

Fundamentos de Bases de Datos de Grafos

Marcelo Arenas

Pontificia Universidad Católica de Chile
Instituto Milenio Fundamentos de los Datos

¿Cuáles son los objetivos de esta charla?

¿Cuáles son los objetivos de esta charla?

- ▶ Que ustedes aprendan sobre bases datos de grafos

¿Cuáles son los objetivos de esta charla?

- ▶ Que ustedes aprendan sobre bases datos de grafos
 - ▶ ¡Preguntas son muy bienvenidas!

¿Cuáles son los objetivos de esta charla?

- ▶ Que ustedes aprendan sobre bases datos de grafos
 - ▶ ¡Preguntas son muy bienvenidas!

- ▶ Que vamos a aprender:

¿Cuáles son los objetivos de esta charla?

- ▶ Que ustedes aprendan sobre bases datos de grafos
 - ▶ ¡Preguntas son muy bienvenidas!
- ▶ Que vamos a aprender:
 - ▶ Cómo un grafo puede verse como un modelo de datos

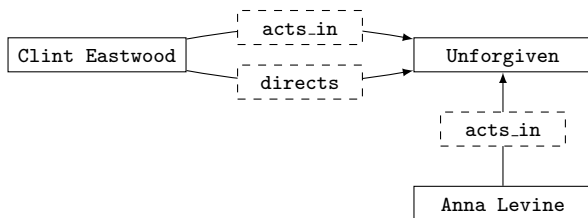
¿Cuáles son los objetivos de esta charla?

- ▶ Que ustedes aprendan sobre bases datos de grafos
 - ▶ ¡Preguntas son muy bienvenidas!
- ▶ Que vamos a aprender:
 - ▶ Cómo un grafo puede verse como un modelo de datos
 - ▶ Cómo se ve un lenguaje de consultas para grafos

¿Cuáles son los objetivos de esta charla?

- ▶ Que ustedes aprendan sobre bases datos de grafos
 - ▶ ¡Preguntas son muy bienvenidas!
- ▶ Que vamos a aprender:
 - ▶ Cómo un grafo puede verse como un modelo de datos
 - ▶ Cómo se ve un lenguaje de consultas para grafos
 - ▶ Qué rol juegan los caminos en las bases de datos de grafos

Un primer modelo: grafos con arcos etiquetados



Grafos con arcos etiquetados: formalización

Un grafo con arcos etiquetados es un par (V, A) , donde:

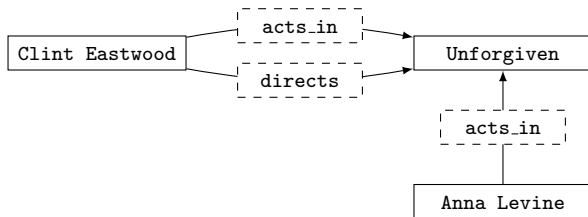
- ▶ V es un conjunto finito de vértices (o nodos)
- ▶ A es un conjunto finito de arcos etiquetados

Grafos con arcos etiquetados: formalización

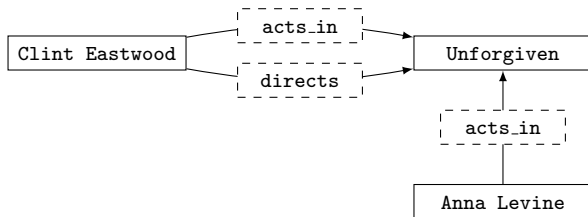
Un grafo con arcos etiquetados es un par (V, A) , donde:

- ▶ V es un conjunto finito de vértices (o nodos)
- ▶ A es un conjunto finito de arcos etiquetados
 - ▶ $A \subseteq V \times Lab \times V$ donde Lab es un conjunto de etiquetas

La formalización del ejemplo inicial

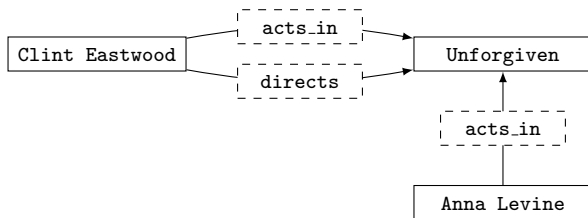


La formalización del ejemplo inicial



$V = \{ \text{Clint Eastwood, Unforgiven, Anna Levine} \}$

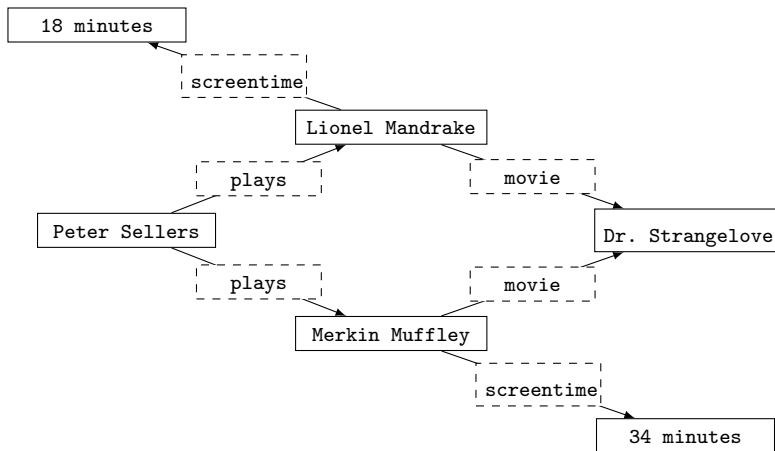
La formalización del ejemplo inicial



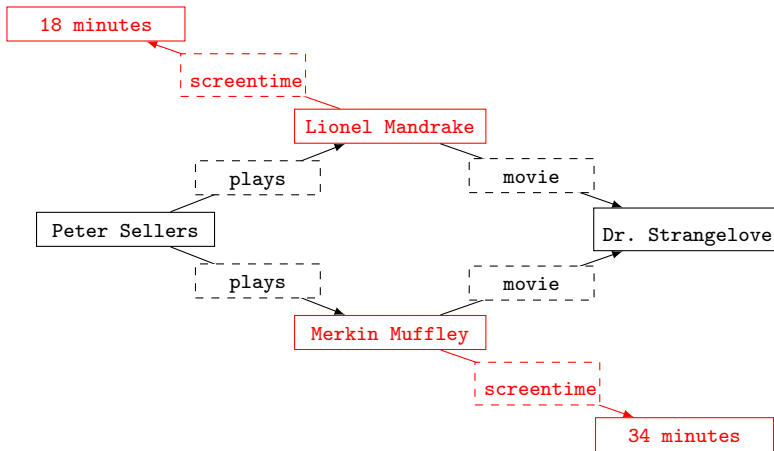
$V = \{ \text{Clint Eastwood, Unforgiven, Anna Levine} \}$

$A = \{ (\text{Clint Eastwood, acts_in, Unforgiven}),$
 $(\text{Clint Eastwood, directs, Unforgiven}),$
 $(\text{Anna Levine, acts_in, Unforgiven}) \}$

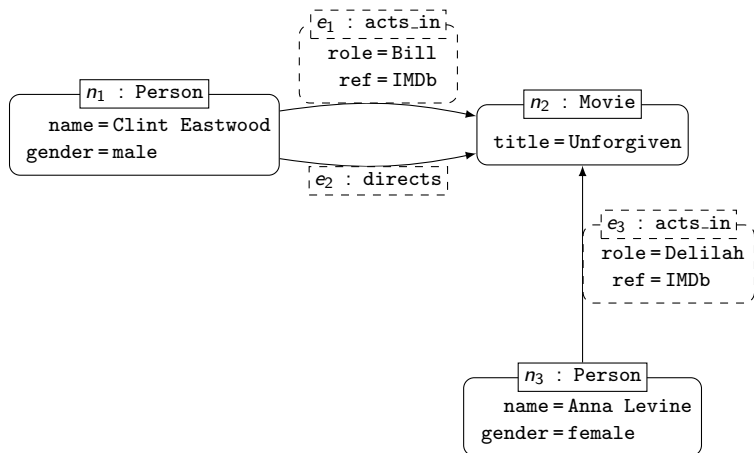
Un actor con papeles multiples en la misma película



Un actor con papeles multiples en la misma película



Un segundo modelo: grafos con propiedades



Grafos con propiedades: formalización

Un grafo con propiedades G es una tupla $(V, A, \rho, \lambda, \sigma)$ tal que:

- ▶ V es un conjunto finito de vértices
- ▶ A es un conjunto finito de arcos
- ▶ $\rho : A \rightarrow (V \times V)$ es una función (total)
 - ▶ $\rho(e) = (v_1, v_2)$ indica que e va desde v_1 a v_2

Grafos con propiedades: formalización

Un grafo con propiedades G es una tupla $(V, A, \rho, \lambda, \sigma)$ tal que:

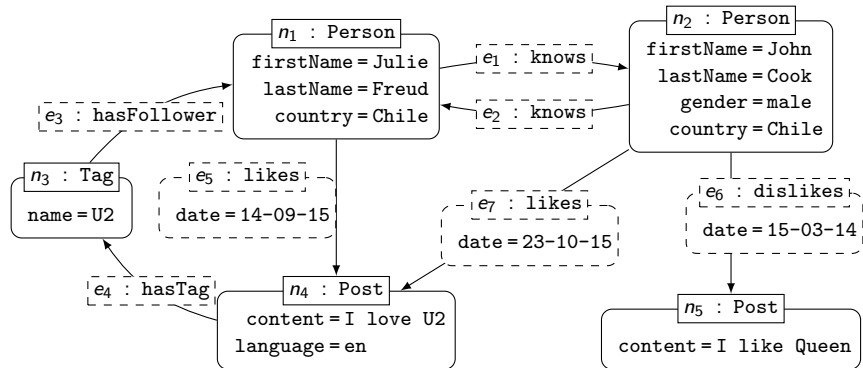
- ▶ V es un conjunto finito de vértices
- ▶ A es un conjunto finito de arcos
- ▶ $\rho : A \rightarrow (V \times V)$ es una función (total)
 - ▶ $\rho(e) = (v_1, v_2)$ indica que e va desde v_1 a v_2
- ▶ $\lambda : (V \cup E) \rightarrow Lab$ es una función con Lab un conjunto de etiquetas

