$, si para toda Q en **CQ**

Si D_3 es la BD obtenida usando $\mathcal{M}'$ desde D_2

$$Q(D_3) \subseteq Q(D_1)$$

Para todo otro mapeo $\mathcal{M}''$, si D_4 es obtenida usando $\mathcal{M}''$ desde D_2

$$Q(D_4) \subseteq Q(D_3)$$

CQ-maximum recovery ejemplo

$$
\begin{array}{rcl}
T(x,y) & \rightarrow & R(x,y) \\
A(x) & \rightarrow & R(x,x)
\end{array}
$$

CQ-maximum recovery ejemplo

$$\begin{aligned} T(x,y) &\rightarrow R(x,y) \\ A(x) &\rightarrow R(x,x) \end{aligned}$$

$$R(x,y) \land x \neq y \rightarrow T(x,y)$$

CQ-maximum recovery admite un lenguaje cerrado

Teorema

Todo mapeo expresado en $\mathbf{CQ}^{\neq}$**-to-CQ** tiene un **CQ**-maximum recovery expresado en $\mathbf{CQ}^{\neq}$**-to-CQ**.

CQ-maximum recovery admite un lenguaje cerrado

Teorema

Todo mapeo expresado en $\mathbf{CQ}^{\neq}$**-to-CQ** tiene un **CQ**-maximum recovery expresado en $\mathbf{CQ}^{\neq}$**-to-CQ**.

$\mathbf{CQ}^{\neq}$**-to-CQ** es un lenguaje *cerrado* bajo la nueva noción de inverso.

Otros resultados

Otros resultados

- Algoritmo (de tiempo exponencial) para computar el **CQ**-maximum recovery.

Otros resultados

- Algoritmo (de tiempo exponencial) para computar el **CQ**-maximum recovery.
- Si usas más que **CQ**, no se tiene clausura.

Otros resultados

- Algoritmo (de tiempo exponencial) para computar el **CQ**-maximum recovery.
- Si usas más que **CQ**, no se tiene clausura.
- Resultados se extienden para la composición.

Extensiones, trabajo futuro

- Otras operaciones
- Propiedades de clausura
- Otros modelos (XML, objeto-relacional)

Propiedades de Clausura para Operadores de Mapeos

Jorge Pérez

Alumno de Doctorado
Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

trabajo conjunto con Marcelo Arenas, Cristian Riveros, Juan Reutter