

# Complexity of Answering Queries Using Materialized Views

Serge Abiteboul, Oliver M. Duschka

presentado por Jorge Pérez

# Responder consultas usando vistas precalculadas

Problema:

- ▶ Tenemos vistas precalculadas obtenidas desde una BD
- ▶ Queremos responder consultas a la BD usando sólo las vistas

## Motivación principal

- ▶ Estudiar ampliamente la **complejidad de datos** del problema.
- ▶ Considerar como input la **materialización de las vistas**.

# Responder consultas usando vistas precalculadas

Problema:

- ▶ Tenemos vistas precalculadas obtenidas desde una BD
- ▶ Queremos responder consultas a la BD usando sólo las vistas

## Motivación principal

- ▶ Estudiar ampliamente la **complejidad de datos** del problema.
- ▶ Considerar como input la **materialización de las vistas**.

## Resultados

Cotas para distintas combinaciones de:

- ▶ lenguajes de consulta y de definición de vistas.
- ▶ vistas exactas y vistas sanas

Algo de complejidad: NP, coNP, no recursivos

Conceptos previos

Tablas condicionales

Complejidad usando vistas sanas (OWA)

Complejidad usando vistas exactas (CWA)

Algo de complejidad: NP, coNP, no recursivos

Conceptos previos

Tablas condicionales

Complejidad usando vistas sanas (OWA)

Complejidad usando vistas exactas (CWA)

Algo de complejidad: NP, coNP, no recursivos

Conceptos previos

Tablas condicionales

Complejidad usando vistas sanas (OWA)

Complejidad usando vistas exactas (CWA)

# El problema: definiciones de vistas, instancias, consultas

Componentes del problema:

- ▶ Base de datos **que desconocemos**
- ▶ Definiciones de vistas
- ▶ Vistas materializadas (instancia)
- ▶ Consulta sobre el esquema de la base de datos

## Ejemplo

$D$

$$\mathcal{V}: \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\mathcal{Q}: q(X, Y) \leftarrow p(X, Y)$$

# El problema: definiciones de vistas, instancias, consultas

Componentes del problema:

- ▶ Base de datos **que desconocemos**
- ▶ Definiciones de vistas
- ▶ Vistas materializadas (instancia)
- ▶ Consulta sobre el esquema de la base de datos

## Ejemplo

$D$

$$\mathcal{V}: \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\mathcal{Q}: q(X, Y) \leftarrow p(X, Y)$$

# El problema: definiciones de vistas, instancias, consultas

Componentes del problema:

- ▶ Base de datos **que desconocemos**
- ▶ Definiciones de vistas
- ▶ Vistas materializadas (instancia)
- ▶ Consulta sobre el esquema de la base de datos

## Ejemplo

$D$

$$\mathcal{V}: \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\mathcal{Q}: q(X, Y) \leftarrow p(X, Y)$$

# El problema: definiciones de vistas, instancias, consultas

Componentes del problema:

- ▶ Base de datos **que desconocemos**
- ▶ Definiciones de vistas
- ▶ Vistas materializadas (instancia)
- ▶ Consulta sobre el esquema de la base de datos

## Ejemplo

$D$

$$\mathcal{V}: \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\mathcal{Q}: q(X, Y) \leftarrow p(X, Y)$$

# El problema: definiciones de vistas, instancias, consultas

Componentes del problema:

- ▶ Base de datos **que desconocemos**
- ▶ Definiciones de vistas
- ▶ Vistas materializadas (instancia)
- ▶ Consulta sobre el esquema de la base de datos

Queremos responder la consulta usando las vistas materializadas.

## Ejemplo

$D$

$$\mathcal{V}: \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\mathcal{Q}: q(X, Y) \leftarrow p(X, Y)$$

## Vistas sanas (OWA):

- ▶ las vistas almacenan un **subconjunto** de las tuplas que resultan al evaluarlas sobre la BD

## Vistas exactas (CWA):

- ▶ las vistas almacenan **exactamente** las tuplas que resultan al evaluarlas sobre la BD

Tendrá efectos en el proceso de responder consultas.

Vistas sanas (OWA):

- ▶ las vistas almacenan un **subconjunto** de las tuplas que resultan al evaluarlas sobre la BD

$$I \subseteq \mathcal{V}(D)$$

Vistas exactas (CWA):

- ▶ las vistas almacenan **exactamente** las tuplas que resultan al evaluarlas sobre la BD

Tendrá efectos en el proceso de responder consultas.

Vistas sanas (OWA):

- ▶ las vistas almacenan un **subconjunto** de las tuplas que resultan al evaluarlas sobre la BD

$$I \subseteq \mathcal{V}(D)$$

Vistas exactas (CWA):

- ▶ las vistas almacenan **exactamente** las tuplas que resultan al evaluarlas sobre la BD

$$I = \mathcal{V}(D)$$

Tendrá efectos en el proceso de responder consultas.

# Vistas sanas vs exactas afectan las posibles BDs

Dada una definición de vistas y una materialización de estas existen (en general) muchas BDs *posibles* que generan esa materialización.

## Ejemplo

$$\mathcal{V} : \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$D_1 = \{p(a, b)\}$$

$$D_2 = \{p(a, j), p(i, b), p(k, b)\}$$

$$D_3 = \{r(a, b), s(b, a), p(a, b)\}$$

- ▶  $D_1$ ,  $D_2$  y  $D_3$  son posibles dadas  $\mathcal{V}$  e  $I$  sanas.
- ▶ sólo  $D_1$  y  $D_3$  son posibles dadas  $\mathcal{V}$  e  $I$  exactas.

# Vistas sanas vs exactas afectan las posibles BDs

Dada una definición de vistas y una materialización de estas existen (en general) muchas BDs *posibles* que generan esa materialización.