Grafos con propiedades: formalización

Un grafo con propiedades G es una tupla $(V, A, \rho, \lambda, \sigma)$ tal que:

- ▶ V es un conjunto finito de vértices
- ▶ A es un conjunto finito de arcos
- ▶ $\rho : A \rightarrow (V \times V)$ es una función (total)
 - ▶ $\rho(e) = (v_1, v_2)$ indica que e va desde v_1 a v_2
- ▶ $\lambda : (V \cup E) \rightarrow Lab$ es una función con Lab un conjunto de etiquetas
- ▶ $\sigma : (V \cup E) \times Prop \rightarrow Val$ es una función parcial con $Prop$ un conjunto de propiedades y Val un conjunto de valores
 - ▶ $\sigma(o, p) = s$ indica que el valor de la propiedad p para o es s

Un ejemplo de redes sociales



Extracción de información desde un grafo

¿Qué funcionalidades debería tener un lenguaje de consultas para bases de datos de grafos?

Extracción de información desde un grafo

¿Qué funcionalidades debería tener un lenguaje de consultas para bases de datos de grafos?

- ▶ Queremos un lenguaje declarativo

Extracción de información desde un grafo

¿Qué funcionalidades debería tener un lenguaje de consultas para bases de datos de grafos?

- ▶ Queremos un lenguaje declarativo

Vamos a mostrar de manera incremental cómo se construye tal lenguaje

Extracción de información desde un grafo

¿Qué funcionalidades debería tener un lenguaje de consultas para bases de datos de grafos?

- ▶ Queremos un lenguaje declarativo

Vamos a mostrar de manera incremental cómo se construye tal lenguaje

- ▶ **Patrón básico:** bloque *mínimo* para extraer información desde un grafo

Extracción de información desde un grafo

¿Qué funcionalidades debería tener un lenguaje de consultas para bases de datos de grafos?

- ▶ Queremos un lenguaje declarativo

Vamos a mostrar de manera incremental cómo se construye tal lenguaje

- ▶ **Patrón básico:** bloque *mínimo* para extraer información desde un grafo
- ▶ **Patrón complejo:** combinación de patrones básicos

Extracción de información desde un grafo

¿Qué funcionalidades debería tener un lenguaje de consultas para bases de datos de grafos?

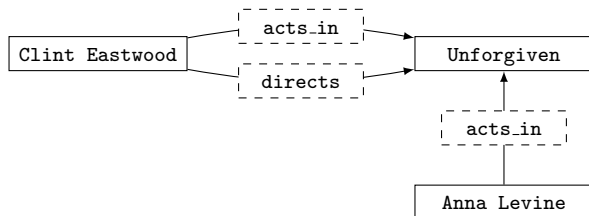
- ▶ Queremos un lenguaje declarativo

Vamos a mostrar de manera incremental cómo se construye tal lenguaje

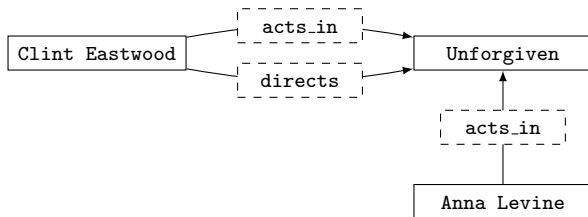
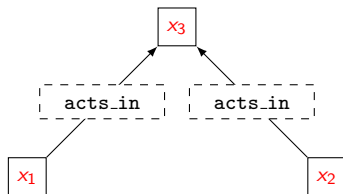
- ▶ **Patrón básico:** bloque *mínimo* para extraer información desde un grafo
- ▶ **Patrón complejo:** combinación de patrones básicos
- ▶ **Patrón de navegación:** permite extraer caminos desde un grafo

Patrones básicos: grafos con arcos etiquetados

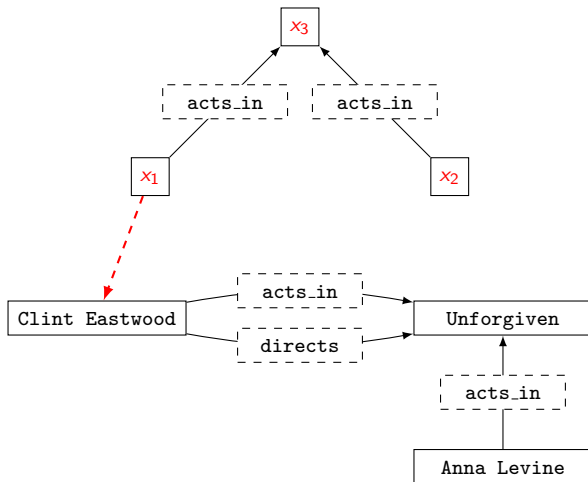
Consideremos primero un grafo con arcos etiquetados:



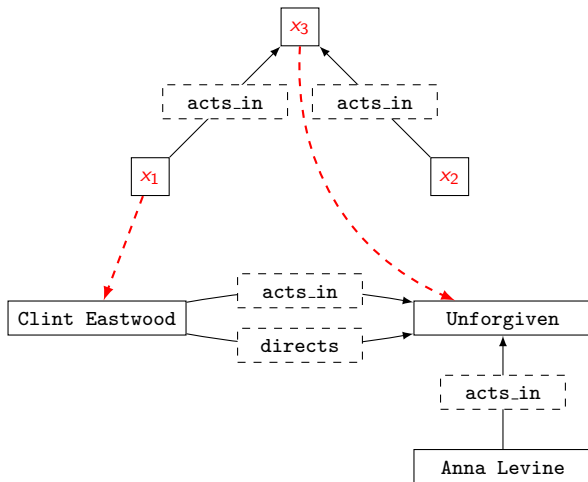
Patrones básicos: grafos con arcos etiquetados



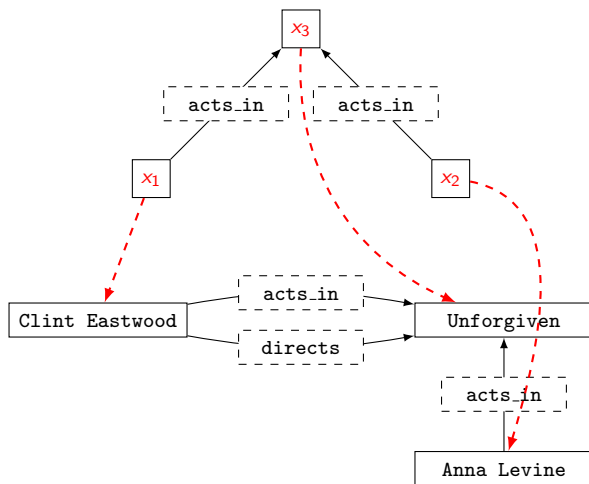
Patrones básicos: grafos con arcos etiquetados



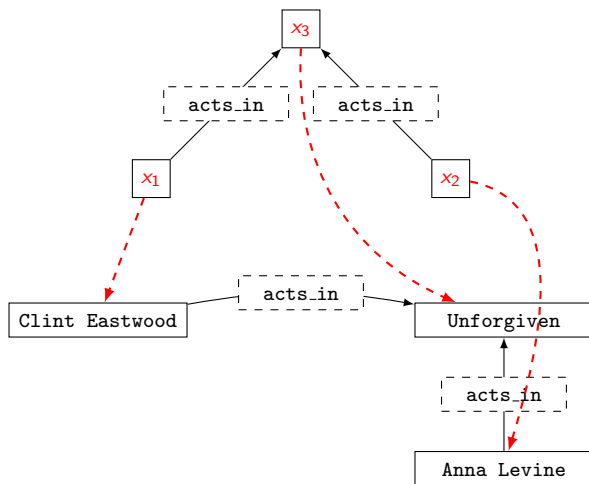
Patrones básicos: grafos con arcos etiquetados



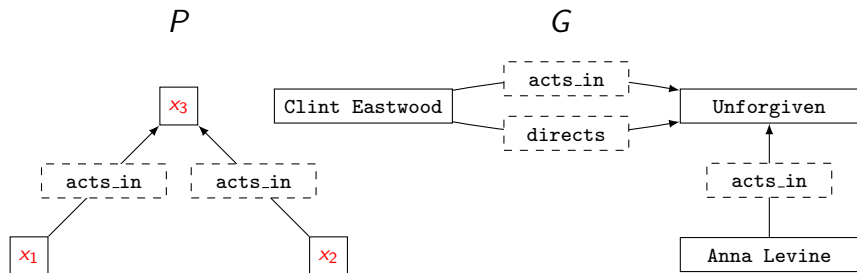
Patrones básicos: grafos con arcos etiquetados



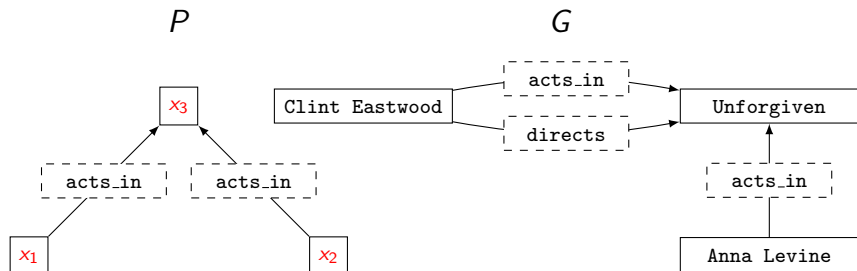
Patrones básicos: grafos con arcos etiquetados



La respuesta completa $P(G)$



La respuesta completa $P(G)$



$P(G)$:

x_1	x_2	x_3
Clint Eastwood	Anna Levine	Unforgiven
Anna Levine	Clint Eastwood	Unforgiven
Clint Eastwood	Clint Eastwood	Unforgiven
Anna Levine	Anna Levine	Unforgiven

La respuesta completa $P(G)$

$P(G)$ es llamada la salida del patrón P al ser evaluado sobre un grafo G

- ▶ En general, la notación $P(G)$ se utilizada para la salida de una consulta P al ser evaluada sobre un grafo G

$P(G)$ es un conjunto de respuestas, donde cada respuesta es una función desde las variables de P a los distintos elementos que componen G

- ▶ Nodos, arcos y valores en G

La respuesta completa $P(G)$

$P(G)$ es llamada la salida del patrón P al ser evaluado sobre un grafo G

- ▶ En general, la notación $P(G)$ se utilizada para la salida de una consulta P al ser evaluada sobre un grafo G

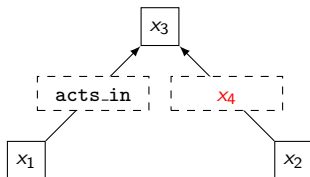
$P(G)$ es un conjunto de respuestas, donde cada respuesta es una función desde las variables de P a los distintos elementos que componen G

- ▶ Nodos, arcos y valores en G

Para enfatizar que las respuestas son funciones decimos que son mappings, y que $P(G)$ es un conjunto de mappings.

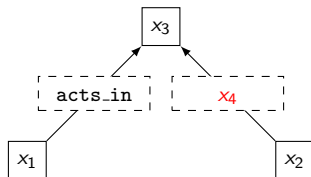
La versión general de un patrón básico

Las variables pueden ser utilizadas tanto para nodos como etiquetas de arcos:

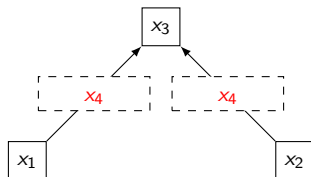


La versión general de un patrón básico

Las variables pueden ser utilizadas tanto para nodos como etiquetas de arcos:

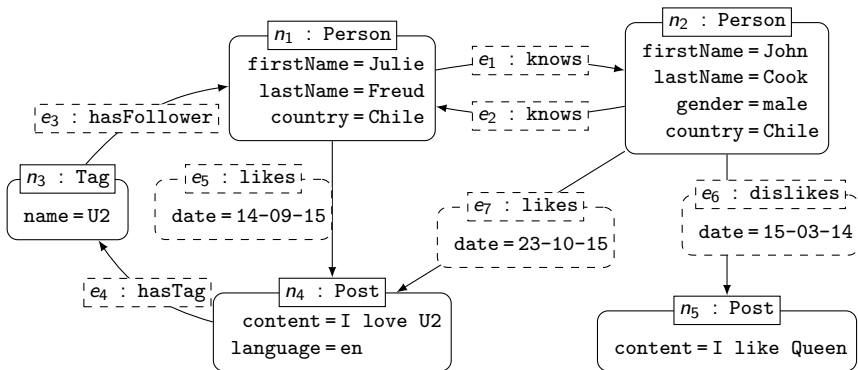


Las variables pueden ser reutilizadas:

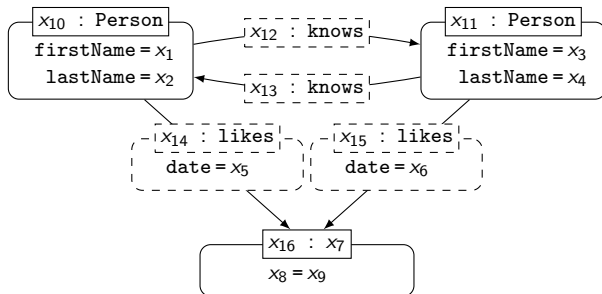


Patrones básicos: grafos con propiedades

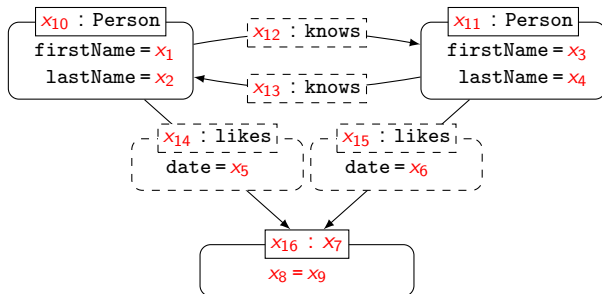
Consideramos ahora un grafo con propiedades:



Patrones básicos: grafos con propiedades



Patrones básicos: grafos con propiedades



La respuesta completa

x_1	x_2	x_3	x_4	x_5	...
Julie	Freud	John	Cook	14-09-15	...
John	Cook	Julie	Freud	23-10-15	...
Julie	Freud	John	Cook	14-09-15	...
John	Cook	Julie	Freud	23-10-15	...

...	x_6	x_7	x_8	x_9	x_{10}	...
...	23-10-15	Post	content	I love U2	n_1	...
...	14-09-15	Post	content	I love U2	n_2	...
...	23-10-15	Post	language	en	n_1	...
...	14-09-15	Post	language	en	n_2	...

La respuesta completa

x ₁	x ₂	x ₃	x ₄	x ₅	...
Julie	Freud	John	Cook	14-09-15	...
John	Cook	Julie	Freud	23-10-15	...
Julie	Freud	John	Cook	14-09-15	...
John	Cook	Julie	Freud	23-10-15	...

...	x ₆	x ₇	x ₈	x ₉	x ₁₀	...
...	23-10-15	Post	content	I love U2	n ₁	...
...	14-09-15	Post	content	I love U2	n ₂	...
...	23-10-15	Post	language	en	n ₁	...
...	14-09-15	Post	language	en	n ₂	...

La respuesta completa

x ₁	x ₂	x ₃	x ₄	x ₅	...
Julie	Freud	John	Cook	14-09-15	...
John	Cook	Julie	Freud	23-10-15	...
Julie	Freud	John	Cook	14-09-15	...
John	Cook	Julie	Freud	23-10-15	...

...	x ₆	x ₇	x ₈	x ₉	x ₁₀	...
...	23-10-15	Post	content	I love U2	n ₁	...
...	14-09-15	Post	content	I love U2	n ₂	...
...	23-10-15	Post	language	en	n ₁	...
...	14-09-15	Post	language	en	n ₂	...

Sobre la semántica de los patrones básicos

Sobre la semántica de los patrones básicos

- ▶ La forma de evaluación que definimos está basada en la noción de **homomorfismo**
 - ▶ En cada respuesta definimos una función f del patrón básico P al grafo G , y esta función f es un homomorfismo

Sobre la semántica de los patrones básicos

- ▶ La forma de evaluación que definimos está basada en la noción de **homomorfismo**
 - ▶ En cada respuesta definimos una función f del patrón básico P al grafo G , y esta función f es un homomorfismo

- ▶ Una forma alternativa de evaluación está basada en la noción de **isomorfismo**

Sobre la semántica de los patrones básicos

- ▶ La forma de evaluación que definimos está basada en la noción de **homomorfismo**
 - ▶ En cada respuesta definimos una función f del patrón básico P al grafo G , y esta función f es un homomorfismo
- ▶ Una forma alternativa de evaluación está basada en la noción de **isomorfismo**
 - ▶ Versión estricta: cada función f debe ser inyectiva

Sobre la semántica de los patrones básicos

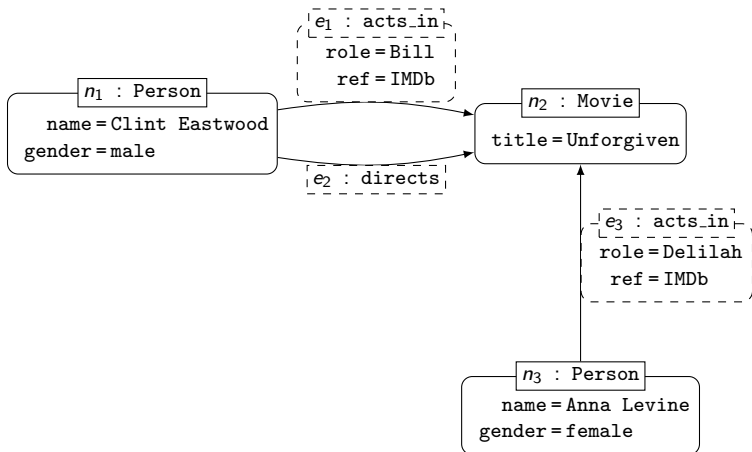
- ▶ La forma de evaluación que definimos está basada en la noción de **homomorfismo**
 - ▶ En cada respuesta definimos una función f del patrón básico P al grafo G , y esta función f es un homomorfismo
- ▶ Una forma alternativa de evaluación está basada en la noción de **isomorfismo**
 - ▶ Versión estricta: cada función f debe ser inyectiva
 - ▶ Sin repetición de vértices: cada función f debe ser inyectiva cuando se restringe a las variables que contienen vértices