## Ejemplo

$$\mathcal{V} : \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$D_1 = \{p(a, b)\}$$

$$D_2 = \{p(a, j), p(i, b), p(k, b)\}$$

$$D_3 = \{r(a, b), s(b, a), p(a, b)\}$$

- $D_1$ ,  $D_2$  y  $D_3$  son posibles dadas  $\mathcal{V}$  e  $I$  sanas.
- sólo  $D_1$  y  $D_3$  son posibles dadas  $\mathcal{V}$  e  $I$  exactas.

# Vistas sanas vs exactas afectan las posibles BDs

Dada una definición de vistas y una materialización de estas existen (en general) muchas BDs *posibles* que generan esa materialización.

## Ejemplo

$$\mathcal{V} : \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$D_1 = \{p(a, b)\}$$

$$D_2 = \{p(a, j), p(i, b), p(k, b)\}$$

$$D_3 = \{r(a, b), s(b, a), p(a, b)\}$$

- ▶  $D_1$ ,  $D_2$  y  $D_3$  son posibles dadas  $\mathcal{V}$  e  $I$  sanas.
- ▶ sólo  $D_1$  y  $D_3$  son posibles dadas  $\mathcal{V}$  e  $I$  exactas.

# Respuestas certeras

Respuestas que aparecen en todas las BDs posibles

- ▶ Vimos que para una definición y materialización de vistas existen muchas BDs posibles.
- ▶ Para consultas usamos semántica **de respuestas certeras**

## Definicion

*Dadas definición de vistas, materialización de estas, una consulta, una tupla es **certera** si está en la evaluación de la consulta para cada BD posible.*

# Respuestas certeras

Respuestas que aparecen en todas las BDs posibles

- ▶ Vimos que para una definición y materialización de vistas existen muchas BDs posibles.
- ▶ Para consultas usamos semántica **de respuestas certeras**

## Definición

*Dadas definición de vistas, materialización de estas, una consulta, una tupla es **certera** si está en la evaluación de la consulta para cada BD posible.*

$t \in Q(D)$  para toda posible  $D$  dadas  $\mathcal{V}, I$

# Vistas sanas vs exactas afectan las respuestas.

## Ejemplo

$$\mathcal{V} : \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\begin{array}{l} Q_1 : q_1(X) \leftarrow p(X, Y) \\ Q_2 : q_2(X, Y) \leftarrow p(X, Y) \end{array}$$

Respuestas certeras:

- ▶ vistas sanas
- ▶ vistas exactas

# Vistas sanas vs exactas afectan las respuestas.

## Ejemplo

$$\mathcal{V} : \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\begin{array}{l} Q_1 : q_1(X) \leftarrow p(X, Y) \\ Q_2 : q_2(X, Y) \leftarrow p(X, Y) \end{array}$$

Respuestas certeras:

- ▶ vistas sanas
  - ▶  $Q_1 : \{q_1(a)\}$
  - ▶  $Q_2 : \{\}$
- ▶ vistas exactas

# Vistas sanas vs exactas afectan las respuestas.

## Ejemplo

$$\mathcal{V} : \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\begin{array}{l} Q_1 : q_1(X) \leftarrow p(X, Y) \\ Q_2 : q_2(X, Y) \leftarrow p(X, Y) \end{array}$$

Respuestas certeras:

- ▶ vistas sanas
  - ▶  $Q_1 : \{q_1(a)\}$
  - ▶  $Q_2 : \{\}$
- ▶ vistas exactas
  - ▶  $Q_1 : \{q_1(a)\}$
  - ▶  $Q_2 : \{q_2(a, b)\}$

# Estamos interesados en la complejidad de los datos.

Complejidad de los datos:

- ▶ El problema es determinar si una tupla dada es certera:  
*problema de evaluación certera*.
- ▶ Medimos complejidad sólo en función del tamaño de la materialización de las vistas.

## Ejemplo

$$\mathcal{V}: \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\mathcal{Q}: q(X, Y) \leftarrow p(X, Y)$$

¿es  $q(a, b)$  certera?

# Estamos interesados en la complejidad de los datos.

Complejidad de los datos:

- ▶ El problema es determinar si una tupla dada es certera:  
*problema de evaluación certera*.
- ▶ Medimos complejidad sólo en función del tamaño de la materialización de las vistas.

## Ejemplo

$$\mathcal{V}: \begin{array}{l} v_1(X) \leftarrow p(X, Y) \\ v_2(Y) \leftarrow p(X, Y) \end{array} \quad I = \{v_1(a), v_2(b)\}$$

$$\mathcal{Q}: q(X, Y) \leftarrow p(X, Y)$$

¿es  $q(a, b)$  certera?

Complejidad solo en función de  $I$ , todo lo demás se supone fijo.

Algo de complejidad: NP, coNP, no recursivos

Conceptos previos

Tablas condicionales

Complejidad usando vistas sanas (OWA)

Complejidad usando vistas exactas (CWA)

# El problema es coNP para una amplia gama de lenguajes

## Teorema

*Para vistas definidas en consultas-positivas con  $\neq$ , y consultas en DATALOG con  $\neq$*

*el problema de evaluación certera es coNP*

*para vistas sanas y exactas.*

# El problema es coNP para una amplia gama de lenguajes

## Teorema

*Para vistas definidas en consultas-positivas con  $\neq$ , y consultas en DATALOG con  $\neq$*

*el problema de evaluación certera es coNP*

*para vistas sanas y exactas.*

## Demostracion (Idea)

*Dada una tupla  $t$  que no es certera:*

- ▶ *El certificado es una base de datos  $D$ .*
- ▶  *$D$  se puede escoger de tamaño polinomial dados los lenguajes usados para vistas y consulta.*
- ▶ *En tiempo polinomial se chequea que  $D$  es BD posible y que  $t$  no está en la evaluación de la consulta sobre  $D$ .*