Sobre la semántica de los patrones básicos

- ▶ La forma de evaluación que definimos está basada en la noción de **homomorfismo**
 - ▶ En cada respuesta definimos una función f del patrón básico P al grafo G , y esta función f es un homomorfismo
- ▶ Una forma alternativa de evaluación está basada en la noción de **isomorfismo**
 - ▶ Versión estricta: cada función f debe ser inyectiva
 - ▶ Sin repetición de vértices: cada función f debe ser inyectiva cuando se restringe a las variables que contienen vértices
 - ▶ Sin repetición de arcos: cada función f debe ser inyectiva cuando se restringe a las variables que contienen arcos

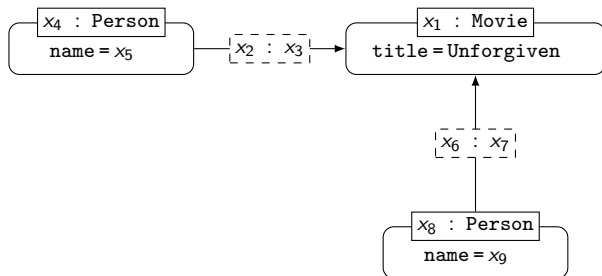
Un ejemplo de las diferentes semánticas

Considere el siguiente grafo con propiedades:



Un ejemplo de las diferentes semánticas

Queremos evaluar el siguiente patrón básico:



Un ejemplo de las diferentes semánticas

Evaluación basada en homomorfismos:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
n_2	e_2	directs	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood
n_2	e_3	acts_in	n_1	Anna Levine	e_3	acts_in	n_1	Anna Levine

Un ejemplo de las diferentes semánticas

Evaluación basada en **isomorfismos (versión estricta)**:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
n_2	e_2	directs	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood
n_2	e_3	acts_in	n_1	Anna Levine	e_3	acts_in	n_1	Anna Levine

Un ejemplo de las diferentes semánticas

Evaluación basada en isomorfismos (versión estricta):

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
n_2	e_2	directs	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_2	directs	n_1	Clint Eastwood

Un ejemplo de las diferentes semánticas

Evaluación basada en **isomorfismos (sin repetición de nodos)**:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
n_2	e_2	directs	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood
n_2	e_3	acts_in	n_1	Anna Levine	e_3	acts_in	n_1	Anna Levine

Un ejemplo de las diferentes semánticas

Evaluación basada en isomorfismos (sin repetición de nodos):

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
n_2	e_2	directs	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_1	acts_in	n_1	Clint Eastwood

Un ejemplo de las diferentes semánticas

Evaluación basada en **isomorfismos (sin repetición de arcos)**:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
n_2	e_2	directs	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood
n_2	e_3	acts_in	n_1	Anna Levine	e_3	acts_in	n_1	Anna Levine

Un ejemplo de las diferentes semánticas

Evaluación basada en isomorfismos (sin repetición de arcos):

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
n_2	e_2	directs	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_2	directs	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_3	acts_in	n_3	Anna Levine
n_2	e_3	acts_in	n_3	Anna Levine	e_1	acts_in	n_1	Clint Eastwood
n_2	e_2	directs	n_1	Clint Eastwood	e_1	acts_in	n_1	Clint Eastwood
n_2	e_1	acts_in	n_1	Clint Eastwood	e_2	directs	n_1	Clint Eastwood

Patrones complejos: combinando patrones básicos

Utilizamos una semántica para patrones básicos basada en la noción de homomorfismo

- ▶ Es la semántica más utilizada en la práctica

Patrones complejos: combinando patrones básicos

Utilizamos una semántica para patrones básicos basada en la noción de homomorfismo

- ▶ Es la semántica más utilizada en la práctica

Un patrón complejo se construye utilizando operadores para conjuntos de mappings.

- ▶ Los operadores reciben conjuntos de mappings, y producen un conjunto de mappings

Patrones complejos: combinando patrones básicos

Utilizamos una semántica para patrones básicos basada en la noción de homomorfismo

- ▶ Es la semántica más utilizada en la práctica

Un patrón complejo se construye utilizando operadores para conjuntos de mappings.

- ▶ Los operadores reciben conjuntos de mappings, y producen un conjunto de mappings

La evaluación de un patrón complejo es bottom-up, partiendo desde los patrones básicos, para luego evaluar los operadores que los combinan de acuerdo al orden establecido en el patrón

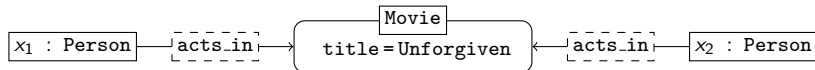
Filtrando respuestas

Se debe permitir filtrar cada respuesta a una consulta de acuerdo a una condición sobre las variables.

Filtrando respuestas

Se debe permitir filtrar cada respuesta a una consulta de acuerdo a una condición sobre las variables.

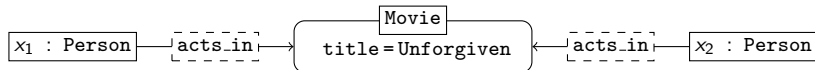
Considere un patrón básico P_1 :



Filtrando respuestas

Se debe permitir filtrar cada respuesta a una consulta de acuerdo a una condición sobre las variables.

Considere un patrón básico P_1 :

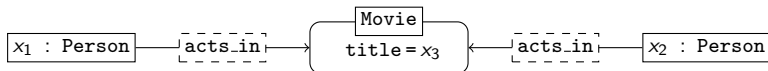


El siguiente patrón (complejo) permite retornar la lista de pares de actores distintos que actuaron en la película Unforgiven:

P_1 **FILTER** ($x_1 \neq x_2$)

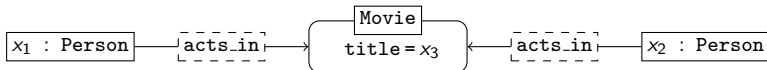
Filtrando respuestas

Considere ahora un patrón básico P_2 :



Filtrando respuestas

Considere ahora un patrón básico P_2 :



El siguiente patrón permite retornar la lista de pares de actores distintos que actuaron en la misma película:

P_2 **FILTER** ($x_1 \neq x_2$)

Nótese que cada respuesta también incluye a la película en que actuó cada par de personas.

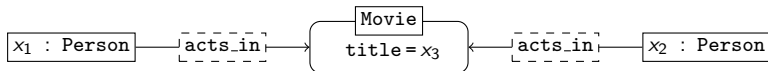
Proyección: especificando las variables de salida

Se debe permitir especificar cuáles son las variables de salida de una consulta.

Proyección: especificando las variables de salida

Se debe permitir especificar cuáles son las variables de salida de una consulta.

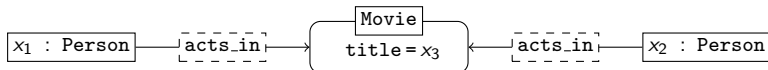
Considere nuevamente el patrón básico P_2 :



Proyección: especificando las variables de salida

Se debe permitir especificar cuáles son las variables de salida de una consulta.

Considere nuevamente el patrón básico P_2 :



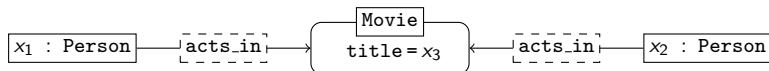
El siguiente patrón permite retornar la lista de pares de actores distintos que actuaron en la misma película:

```
SELECT x1, x2 (P2 FILTER (x1 ≠ x2))
```

En este caso **no** retornamos los nombres de las películas.

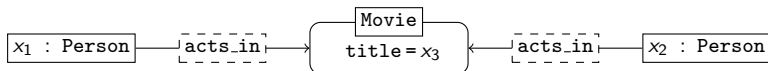
Proyección: especificando las variables de salida

Considere nuevamente P_2 :



Proyección: especificando las variables de salida

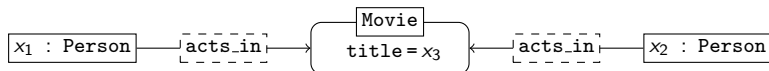
Considere nuevamente P_2 :



¿Cómo retornamos la lista de películas que tienen al menos dos actores?

Proyección: especificando las variables de salida

Considere nuevamente P_2 :

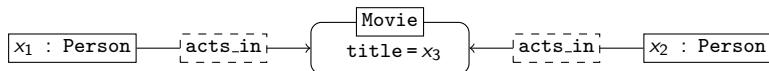


¿Cómo retornamos la lista de películas que tienen al menos dos actores?

```
SELECT x3 (P2 FILTER (x1 ≠ x2))
```

Proyección: especificando las variables de salida

Considere nuevamente P_2 :



¿Cómo retornamos la lista de películas que tienen al menos dos actores?

```
SELECT x3 (P2 FILTER (x1 ≠ x2))
```

¿Qué retorna la siguiente consulta donde el operador de filtrado fue removido?

```
SELECT x3 P2
```

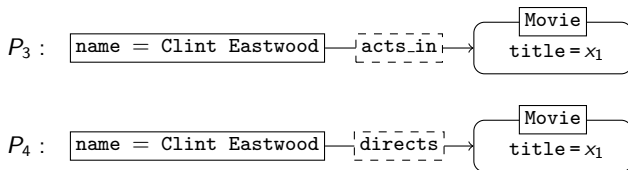
Uniendo las respuestas de dos consultas

Unir los conjuntos de resultados de dos consultas es una funcionalidad muy útil.

Uniendo las respuestas de dos consultas

Unir los conjuntos de resultados de dos consultas es una funcionalidad muy útil.

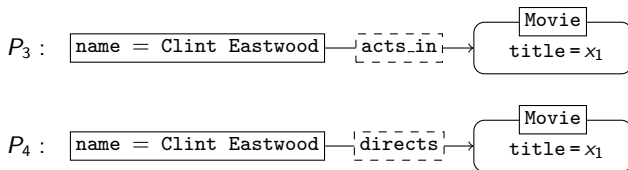
Considere los siguientes patrones básicos:



Uniendo las respuestas de dos consultas

Unir los conjuntos de resultados de dos consultas es una funcionalidad muy útil.

Considere los siguientes patrones básicos:

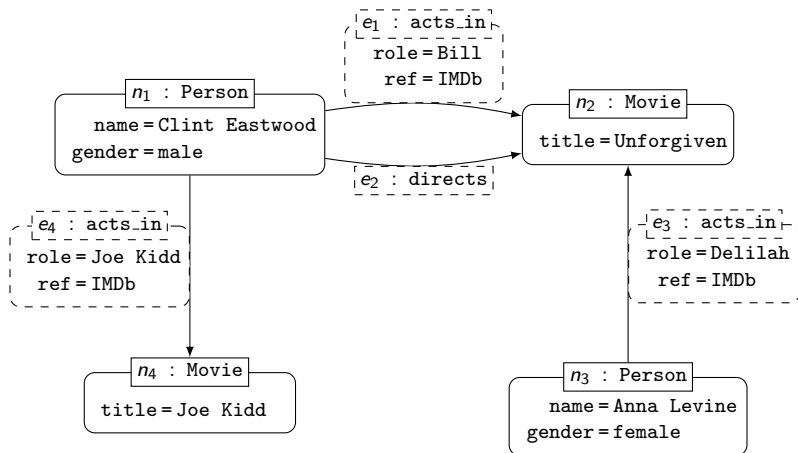


El siguiente patrón retorna la lista de las películas en las que actuó o fueron dirigidas por Clint Eastwood:

$(P_1 \text{ UNION } P_2)$

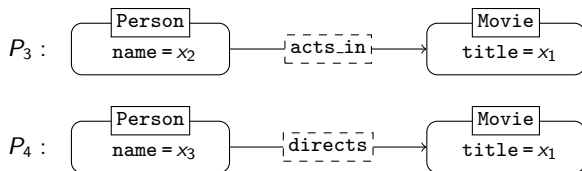
Join: cruzando datos

Considere el siguiente grafo con propiedades:



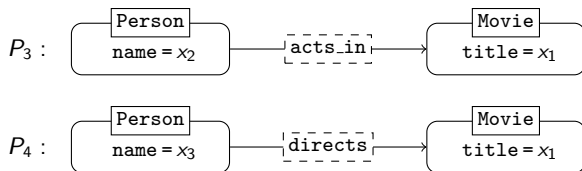
Join: cruzando datos

Considere los siguientes patrones básicos:



Join: cruzando datos

Considere los siguientes patrones básicos:



El siguiente patrón (complejo) permite cruzar los conjuntos de resultados de los patrones P_3 y P_4 :

$P_3 \text{ JOIN } P_4$

$(P_3 \text{ JOIN } P_4)$: un join en la variable x_1

P_3 :

P_4 :

$P_3 \text{ JOIN } P_4$:

$(P_3 \text{ JOIN } P_4)$: un join en la variable x_1

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

$P_3 \text{ JOIN } P_4$:

$(P_3 \text{ JOIN } P_4)$: un join en la variable x_1

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

x_1	x_3
Unforgiven	Clint Eastwood

$P_3 \text{ JOIN } P_4$:

$(P_3 \text{ JOIN } P_4)$: un join en la variable x_1

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

x_1	x_3
Unforgiven	Clint Eastwood

$P_3 \text{ JOIN } P_4$:

$(P_3 \text{ JOIN } P_4)$: un join en la variable x_1

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

x_1	x_3
Unforgiven	Clint Eastwood

$P_3 \text{ JOIN } P_4$:

$(P_3 \text{ JOIN } P_4)$: un join en la variable x_1

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

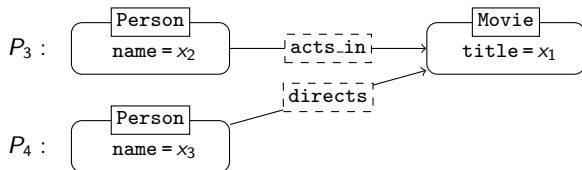
x_1	x_3
Unforgiven	Clint Eastwood

$P_3 \text{ JOIN } P_4$:

x_1	x_2	x_3
Unforgiven	Clint Eastwood	Clint Eastwood
Unforgiven	Anna Levine	Clint Eastwood

Join: cruzando datos

Nótese que $(P_3 \text{ JOIN } P_4)$ es equivalente al siguiente patrón básico:



En los lenguajes usados en la práctica un patrón complejo construido con el operador JOIN no es necesariamente equivalente a un patrón básico.

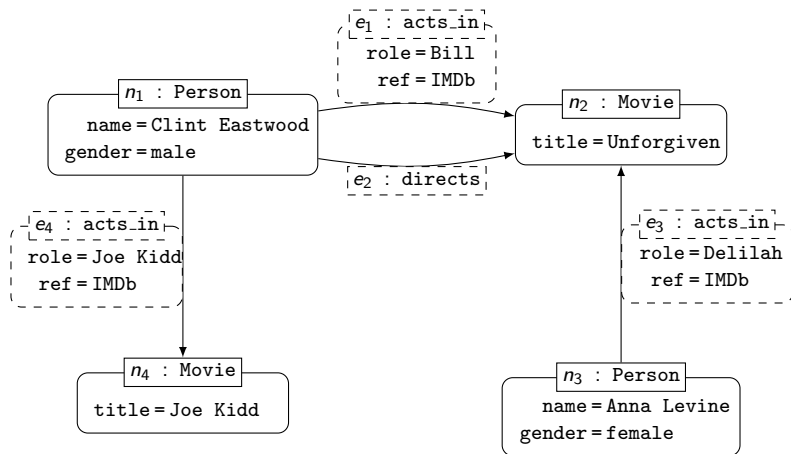
Agregando información adicional

Por la naturaleza heterogénea de las bases de datos de grafos, la posibilidad de agregar información si está disponible es una funcionalidad importante al momento de responder una consulta.

Esta idea es formalizada a través de un operador OPTIONAL

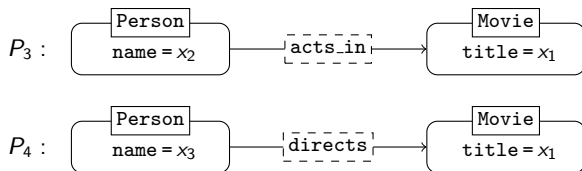
Agregando información adicional

Considere nuevamente:



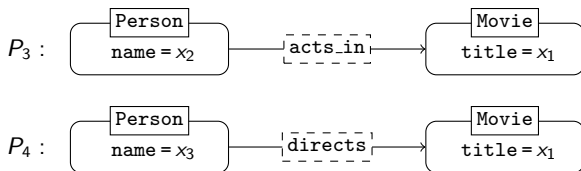
Agregando información adicional

Y considere nuevamente los siguientes patrones básicos:



Agregando información adicional

Y considere nuevamente los siguientes patrones básicos:



Queremos almacenar una lista de películas con sus actores, y **adicionalmente** la información sobre sus directores si está disponible:

P_3 OPTIONAL P_4

Los resultados de las evaluaciones

P_3 :

P_4 :

P_3 JOIN P_4 :

P_3 OPTIONAL P_4 :

Los resultados de las evaluaciones

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

P_3 JOIN P_4 :

P_3 OPTIONAL P_4 :

Los resultados de las evaluaciones

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

x_1	x_3
Unforgiven	Clint Eastwood

P_3 JOIN P_4 :

P_3 OPTIONAL P_4 :

Los resultados de las evaluaciones

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

x_1	x_3
Unforgiven	Clint Eastwood

$P_3 \text{ JOIN } P_4$:

x_1	x_2	x_3
Unforgiven	Clint Eastwood	Clint Eastwood
Unforgiven	Anna Levine	Clint Eastwood

$P_3 \text{ OPTIONAL } P_4$:

Los resultados de las evaluaciones

P_3 :

x_1	x_2
Unforgiven	Clint Eastwood
Unforgiven	Anna Levine
Joe Kidd	Clint Eastwood

P_4 :

x_1	x_3
Unforgiven	Clint Eastwood

$P_3 \text{ JOIN } P_4$:

x_1	x_2	x_3
Unforgiven	Clint Eastwood	Clint Eastwood
Unforgiven	Anna Levine	Clint Eastwood

$P_3 \text{ OPTIONAL } P_4$:

x_1	x_2	x_3
Unforgiven	Clint Eastwood	Clint Eastwood
Unforgiven	Anna Levine	Clint Eastwood
Joe Kidd	Clint Eastwood	

Una forma más general de negación

El lenguaje debe permitir una forma general de negación

- ▶ Una forma restringida de negación es permitida a través del operador FILTER

Una forma más general de negación

El lenguaje debe permitir una forma general de negación

- ▶ Una forma restringida de negación es permitida a través del operador FILTER

Una forma más general de negación es codificada a través de un operador de diferencia

Una forma más general de negación

El lenguaje debe permitir una forma general de negación

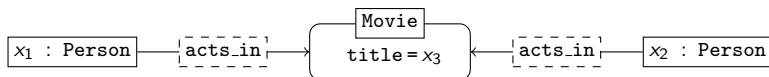
- ▶ Una forma restringida de negación es permitida a través del operador FILTER