# El problema es coNP para una amplia gama de lenguajes

## Teorema

*Para vistas definidas en consultas-positivas con  $\neq$ , y consultas en DATALOG con  $\neq$*

*el problema de evaluación certera es coNP*

*para vistas sanas y exactas.*

## Demostracion (Idea)

*Dada una tupla  $t$  que no es certera:*

- ▶ *El certificado es una base de datos  $D$ .*
- ▶  *$D$  se puede escoger de tamaño polinomial dados los lenguajes usados para vistas y consulta.*
- ▶ *En tiempo polinomial se chequea que  $D$  es BD posible y que  $t$  no está en la evaluación de la consulta sobre  $D$ .*

# El problema es coNP para una amplia gama de lenguajes

## Teorema

*Para vistas definidas en consultas-positivas con  $\neq$ , y consultas en DATALOG con  $\neq$*

*el problema de evaluación certera es coNP*

*para vistas sanas y exactas.*

## Demostracion (Idea)

*Dada una tupla  $t$  que no es certera:*

- ▶ *El certificado es una base de datos  $D$ .*
- ▶  *$D$  se puede escoger de tamaño polinomial dados los lenguajes usados para vistas y consulta.*
- ▶ *En tiempo polinomial se chequea que  $D$  es BD posible y que  $t$  no está en la evaluación de la consulta sobre  $D$ .*

# El problema es coNP para una amplia gama de lenguajes

## Teorema

*Para vistas definidas en consultas-positivas con  $\neq$ , y consultas en DATALOG con  $\neq$*

*el problema de evaluación certera es coNP*

*para vistas sanas y exactas.*

## Demostracion (Idea)

*Dada una tupla  $t$  que no es certera:*

- ▶ *El certificado es una base de datos  $D$ .*
- ▶  *$D$  se puede escoger de tamaño polinomial dados los lenguajes usados para vistas y consulta.*
- ▶ *En tiempo polinomial se chequea que  $D$  es BD posible y que  $t$  no está en la evaluación de la consulta sobre  $D$ .*

# Tablas condicionales pueden representar las BDs posibles.

- ▶ Una *Tabla Condicional* (TC) es una tabla con variables y condiciones en las tuplas
- ▶ Una BD posible se obtiene dando valores a las variables y manteniendo las tuplas cuyas condiciones se hacen verdaderas

## Ejemplo

| $T$ | $A$ | $B$ |         |
|-----|-----|-----|---------|
|     | 0   | 1   |         |
|     | 0   | x   | $y = x$ |
|     | 1   | y   | $y = 2$ |

# Tablas condicionales pueden representar las BDs posibles.

- ▶ Una *Tabla Condicional* (TC) es una tabla con variables y condiciones en las tuplas
- ▶ Una BD posible se obtiene dando valores a las variables y manteniendo las tuplas cuyas condiciones se hacen verdaderas

## Ejemplo

| $T$ | $A$ | $B$ |         |
|-----|-----|-----|---------|
|     | 0   | 1   |         |
|     | 0   | x   | $y = x$ |
|     | 1   | y   | $y = 2$ |

$x \rightarrow 1$

$y \rightarrow 2$

# Tablas condicionales pueden representar las BDs posibles.

- ▶ Una *Tabla Condicional* (TC) es una tabla con variables y condiciones en las tuplas
- ▶ Una BD posible se obtiene dando valores a las variables y manteniendo las tuplas cuyas condiciones se hacen verdaderas

## Ejemplo

| $T$ | $A$ | $B$         |
|-----|-----|-------------|
|     | 0   | 1           |
|     | 0   | $x$ $y = x$ |
|     | 1   | $y$ $y = 2$ |

|                   |      |     |     |
|-------------------|------|-----|-----|
| $x \rightarrow 1$ | $T'$ | $A$ | $B$ |
| $y \rightarrow 2$ |      | 0   | 1   |
|                   |      | 1   | 2   |

# Tablas condicionales pueden representar las BDs posibles.

- ▶ Una *Tabla Condicional* (TC) es una tabla con variables y condiciones en las tuplas
- ▶ Una BD posible se obtiene dando valores a las variables y manteniendo las tuplas cuyas condiciones se hacen verdaderas

## Ejemplo

| $T$ | $A$ | $B$ |
|-----|-----|-----|
|     | 0   | 1   |
|     | 0   | $x$ |
|     | 1   | $y$ |

|                   |      |     |     |                   |
|-------------------|------|-----|-----|-------------------|
| $x \rightarrow 1$ | $T'$ | $A$ | $B$ | $x \rightarrow 2$ |
| $y \rightarrow 2$ |      | 0   | 1   | $y \rightarrow 2$ |
|                   |      | 1   | 2   |                   |

# Tablas condicionales pueden representar las BDs posibles.

- ▶ Una *Tabla Condicional* (TC) es una tabla con variables y condiciones en las tuplas
- ▶ Una BD posible se obtiene dando valores a las variables y manteniendo las tuplas cuyas condiciones se hacen verdaderas

## Ejemplo

| $T$ | $A$ | $B$         |
|-----|-----|-------------|
|     | 0   | 1           |
|     | 0   | $x$ $y = x$ |
|     | 1   | $y$ $y = 2$ |

$x \rightarrow 1$   
 $y \rightarrow 2$

| $T'$ | $A$ | $B$ |
|------|-----|-----|
|      | 0   | 1   |
|      | 1   | 2   |

$x \rightarrow 2$   
 $y \rightarrow 2$

| $T''$ | $A$ | $B$ |
|-------|-----|-----|
|       | 0   | 1   |
|       | 0   | 2   |
|       | 1   | 2   |

## Theorem

*Para consultas-positivas con  $\neq$  como vistas, es posible construir una TC tal que los siguientes conjuntos coinciden:*