Una forma más general de negación es codificada a través de un operador de diferencia

- ▶ La evaluación de $(Q_1 \text{ MINUS } Q_2)$ da como resultado las respuestas de Q_1 que no son respuestas de Q_2

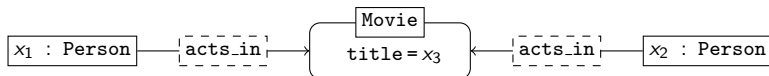
Representando la condición FILTER ($x_1 \neq x_2$)

Considere nuevamente P_2 :



Representando la condición FILTER ($x_1 \neq x_2$)

Considere nuevamente P_2 :



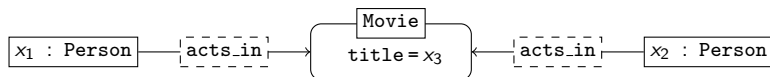
Y considere los siguientes patrones:

$$P_3 = P_2 \text{ FILTER } (x_1 = x_2)$$

$$P_4 = P_2 \text{ MINUS } P_3$$

Representando la condición FILTER ($x_1 \neq x_2$)

Considere nuevamente P_2 :



Y considere los siguientes patrones:

$$P_3 = P_2 \text{ FILTER } (x_1 = x_2)$$

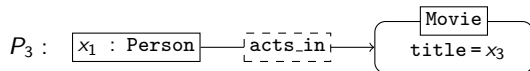
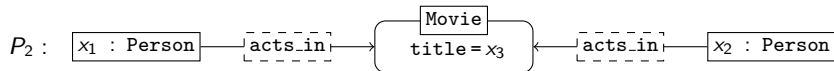
$$P_4 = P_2 \text{ MINUS } P_3$$

Tenemos que P_4 es equivalente al siguiente patrón:

$$P_2 \text{ FILTER } (x_1 \neq x_2)$$

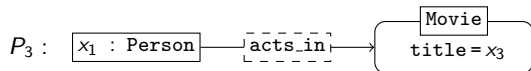
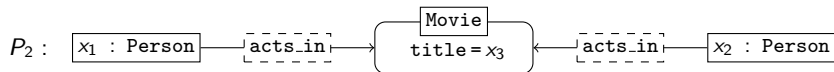
Una forma más general de negación: ejemplo

Considere los patrones:



Una forma más general de negación: ejemplo

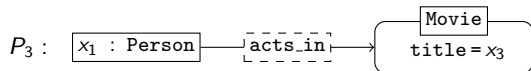
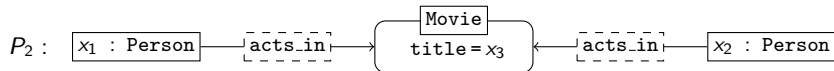
Considere los patrones:



Queremos retornar la lista de películas que sólo tienen un actor:

Una forma más general de negación: ejemplo

Considere los patrones:

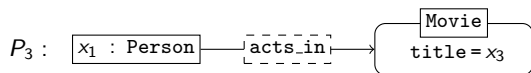
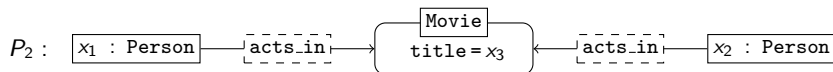


Queremos retornar la lista de películas que sólo tienen un actor:

$$P_4 = \text{SELECT } x_3 P_3$$

Una forma más general de negación: ejemplo

Considere los patrones:



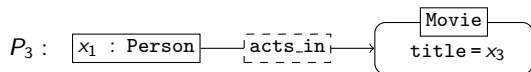
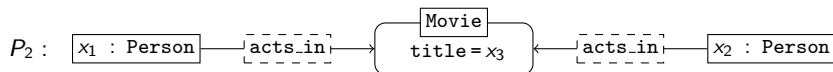
Queremos retornar la lista de películas que sólo tienen un actor:

$$P_4 = \text{SELECT } x_3 \text{ } P_3$$

$$P_5 = \text{SELECT } x_3 \text{ } (P_2 \text{ FILTER } (x_1 \neq x_2))$$

Una forma más general de negación: ejemplo

Considere los patrones:



Queremos retornar la lista de películas que sólo tienen un actor:

$$P_4 = \text{SELECT } x_3 \text{ } P_3$$

$$P_5 = \text{SELECT } x_3 \text{ (} P_2 \text{ FILTER (} x_1 \neq x_2 \text{))}$$

$$P_6 = (P_4 \text{ MINUS } P_5)$$

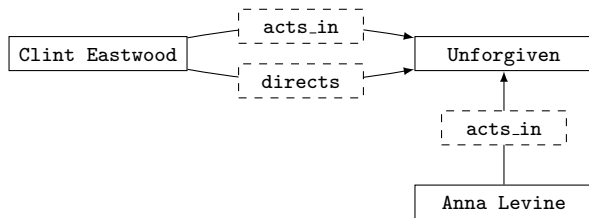
Un tercer ingrediente: construcción de caminos en grafos

Los caminos son fundamentales para los modelos de datos de grafos.

Su utilización distingue a las bases de datos de grafos de las bases de datos relacionales.

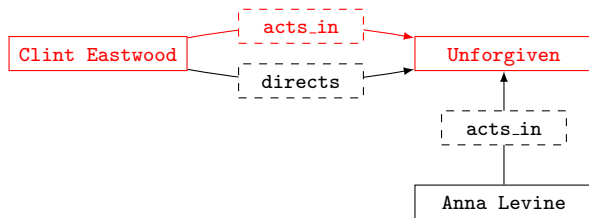
Caminos en un grafo

Considere un grafo con arcos etiquetados:



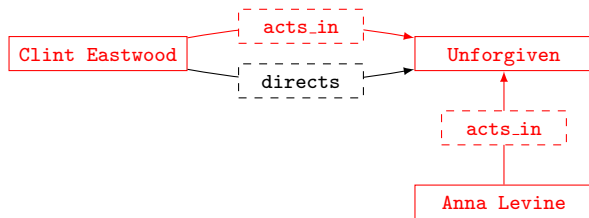
Caminos en un grafo

Considere un grafo con arcos etiquetados:



Caminos en un grafo

Considere un grafo con arcos etiquetados:



Definición de un camino

Un camino en un grafo con arcos etiquetados $G = (V, A)$ es una secuencia:

$$v_1, \ell_1, v_2, \ell_2, v_3, \dots, v_{n-1}, \ell_{n-1}, v_n$$

donde:

Definición de un camino

Un camino en un grafo con arcos etiquetados $G = (V, A)$ es una secuencia:

$$v_1, \ell_1, v_2, \ell_2, v_3, \dots, v_{n-1}, \ell_{n-1}, v_n$$

donde:

- ▶ cada $v_i \in V$

Definición de un camino

Un camino en un grafo con arcos etiquetados $G = (V, A)$ es una secuencia:

$$v_1, \ell_1, v_2, \ell_2, v_3, \dots, v_{n-1}, \ell_{n-1}, v_n$$

donde:

- ▶ cada $v_i \in V$
- ▶ para cada ℓ_i , existe una etiqueta c para arcos usada en G tal que $\ell_i = c$ o $\ell_i = c^-$

Definición de un camino

Un camino en un grafo con arcos etiquetados $G = (V, A)$ es una secuencia:

$$v_1, \ell_1, v_2, \ell_2, v_3, \dots, v_{n-1}, \ell_{n-1}, v_n$$

donde:

- ▶ cada $v_i \in V$
- ▶ para cada ℓ_i , existe una etiqueta c para arcos usada en G tal que $\ell_i = c$ o $\ell_i = c^-$
- ▶ Si $\ell_i = c$, entonces $(v_i, c, v_{i+1}) \in A$

Definición de un camino

Un camino en un grafo con arcos etiquetados $G = (V, A)$ es una secuencia:

$$v_1, \ell_1, v_2, \ell_2, v_3, \dots, v_{n-1}, \ell_{n-1}, v_n$$

donde:

- ▶ cada $v_i \in V$
- ▶ para cada ℓ_i , existe una etiqueta c para arcos usada en G tal que $\ell_i = c$ o $\ell_i = c^-$
- ▶ Si $\ell_i = c$, entonces $(v_i, c, v_{i+1}) \in A$
- ▶ Si $\ell_i = c^-$, entonces $(v_{i+1}, c, v_i) \in A$

Definición de un camino

Un camino en un grafo con arcos etiquetados $G = (V, A)$ es una secuencia:

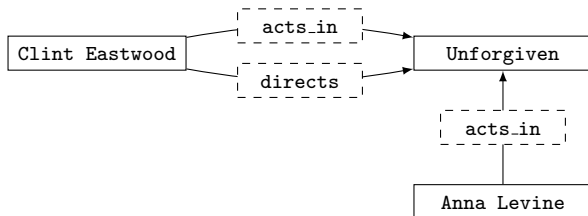
$$v_1, \ell_1, v_2, \ell_2, v_3, \dots, v_{n-1}, \ell_{n-1}, v_n$$

donde:

- ▶ cada $v_i \in V$
- ▶ para cada ℓ_i , existe una etiqueta c para arcos usada en G tal que $\ell_i = c$ o $\ell_i = c^-$
- ▶ Si $\ell_i = c$, entonces $(v_i, c, v_{i+1}) \in A$
- ▶ Si $\ell_i = c^-$, entonces $(v_{i+1}, c, v_i) \in A$

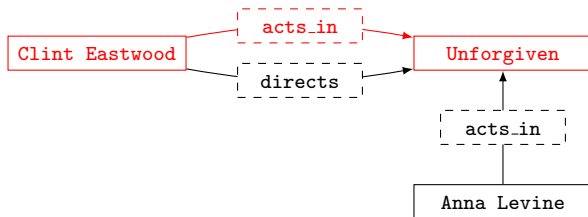
La definición es análoga para el caso de grafos con propiedades.