- ▶ tuplas ciertas de las BDs posibles de TC*
- ▶ tuplas ciertas dadas las materializaciones de las vistas*

*para cada posible consulta en DATALOG con  $\neq$ .*

## Theorem

*Para consultas-positivas con  $\neq$  como vistas, es posible construir una TC tal que los siguientes conjuntos coinciden:*

- ▶ *tuplas certeras de las BDs posibles de TC*
- ▶ *tuplas certeras dadas las materializaciones de las vistas*

*para cada posible consulta en DATALOG con  $\neq$ .*

## Theorem

*Para consultas-positivas con  $\neq$  como vistas, es posible construir una TC tal que los siguientes conjuntos coinciden:*

- ▶ *tuplas certeras de las BDs posibles de TC*
- ▶ *tuplas certeras dadas las materializaciones de las vistas*

*para cada posible consulta en DATALOG con  $\neq$ .*

## Theorem

*Para consultas-positivas con  $\neq$  como vistas, es posible construir una TC tal que los siguientes conjuntos coinciden:*

- ▶ *tuplas certeras de las BDs posibles de TC*
- ▶ *tuplas certeras dadas las materializaciones de las vistas*

*para cada posible consulta en DATALOG con  $\neq$ .*

Obtenemos un algoritmo para calcular las respuestas certeras:

- ▶ construya la TC correspondiente,
- ▶ consúltela usando técnicas standard de consulta sobre TCs  
**[AHV95, IL84].**

## Ejemplo

$$\mathcal{V}: \begin{array}{l} v(0, Y) \leftarrow p(0, Y) \\ v(X, Y) \leftarrow p(X, Z), p(Z, Y) \end{array} \quad I = \{v(0, 1), v(1, 1)\}$$

TC

## Ejemplo

$$\mathcal{V}: \begin{array}{l} v(0, Y) \leftarrow p(0, Y) \\ v(X, Y) \leftarrow p(X, Z), p(Z, Y) \end{array} \quad I = \{v(0, 1), v(1, 1)\}$$

TC

0 1

## Ejemplo

$$\mathcal{V}: \begin{array}{l} v(0, Y) \leftarrow p(0, Y) \\ v(X, Y) \leftarrow p(X, Z), p(Z, Y) \end{array} \quad I = \{v(0, 1), v(1, 1)\}$$

TC

|   |   |
|---|---|
| 0 | 1 |
| 0 | z |
| z | 1 |

# Tablas condicionales nos dan un algoritmos

## Ejemplo

$$\mathcal{V}: \begin{array}{l} v(0, Y) \leftarrow p(0, Y) \\ v(X, Y) \leftarrow p(X, Z), p(Z, Y) \end{array} \quad I = \{v(0, 1), v(1, 1)\}$$

TC

|   |   |
|---|---|
| 0 | 1 |
| 0 | z |
| z | 1 |
| 1 | u |
| u | 1 |

## Ejemplo

$$\mathcal{V}: \begin{array}{l} v(0, Y) \leftarrow p(0, Y) \\ v(X, Y) \leftarrow p(X, Z), p(Z, Y) \end{array} \quad I = \{v(0, 1), v(1, 1)\}$$

TC

|     |     |            |
|-----|-----|------------|
| 0   | 1   | $w = 1$    |
| 0   | $z$ | $w \neq 1$ |
| $z$ | 1   | $w \neq 1$ |
| 1   | $u$ |            |
| $u$ | 1   |            |

Algo de complejidad: NP, coNP, no recursivos

Conceptos previos

Tablas condicionales

Complejidad usando vistas sanas (OWA)

Complejidad usando vistas exactas (CWA)

## Teorema

*Para vistas conjuntivas y consultas DATALOG*

*el problema de evaluación certera es polinomial.*

- ▶ Demostración usa **planes de consulta maximales** (PCM).
- ▶ Un PCM puede ser obtenido efectivamente **[DG97]** y calcula exactamente las respuestas certeras.
- ▶ Todo PCM es un programa DATALOG  $\Rightarrow$  tiene complejidad de datos polinomial.

# Polinomial para vistas conjuntivas con $\neq$ y consultas DATALOG

## Teorema

*Para vistas conjuntivas con  $\neq$  y consultas DATALOG*

*el problema de evaluación certera es polinomial.*

# Polinomial para vistas conjuntivas con $\neq$ y consultas DATALOG

## Teorema

*Para vistas conjuntivas con  $\neq$  y consultas DATALOG*

*el problema de evaluación certera es polinomial.*

## Demostracion (Idea)

*Considere las vistas sin  $\neq$ , las respuestas certeras se mantienen:*

- ▶ *Toda BD posible para las vistas originales es también posible para las vistas sin  $\neq$  (vistas sanas y  $\neq$  solo puede filtrar).*
- ▶ *Al contrario no es siempre cierto, pero*

# Polinomial para vistas conjuntivas con $\neq$ y consultas DATALOG

## Teorema

*Para vistas conjuntivas con  $\neq$  y consultas DATALOG*

*el problema de evaluación certera es polinomial.*

## Demostracion (Idea)

*Considere las vistas sin  $\neq$ , las respuestas certeras se mantienen:*

- ▶ *Toda BD posible para las vistas originales es también posible para las vistas sin  $\neq$  (vistas sanas y  $\neq$  solo puede filtrar).*
- ▶ *Al contrario no es siempre cierto, pero*

# Polinomial para vistas conjuntivas con $\neq$ y consultas DATALOG

## Teorema

*Para vistas conjuntivas con  $\neq$  y consultas DATALOG*

*el problema de evaluación certera es polinomial.*

## Demostracion (Idea)

*Considere las vistas sin  $\neq$ , las respuestas certeras se mantienen:*

- ▶ *Toda BD posible para las vistas originales es también posible para las vistas sin  $\neq$  (vistas sanas y  $\neq$  solo puede filtrar).*
- ▶ *Al contrario no es siempre cierto, pero*

# Polinomial para vistas conjuntivas con $\neq$ y consultas DATALOG

## Teorema

*Para vistas conjuntivas con  $\neq$  y consultas DATALOG*

*el problema de evaluación certera es polinomial.*

## Demostracion (Idea)

*Considere las vistas sin  $\neq$ , las respuestas certeras se mantienen:*

- ▶ *Toda BD posible para las vistas originales es también posible para las vistas sin  $\neq$  (vistas sanas y  $\neq$  solo puede filtrar).*
- ▶ *Al contrario no es siempre cierto, pero*
  - ▶ *dada una BD posible  $D$  para las vistas sin  $\neq$ , se construye  $D'$  posible para las vistas originales (agregando tuplas)*

# Polinomial para vistas conjuntivas con $\neq$ y consultas DATALOG

## Teorema

*Para vistas conjuntivas con  $\neq$  y consultas DATALOG*

*el problema de evaluación certera es polinomial.*

## Demostracion (Idea)

*Considere las vistas sin  $\neq$ , las respuestas certeras se mantienen:*

- ▶ *Toda BD posible para las vistas originales es también posible para las vistas sin  $\neq$  (vistas sanas y  $\neq$  solo puede filtrar).*
- ▶ *Al contrario no es siempre cierto, pero*
  - ▶ *dada una BD posible  $D$  para las vistas sin  $\neq$ , se construye  $D'$  posible para las vistas originales (agregando tuplas)*
  - ▶ *si  $t$  es repuesta en  $D'$ ,  $t$  es respuesta en  $D$ .*

## Ejemplo

(Idea de la demostración anterior)

$\mathcal{V} : v(X, Y) \leftarrow p(X, Z), p(Z, Y), Z \neq Y$

$I : \{v(1, 2), v(2, 2)\}$

$D = \{p(1, 2), p(2, 2)\}$  es posible para  $\mathcal{V}$  sin  $\neq$

$D' = \{p(1, 2), p(1, 3), p(3, 2), p(2, 3), p(2, 2)\}$  es posible para  $\mathcal{V}$

$\mathcal{Q} : q(X) \leftarrow p(X, Y)$

Para toda  $t$  certera, si  $t$  puede ser obtenida en  $D'$  entonces  $t$  puede ser obtenida en  $D$ .

## Ejemplo

(Idea de la demostración anterior)

$\mathcal{V} : v(X, Y) \leftarrow p(X, Z), p(Z, Y), Z \neq Y$

$I : \{v(1, 2), v(2, 2)\}$

$D = \{p(1, 2), p(2, 2)\}$  es posible para  $\mathcal{V}$  sin  $\neq$

$D' = \{p(1, 2), p(1, 3), p(3, 2), p(2, 3), p(2, 2)\}$  es posible para  $\mathcal{V}$

$\mathcal{Q} : q(X) \leftarrow p(X, Y)$

Para toda  $t$  certera, si  $t$  puede ser obtenida en  $D'$  entonces  $t$  puede ser obtenida en  $D$ .

## Ejemplo

(Idea de la demostración anterior)

$\mathcal{V} : v(X, Y) \leftarrow p(X, Z), p(Z, Y), Z \neq Y$

$I : \{v(1, 2), v(2, 2)\}$

$D = \{p(1, 2), p(2, 2)\}$  es posible para  $\mathcal{V}$  sin  $\neq$

$D' = \{p(1, 2), p(1, 3), p(3, 2), p(2, 3), p(2, 2)\}$  es posible para  $\mathcal{V}$

$\mathcal{Q} : q(X) \leftarrow p(X, Y)$

Para toda  $t$  certera, si  $t$  puede ser obtenida en  $D'$  entonces  $t$  puede ser obtenida en  $D$ .

## Ejemplo

(Idea de la demostración anterior)

$\mathcal{V} : v(X, Y) \leftarrow p(X, Z), p(Z, Y), Z \neq Y$

$I : \{v(1, 2), v(2, 2)\}$

$D = \{p(1, 2), p(2, 2)\}$  es posible para  $\mathcal{V}$  sin  $\neq$

$D' = \{p(1, 2), p(1, 3), p(3, 2), p(2, 3), p(2, 2)\}$  es posible para  $\mathcal{V}$

$\mathcal{Q} : q(X) \leftarrow p(X, Y)$

Para toda  $t$  certera, si  $t$  puede ser obtenida en  $D'$  entonces  $t$  puede ser obtenida en  $D$ .

## Ejemplo

(Idea de la demostración anterior)

$\mathcal{V} : v(X, Y) \leftarrow p(X, Z), p(Z, Y), Z \neq Y$

$I : \{v(1, 2), v(2, 2)\}$

$D = \{p(1, 2), p(2, 2)\}$  es posible para  $\mathcal{V}$  sin  $\neq$

$D' = \{p(1, 2), p(1, 3), p(3, 2), p(2, 3), p(2, 2)\}$  es posible para  $\mathcal{V}$

$\mathcal{Q} : q(X) \leftarrow p(X, Y)$

Para toda  $t$  certera, si  $t$  puede ser obtenida en  $D'$  entonces  $t$  puede ser obtenida en  $D$ .

## Teorema

*Para vistas conjuntivas, y consultas conjuntivas con  $\neq$   
el problema de evaluación certera es coNP-hard.*

La demostración usa una reducción desde satisfactibilidad de fórmulas en CNF:

Fórmula

$$\varphi = c_1 \wedge \cdots \wedge c_m$$

con  $c_i$  una disyunción de literales

¿existe una valuación que haga verdadera a  $\varphi$ ?