Caminos en un grafo: ejemplos



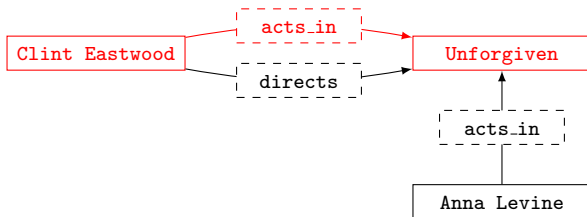
Caminos:

Caminos en un grafo: ejemplos



Caminos:

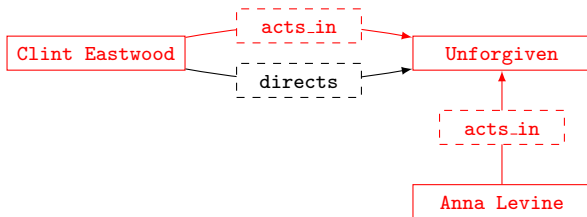
Caminos en un grafo: ejemplos



Caminos:

Clint Eastwood, acts_in, Unforgiven

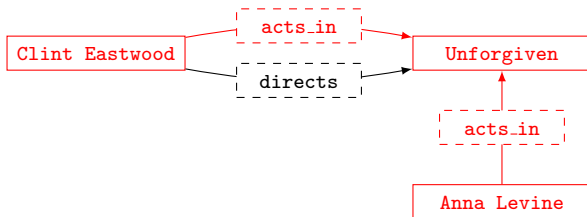
Caminos en un grafo: ejemplos



Caminos:

Clint Eastwood, acts_in, Unforgiven

Caminos en un grafo: ejemplos



Caminos:

Clint Eastwood, acts_in, Unforgiven

Clint Eastwood, acts_in, Unforgiven, acts_in⁻, Anna Levine

Patrón de navegación

La forma básica de un patrón de navegación:

$$x \xrightarrow{\alpha} y$$

En esta consulta x e y son variables, y α es una expresión regular sobre la etiquetas usadas para los arcos.

Expresiones regulares

Recuerde que una expresión regular define un conjunto de palabras sobre un alfabeto.

Expresiones regulares

Recuerde que una expresión regular define un conjunto de palabras sobre un alfabeto.

Suponiendo que el alfabeto para las expresiones regulares es $\{0, 1\}$:

Expresiones regulares

Recuerde que una expresión regular define un conjunto de palabras sobre un alfabeto.

Suponiendo que el alfabeto para las expresiones regulares es $\{0, 1\}$:

- ▶ 0 define el lenguaje $\{0\}$

Expresiones regulares

Recuerde que una expresión regular define un conjunto de palabras sobre un alfabeto.

Suponiendo que el alfabeto para las expresiones regulares es $\{0, 1\}$:

- ▶ 0 define el lenguaje $\{0\}$
- ▶ $1 \cdot 1$ define el lenguaje $\{11\}$

Expresiones regulares

Recuerde que una expresión regular define un conjunto de palabras sobre un alfabeto.

Suponiendo que el alfabeto para las expresiones regulares es $\{0, 1\}$:

- ▶ 0 define el lenguaje $\{0\}$
- ▶ $1 \cdot 1$ define el lenguaje $\{11\}$
- ▶ $0 + 1$ define el lenguaje $\{0, 1\}$

Expresiones regulares

Recuerde que una expresión regular define un conjunto de palabras sobre un alfabeto.

Suponiendo que el alfabeto para las expresiones regulares es $\{0, 1\}$:

- ▶ 0 define el lenguaje $\{0\}$
- ▶ $1 \cdot 1$ define el lenguaje $\{11\}$
- ▶ $0 + 1$ define el lenguaje $\{0, 1\}$
- ▶ $(0 \cdot 0 + 1 \cdot 1) \cdot (0 + 1)$ define el lenguaje $\{000, 001, 110, 111\}$

Expresiones regulares

Recuerde que una expresión regular define un conjunto de palabras sobre un alfabeto.

Suponiendo que el alfabeto para las expresiones regulares es $\{0, 1\}$:

- ▶ 0 define el lenguaje $\{0\}$
- ▶ $1 \cdot 1$ define el lenguaje $\{11\}$
- ▶ $0 + 1$ define el lenguaje $\{0, 1\}$
- ▶ $(0 \cdot 0 + 1 \cdot 1) \cdot (0 + 1)$ define el lenguaje $\{000, 001, 110, 111\}$
- ▶ 0^* define el lenguaje $\{\varepsilon, 0, 00, 000, 0000, 00000, \dots\}$

Expresiones regulares

Recuerde que una expresión regular define un conjunto de palabras sobre un alfabeto.

Suponiendo que el alfabeto para las expresiones regulares es $\{0, 1\}$:

- ▶ 0 define el lenguaje $\{0\}$
- ▶ $1 \cdot 1$ define el lenguaje $\{11\}$
- ▶ $0 + 1$ define el lenguaje $\{0, 1\}$
- ▶ $(0 \cdot 0 + 1 \cdot 1) \cdot (0 + 1)$ define el lenguaje $\{000, 001, 110, 111\}$
- ▶ 0^* define el lenguaje $\{\varepsilon, 0, 00, 000, 0000, 00000, \dots\}$
- ▶ $(0 \cdot 0 + 1 \cdot 1)^*$ define el lenguaje $\{\varepsilon, 00, 11, 0000, 0011, 1100, 1111, \dots\}$

Patrón de navegación: definición

Un camino $v_1, \ell_1, v_2, \dots, v_{n-1}, \ell_{n-1}, v_n$ satisface a la expresión regular α si la palabra $\ell_1 \cdots \ell_{n-1}$ pertenece al lenguaje definido por α

Patrón de navegación: definición

Un camino $v_1, \ell_1, v_2, \dots, v_{n-1}, \ell_{n-1}, v_n$ satisface a la expresión regular α si la palabra $\ell_1 \cdots \ell_{n-1}$ pertenece al lenguaje definido por α

- ▶ α puede contener expresiones de la forma c^-

Patrón de navegación: definición

Un camino $v_1, \ell_1, v_2, \dots, v_{n-1}, \ell_{n-1}, v_n$ satisface a la expresión regular α si la palabra $\ell_1 \cdots \ell_{n-1}$ pertenece al lenguaje definido por α

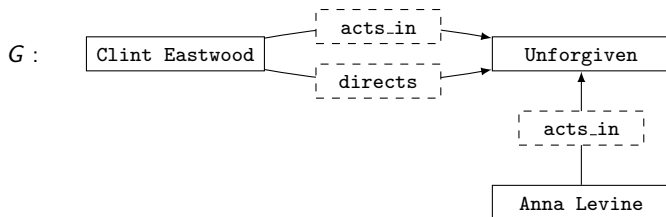
- ▶ α puede contener expresiones de la forma c^-

Si $v_1, \ell_1, v_2, \dots, v_{n-1}, \ell_{n-1}, v_n$ satisface a α , entonces la siguiente es una respuesta válida para el patrón de navegación $x \xrightarrow{\alpha} y$:

$x \mapsto v_1$

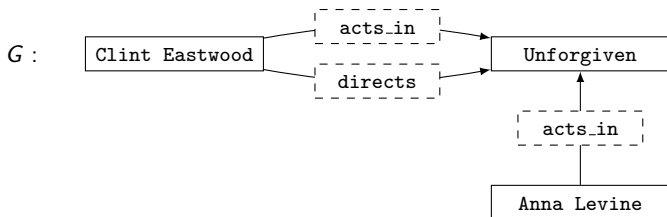
$y \mapsto v_n$

Patrones de navegación: un ejemplo



Si P es el patrón de navegación $x \xrightarrow{\text{acts_in} \cdot \text{acts_in}^-} y$:

Patrones de navegación: un ejemplo



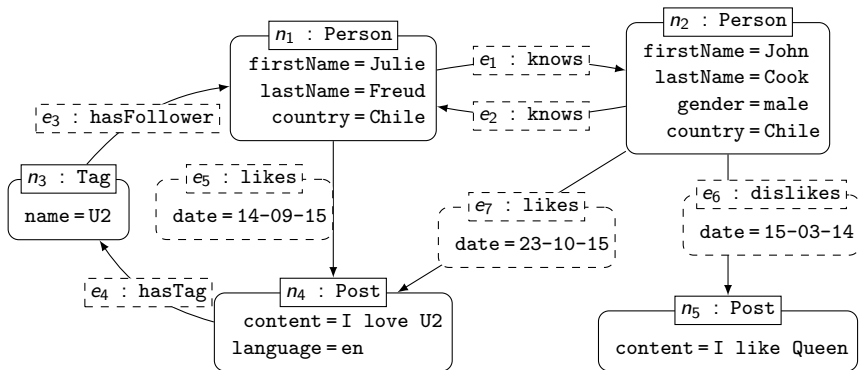
Si P es el patrón de navegación $x \xrightarrow{\text{acts_in} \cdot \text{acts_in}^-} y$:

$P(G)$:

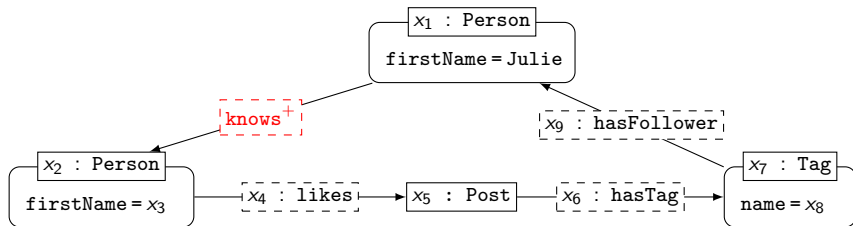
x	y
Clint Eastwood	Anna Levine
Anna Levine	Clint Eastwood
Clint Eastwood	Clint Eastwood
Anna Levine	Anna Levine

Patrones de navegación y patrones básicos

Considere nuevamente el siguiente grafo con propiedades:



Patrones de navegación y patrones básicos



¿Qué debe retornar un patrón de navegación?