# coNP-hard para consultas con $\neq$ , demostración (¿incorrecta?)

## Demostracion (Idea)

Dada  $\varphi = c_1 \wedge \dots \wedge c_m$  con variables  $\{x_1, \dots, x_n\}$  se construye

*vistas*

$$v_1(X, Y, Z) \leftarrow p(X, Y, Z)$$

$$v_2(X) \leftarrow r(X, Y)$$

$$v_3(Y) \leftarrow p(X, Y, Z), r(X, Z)$$

*instancias*

$$\{v_1(i, j, 1) \mid \text{si } x_i \text{ ocurre en } c_j\} \cup$$

$$\{v_1(i, j, 0) \mid \text{si } \neg x_i \text{ ocurre en } c_j\}$$

$$\{v_2(1), v_2(2), \dots, v_2(n)\}$$

$$\{v_3(1), v_3(2), \dots, v_3(m)\}$$

*Idea: BD posibles representan asignaciones que satisfacen  $\varphi$*

# coNP-hard para consultas con $\neq$ , demostración (¿incorrecta?)

## Demostracion (Idea)

Dada  $\varphi = c_1 \wedge \dots \wedge c_m$  con variables  $\{x_1, \dots, x_n\}$  se construye

### vistas

$$v_1(X, Y, Z) \leftarrow p(X, Y, Z)$$

$$v_2(X) \leftarrow r(X, Y)$$

$$v_3(Y) \leftarrow p(X, Y, Z), r(X, Z)$$

### instancias

$$\{v_1(i, j, 1) \mid \text{si } x_i \text{ ocurre en } c_j\} \cup$$

$$\{v_1(i, j, 0) \mid \text{si } \neg x_i \text{ ocurre en } c_j\}$$

$$\{v_2(1), v_2(2), \dots, v_2(n)\}$$

$$\{v_3(1), v_3(2), \dots, v_3(m)\}$$

Idea: BD posibles representan asignaciones que satisfacen  $\varphi$

- ▶  $r(X, Y)$ : a la variable  $X$  le asigno valor de verdad  $Y$
- ▶ materialización de  $v_2$  chequea que sea asignación completa.

# coNP-hard para consultas con $\neq$ , demostración (¿incorrecta?)

## Demostracion (Idea)

Dada  $\varphi = c_1 \wedge \dots \wedge c_m$  con variables  $\{x_1, \dots, x_n\}$  se construye

### vistas

$$v_1(X, Y, Z) \leftarrow p(X, Y, Z)$$

$$v_2(X) \leftarrow r(X, Y)$$

$$v_3(Y) \leftarrow p(X, Y, Z), r(X, Z)$$

### instancias

$$\{v_1(i, j, 1) \mid \text{si } x_i \text{ ocurre en } c_j\} \cup$$

$$\{v_1(i, j, 0) \mid \text{si } \neg x_i \text{ ocurre en } c_j\}$$

$$\{v_2(1), v_2(2), \dots, v_2(n)\}$$

$$\{v_3(1), v_3(2), \dots, v_3(m)\}$$

Idea: BD posibles representan asignaciones que satisfacen  $\varphi$

- ▶  $p(X, Y, Z)$ : qué literal es verdadero en qué  $c_j$
- ▶ materialización de  $v_1$  representa a la formula  $\varphi$

# coNP-hard para consultas con $\neq$ , demostración (¿incorrecta?)

## Demostracion (Idea)

Dada  $\varphi = c_1 \wedge \dots \wedge c_m$  con variables  $\{x_1, \dots, x_n\}$  se construye

*vistas*

$$v_1(X, Y, Z) \leftarrow p(X, Y, Z)$$

$$v_2(X) \leftarrow r(X, Y)$$

$$v_3(Y) \leftarrow p(X, Y, Z), r(X, Z)$$

*instancias*

$$\{v_1(i, j, 1) \mid \text{si } x_i \text{ ocurre en } c_j\} \cup$$

$$\{v_1(i, j, 0) \mid \text{si } \neg x_i \text{ ocurre en } c_j\}$$

$$\{v_2(1), v_2(2), \dots, v_2(n)\}$$

$$\{v_3(1), v_3(2), \dots, v_3(m)\}$$

*Idea: BD posibles representan asignaciones que satisfacen  $\varphi$*

- *materialización de  $v_3$  exige que haya al menos un literal positivo por cada  $c_i$*

# coNP-hard para consultas con $\neq$ , demostración (¿incorrecta?)

## Demostracion (Idea)

Dada  $\varphi = c_1 \wedge \dots \wedge c_m$  con variables  $\{x_1, \dots, x_n\}$  se construye

*vistas*

$$v_1(X, Y, Z) \leftarrow p(X, Y, Z)$$

$$v_2(X) \leftarrow r(X, Y)$$

$$v_3(Y) \leftarrow p(X, Y, Z), r(X, Z)$$

*instancias*

$$\{v_1(i, j, 1) \mid \text{si } x_i \text{ ocurre en } c_j\} \cup$$

$$\{v_1(i, j, 0) \mid \text{si } \neg x_i \text{ ocurre en } c_j\}$$

$$\{v_2(1), v_2(2), \dots, v_2(n)\}$$

$$\{v_3(1), v_3(2), \dots, v_3(m)\}$$

*Idea: BD posibles representan asignaciones que satisfacen  $\varphi$*

- ▶ *consulta*  $q(c) \leftarrow r(X, Y), r(X, Y'), Y \neq Y'$
- ▶ *chequea (in)consistencia de la asignación.*
- ▶ *c es cierta sii toda BD posible es una asignación inconsistente, sii  $\varphi$  no es satisfactible*

# coNP-hard para consultas con $\neq$ , demostración (¿incorrecta?)

## Demostracion (Idea)

Dada  $\varphi = c_1 \wedge \dots \wedge c_m$  con variables  $\{x_1, \dots, x_n\}$  se construye

*vistas*

$$v_1(X, Y, Z) \leftarrow p(X, Y, Z)$$

$$v_2(X) \leftarrow r(X, Y)$$

$$v_3(Y) \leftarrow p(X, Y, Z), r(X, Z)$$

*instancias*

$$\{v_1(i, j, 1) \mid \text{si } x_i \text{ ocurre en } c_j\} \cup$$

$$\{v_1(i, j, 0) \mid \text{si } \neg x_i \text{ ocurre en } c_j\}$$

$$\{v_2(1), v_2(2), \dots, v_2(n)\}$$

$$\{v_3(1), v_3(2), \dots, v_3(m)\}$$

Idea: BD posibles representan asignaciones que satisfacen  $\varphi$

- ▶ *consulta*  $q(c) \leftarrow r(X, Y), r(X, Y'), Y \neq Y'$
- ▶ chequea (in)consistencia de la asignación.
- ▶  $c$  es certera sii toda BD posible es una asignación inconsistente, sii  $\varphi$  no es satisfactible

# Indecible para vistas DATALOG y consultas conjuntivas con $\neq$

## Teorema

*Para vistas DATALOG y consultas conjuntivas con  $\neq$   
el problema de evaluación certera es indecible.*

La demostración usa una reducción desde el *Problema de Correspondencia de Post*:

Dadas dos listas finitas de palabras (de largo al menos 2) sobre  $\Sigma$

$$w_1, w_2, \dots, w_n \quad z_1, z_2, \dots, z_n$$

determinar si existe una secuencia de índices  $i_1, i_2, \dots, i_k$  tales que

$$w_{i_1} w_{i_2} \cdots w_{i_k} = z_{i_1} z_{i_2} \cdots z_{i_k}$$

Reducción no trivial (programa recursivo).

# Indecible para vistas DATALOG y consultas conjuntivas con $\neq$

## Teorema

*Para vistas DATALOG y consultas conjuntivas con  $\neq$   
el problema de evaluación certera es indecible.*

La demostración usa una reducción desde el *Problema de Correspondencia de Post*:

Dadas dos listas finitas de palabras (de largo al menos 2) sobre  $\Sigma$

$$w_1, w_2, \dots, w_n \quad z_1, z_2, \dots, z_n$$

determinar si existe una secuencia de índices  $i_1, i_2, \dots, i_k$  tales que

$$w_{i_1} w_{i_2} \cdots w_{i_k} = z_{i_1} z_{i_2} \cdots z_{i_k}$$

Reducción no trivial (programa recursivo).

# Indecidible para consultas de primer orden

## Teorema

*Para vistas conjuntivas y consultas en lógica de primer orden  
el problema de evaluación certera es indecidible.*

## Demostracion (Idea)

*Considere la consulta*

$$q(c) \leftarrow \neg \varphi$$

*con  $\varphi$  una fórmula de primer orden*

- ▶  *$c$  es respuesta certera sii  $\varphi \models D$  es falso para toda BD  $D$*
- ▶ *si y solo si  $\varphi$  no es finitamente satisfactible.*
- ▶ *chequear si  $\varphi$  admite un modelo finito es indecidible.*

El resultado también se cumple para vistas exactas.

# Indecidible para consultas de primer orden

## Teorema

*Para vistas conjuntivas y consultas en lógica de primer orden  
el problema de evaluación certera es indecible.*

## Demostracion (Idea)

*Considere la consulta*

$$q(c) \leftarrow \neg \varphi$$

*con  $\varphi$  una fórmula de primer orden*

- ▶  *$c$  es respuesta certera sii  $\varphi \models D$  es falso para toda BD  $D$*
- ▶ *si y solo si  $\varphi$  no es finitamente satisfactible.*
- ▶ *chequear si  $\varphi$  admite un modelo finito es indecible.*

El resultado también se cumple para vistas exactas.

# Indecidible para consultas de primer orden

## Teorema

*Para vistas conjuntivas y consultas en lógica de primer orden  
el problema de evaluación certera es indecible.*

## Demostracion (Idea)

*Considere la consulta*

$$q(c) \leftarrow \neg \varphi$$

*con  $\varphi$  una fórmula de primer orden*

- ▶  *$c$  es respuesta certera sii  $\varphi \models D$  es falso para toda BD  $D$*
- ▶ *si y solo si  $\varphi$  no es finitamente satisfactible.*
- ▶ *chequear si  $\varphi$  admite un modelo finito es indecible.*

*El resultado también se cumple para vistas exactas.*

# Indecidable para consultas de primer orden

## Teorema

*Para vistas conjuntivas y consultas en lógica de primer orden  
el problema de evaluación certera es indecidible.*

## Demostracion (Idea)

*Considere la consulta*

$$q(c) \leftarrow \neg \varphi$$

*con  $\varphi$  una fórmula de primer orden*

- ▶  *$c$  es respuesta certera sii  $\varphi \models D$  es falso para toda BD  $D$*
- ▶ *si y solo si  $\varphi$  no es finitamente satisfactible.*
- ▶ *chequear si  $\varphi$  admite un modelo finito es indecidible.*

El resultado también se cumple para vistas exactas.

# Indecidible para consultas de primer orden

## Teorema

*Para vistas conjuntivas y consultas en lógica de primer orden  
el problema de evaluación certera es indecidible.*

## Demostracion (Idea)

*Considere la consulta*

$$q(c) \leftarrow \neg \varphi$$

*con  $\varphi$  una fórmula de primer orden*

- ▶  *$c$  es respuesta certera sii  $\varphi \models D$  es falso para toda BD  $D$*
- ▶ *si y solo si  $\varphi$  no es finitamente satisfactible.*
- ▶ *chequear si  $\varphi$  admite un modelo finito es indecidible.*

El resultado también se cumple para vistas exactas.