En la semántica definida anteriormente un patrón de navegación retorna un conjunto de mappings.

- ▶ Donde cada mapping tiene como dominio un conjunto de dos variables

¿Qué debe retornar un patrón de navegación?

En la semántica definida anteriormente un patrón de navegación retorna un conjunto de mappings.

- ▶ Donde cada mapping tiene como dominio un conjunto de dos variables

Si extendemos a los patrones básicos con expresiones regulares vamos a seguir obteniendo conjuntos de mappings como salida

¿Qué debe retornar un patrón de navegación?

En la semántica definida anteriormente un patrón de navegación retorna un conjunto de mappings.

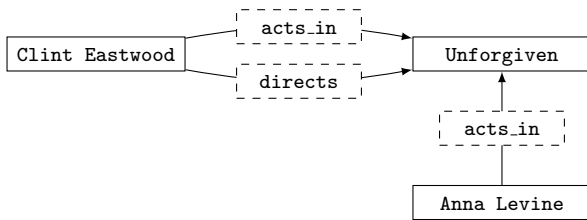
- ▶ Donde cada mapping tiene como dominio un conjunto de dos variables

Si extendemos a los patrones básicos con expresiones regulares vamos a seguir obteniendo conjuntos de mappings como salida

¡Es importante que el lenguaje también pueda retornar caminos!

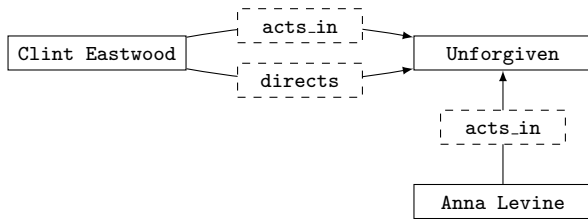
Retornando caminos

¿Cuántos caminos retorna el patrón de navegación $x \xrightarrow{(\text{acts_in} \cdot \text{acts_in}^-)^*} y$ en el siguiente grafo?



Retornando caminos

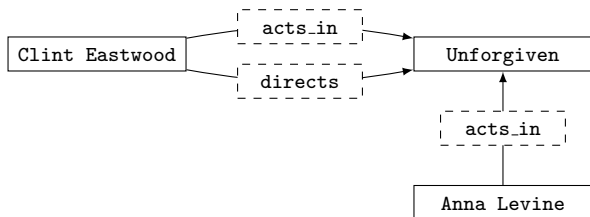
¿Cuántos caminos retorna el patrón de navegación $x \xrightarrow{(\text{acts_in} \cdot \text{acts_in}^-)^*} y$ en el siguiente grafo?



Existe una cantidad infinita de caminos en G que satisfacen al patrón.

Retornando caminos

¿Cuántos caminos retorna el patrón de navegación $x \xrightarrow{(\text{acts_in} \cdot \text{acts_in}^-)^*} y$ en el siguiente grafo?



Existe una cantidad infinita de caminos en G que satisfacen al patrón.

- ▶ ¿Qué retornamos entonces? ¿Cuáles caminos deberíamos considerar?

Semánticas alternativas para caminos

Los caminos arbitrarios son una alternativa simple y natural al momento de retornar caminos.

- ▶ Pero podemos tener un número infinito de caminos arbitrarios que satisfacen un patrón de navegación

Semánticas alternativas para caminos

Los caminos arbitrarios son una alternativa simple y natural al momento de retornar caminos.

- ▶ Pero podemos tener un número infinito de caminos arbitrarios que satisfacen un patrón de navegación

¿Qué semánticas alternativas han sido consideradas al retornar caminos?

Semánticas alternativas para caminos

Los caminos arbitrarios son una alternativa simple y natural al momento de retornar caminos.

- ▶ Pero podemos tener un número infinito de caminos arbitrarios que satisfacen un patrón de navegación

¿Qué semánticas alternativas han sido consideradas al retornar caminos?

- ▶ Caminos más cortos

Semánticas alternativas para caminos

Los caminos arbitrarios son una alternativa simple y natural al momento de retornar caminos.

- ▶ Pero podemos tener un número infinito de caminos arbitrarios que satisfacen un patrón de navegación

¿Qué semánticas alternativas han sido consideradas al retornar caminos?

- ▶ Caminos más cortos
- ▶ Caminos simples sin arcos repetidos

Semánticas alternativas para caminos

Los caminos arbitrarios son una alternativa simple y natural al momento de retornar caminos.

- ▶ Pero podemos tener un número infinito de caminos arbitrarios que satisfacen un patrón de navegación

¿Qué semánticas alternativas han sido consideradas al retornar caminos?

- ▶ Caminos más cortos
- ▶ Caminos simples sin arcos repetidos
- ▶ Caminos simples sin vértices repetidos

Consecuencias sobre la complejidad de evaluación

La semánticas para patrones de navegación basadas en caminos más cortos y caminos simples (sin repeticiones de nodos o arcos) solucionan el problema de conjuntos infinitos de repuestas.

Consecuencias sobre la complejidad de evaluación

La semánticas para patrones de navegación basadas en caminos más cortos y caminos simples (sin repeticiones de nodos o arcos) solucionan el problema de conjuntos infinitos de repuestas.

Las semánticas basadas en caminos simples son utilizadas en la práctica, pero tienen un problema de complejidad al ser evaluadas.

Consecuencias sobre la complejidad de evaluación

Considere el patrón de navegación $P = x_1 \xrightarrow{(\text{knows} \cdot \text{knows})^*} x_2$ y el siguiente problema de decisión:

$$\text{PAR} = \{(G, v_1, v_2) \mid (v_1, v_2) \text{ está en } P(G)\}$$

Teorema

Consecuencias sobre la complejidad de evaluación

Considere el patrón de navegación $P = x_1 \xrightarrow{(\text{knows} \cdot \text{knows})^*} x_2$ y el siguiente problema de decisión:

$$\text{PAR} = \{(G, v_1, v_2) \mid (v_1, v_2) \text{ está en } P(G)\}$$

Teorema

- ▶ *Para las semánticas basadas en caminos arbitrarios y caminos más cortos, PAR puede ser resuelto en tiempo lineal ($|G|$)*

Consecuencias sobre la complejidad de evaluación

Considere el patrón de navegación $P = x_1 \xrightarrow{(\text{knows} \cdot \text{knows})^*} x_2$ y el siguiente problema de decisión:

$$\text{PAR} = \{(G, v_1, v_2) \mid (v_1, v_2) \text{ está en } P(G)\}$$

Teorema

- ▶ *Para las semánticas basadas en caminos arbitrarios y caminos más cortos, PAR puede ser resuelto en tiempo lineal ($|G|$)*
- ▶ *Para las semánticas basadas en caminos simples, PAR es NP-completo*

Para recordar

- ▶ Un grafo puede verse como un modelo de datos
 - ▶ Dos alternativas: grafos con arcos etiquetados y grafos con propiedades

Para recordar

- ▶ Un grafo puede verse como un modelo de datos
 - ▶ Dos alternativas: grafos con arcos etiquetados y grafos con propiedades
- ▶ Queremos tener lenguajes de consulta declarativos para extraer información desde un grafo

Para recordar

- ▶ Un grafo puede verse como un modelo de datos
 - ▶ Dos alternativas: grafos con arcos etiquetados y grafos con propiedades
- ▶ Queremos tener lenguajes de consulta declarativos para extraer información desde un grafo
- ▶ Componentes comunes de los lenguajes de consulta para grafos:

Para recordar

- ▶ Un grafo puede verse como un modelo de datos
 - ▶ Dos alternativas: grafos con arcos etiquetados y grafos con propiedades
- ▶ Queremos tener lenguajes de consulta declarativos para extraer información desde un grafo
- ▶ Componentes comunes de los lenguajes de consulta para grafos:
 - ▶ Patrón básico: bloque *mínimo* para extraer información desde un grafo

Para recordar

- ▶ Un grafo puede verse como un modelo de datos
 - ▶ Dos alternativas: grafos con arcos etiquetados y grafos con propiedades
- ▶ Queremos tener lenguajes de consulta declarativos para extraer información desde un grafo
- ▶ Componentes comunes de los lenguajes de consulta para grafos:
 - ▶ Patrón básico: bloque *mínimo* para extraer información desde un grafo
 - ▶ Patrón complejo: combinación de patrones básicos a través de operadores

Para recordar

- ▶ Un grafo puede verse como un modelo de datos
 - ▶ Dos alternativas: grafos con arcos etiquetados y grafos con propiedades
- ▶ Queremos tener lenguajes de consulta declarativos para extraer información desde un grafo
- ▶ Componentes comunes de los lenguajes de consulta para grafos:
 - ▶ Patrón básico: bloque *mínimo* para extraer información desde un grafo
 - ▶ Patrón complejo: combinación de patrones básicos a través de operadores
 - ▶ Patrón de navegación: permite extraer caminos desde un grafo, y puede combinarse con los patrones básicos y complejos.