# Indecidible para vistas de primer orden

## Teorema

*Para vistas en lógica de primer orden y consultas conjuntivas  
el problema de evaluación certera es indecidible.*

## Demostracion (Idea)

*vistas* *instancias*  
 $v(c) \leftarrow \varphi(X) \vee p(X) \quad \{v(c)\}$

*consulta*  $q(c) \leftarrow p(X)$

- ▶ Si  $\varphi$  es satisfactible existe  $D$ , tal que  $\varphi \models D$  y  $p$  es vacío en  $D$ , luego  $c$  no es certera.
- ▶ Si  $c$  no es certera existe un BD posible  $D$  tal que  $p$  es vacío en  $D$ , luego  $\varphi \models D$ .

El resultado también se cumple para vistas exactas

# Indecidable para vistas de primer orden

## Teorema

*Para vistas en lógica de primer orden y consultas conjuntivas  
el problema de evaluación certera es indecible.*

## Demostracion (Idea)

*vistas* *instancias*  
 $v(c) \leftarrow \varphi(X) \vee p(X) \quad \{v(c)\}$

*consulta*  $q(c) \leftarrow p(X)$

- ▶ Si  $\varphi$  es satisfactible existe  $D$ , tal que  $\varphi \models D$  y  $p$  es vacío en  $D$ , luego  $c$  no es certera.
- ▶ Si  $c$  no es certera existe un BD posible  $D$  tal que  $p$  es vacío en  $D$ , luego  $\varphi \models D$ .

El resultado también se cumple para vistas exactas

# Indecidible para vistas de primer orden

## Teorema

*Para vistas en lógica de primer orden y consultas conjuntivas  
el problema de evaluación certera es indecidible.*

## Demostracion (Idea)

*vistas* *instancias*  
 $v(c) \leftarrow \varphi(X) \vee p(X) \quad \{v(c)\}$

*consulta*  $q(c) \leftarrow p(X)$

- ▶ Si  $\varphi$  es satisfactible existe  $D$ , tal que  $\varphi \models D$  y  $p$  es vacío en  $D$ , luego  $c$  no es certera.
- ▶ Si  $c$  no es certera existe un BD posible  $D$  tal que  $p$  es vacío en  $D$ , luego  $\varphi \models D$ .

El resultado también se cumple para vistas exactas

Algo de complejidad: NP, coNP, no recursivos

Conceptos previos

Tablas condicionales

Complejidad usando vistas sanas (OWA)

Complejidad usando vistas exactas (CWA)

# Para vistas exactas todo se complica.

- ▶ Hasta ahora vimos resultados en particular para vistas sanas
- ▶ Cuando se consideran vistas exactas, incluso los casos mas simples se vuelven intratables.

## Teorema

*Para vistas conjuntivas exactas y consultas conjuntivas*

*el problema de evaluación certera es coNP-hard.*

## Demostracion (Idea)

*Reducción desde el complemento de 3-coloración:*

*Dado  $G = (V, E)$*

### *vistas*

$v_1(X) \leftarrow \text{color}(X, Y)$

$v_2(Y) \leftarrow \text{color}(X, Y)$

$v_3(X, Y) \leftarrow \text{arista}(X, Y)$

### *instancias*

$\{v_1(x) \mid x \in V\}$

$\{v_2(B), v_2(A), v_2(R)\}$

$\{v_3(x, y) \mid (x, y) \in E\}$

## Teorema

*Para vistas conjuntivas exactas y consultas conjuntivas*

*el problema de evaluación certera es coNP-hard.*

## Demostracion (Idea)

*Reducción desde el complemento de 3-coloración:*

*Dado  $G = (V, E)$*

### *vistas*

$v_1(X) \leftarrow \text{color}(X, Y)$

$v_2(Y) \leftarrow \text{color}(X, Y)$

$v_3(X, Y) \leftarrow \text{arista}(X, Y)$

### *instancias*

$\{v_1(x) \mid x \in V\}$

$\{v_2(B), v_2(A), v_2(R)\}$

$\{v_3(x, y) \mid (x, y) \in E\}$

- *BDs posibles: todas las posibles asignaciones de tres colores.*

## Teorema

*Para vistas conjuntivas exactas y consultas conjuntivas*

*el problema de evaluación certera es coNP-hard.*

## Demostracion (Idea)

*Reducción desde el complemento de 3-coloración:*

*Dado  $G = (V, E)$*

*vistas*

$v_1(X) \leftarrow \text{color}(X, Y)$

$v_2(Y) \leftarrow \text{color}(X, Y)$

$v_3(X, Y) \leftarrow \text{arista}(X, Y)$

*instancias*

$\{v_1(x) \mid x \in V\}$

$\{v_2(B), v_2(A), v_2(R)\}$

$\{v_3(x, y) \mid (x, y) \in E\}$

► *consulta*  $q(c) \leftarrow \text{arista}(X, Y), \text{color}(X, Z), \text{color}(Y, Z)$

## Teorema

*Para vistas conjuntivas exactas y consultas conjuntivas*

*el problema de evaluación certera es coNP-hard.*

## Demostracion (Idea)

*Reducción desde el complemento de 3-coloración:*

*Dado  $G = (V, E)$*

### *vistas*

$v_1(X) \leftarrow \text{color}(X, Y)$

$v_2(Y) \leftarrow \text{color}(X, Y)$

$v_3(X, Y) \leftarrow \text{arista}(X, Y)$

### *instancias*

$\{v_1(x) \mid x \in V\}$

$\{v_2(B), v_2(A), v_2(R)\}$

$\{v_3(x, y) \mid (x, y) \in E\}$

► *c es certera sii toda BD posible es una coloración inconsistente*

- ▶ El artículo presenta una *batería* importante de resultados de complejidad.
- ▶ Propone (en su tiempo) usar TCs para manejar respuestas basadas en vistas
- ▶ Hace conexiones con otros problemas relevantes en manejo de información a gran escala (no los vimos)