

Paradigmas Avanzados de Programación

Ejemplo de programación relacional

Juan Francisco Díaz Frias

Maestría en Ingeniería, Énfasis en Ingeniería de Sistemas y Computación
Escuela de Ingeniería de Sistemas y Computación,
home page: <http://eisc.univalle.edu.co>
Universidad del Valle - Cali, Colombia

Plan

- 1 Bases de datos relacionales
 - La clase Relación
 - Construyendo grafos
 - Consultas sobre grafos
 - Implementación de la clase Relación

Bases de datos

Definición

Una base de datos es una colección de datos que cuenta con una estructura bien definida.

BD Relacional

Cuando se consideran los datos como un conjunto de relaciones, donde una relación es un conjunto de tuplas, se llaman bases de datos relacionales.

Ejemplo: un grafo

```
arista(1 2)  arista(2 1)  arista(2 3)  arista(3 4)
arista(2 5)  arista(5 6)  arista(4 6)  arista(6 7)
arista(6 8)  arista(1 5)  arista(5 1)
```

Una base de datos relacional almacena explícitamente estas tuplas, de forma que puede calcular con ellas.

Bases de datos

Definición

Una base de datos es una colección de datos que cuenta con una estructura bien definida.

BD Relacional

Cuando se consideran los datos como un conjunto de relaciones, donde una relación es un conjunto de tuplas, se llaman bases de datos relacionales.

Ejemplo: un grafo

```
arista(1 2)  arista(2 1)  arista(2 3)  arista(3 4)  
arista(2 5)  arista(5 6)  arista(4 6)  arista(6 7)  
arista(6 8)  arista(1 5)  arista(5 1)
```

Una base de datos relacional almacena explícitamente estas tuplas, de forma que puede calcular con ellas.

Bases de datos

Definición

Una base de datos es una colección de datos que cuenta con una estructura bien definida.

BD Relacional

Cuando se consideran los datos como un conjunto de relaciones, donde una relación es un conjunto de tuplas, se llaman bases de datos relacionales.

Ejemplo: un grafo

```
arista(1 2)  arista(2 1)  arista(2 3)  arista(3 4)  
arista(2 5)  arista(5 6)  arista(4 6)  arista(6 7)  
arista(6 8)  arista(1 5)  arista(5 1)
```

Una base de datos relacional almacena explícitamente estas tuplas, de forma que puede calcular con ellas.

Bases de datos

Operaciones típicas sobre una base de datos relacional

- Una consulta es mucho más que una simple lectura, pues es una fórmula lógica cuyos elementos básicos son las relaciones en la base de datos. Es responsabilidad del SDBD el encontrar todas las tuplas que satisfacen la fórmula.
- Una actualización significa adicionar información en la base de datos. Esta información debe ser del tipo correcto y no perturbar la organización de la base de datos. La actualización se implementa normalmente como una transacción.

Bases de datos

Operaciones típicas sobre una base de datos relacional

- Una consulta es mucho más que una simple lectura, pues es una fórmula lógica cuyos elementos básicos son las relaciones en la base de datos. Es responsabilidad del SDBD el encontrar todas las tuplas que satisfacen la fórmula.
- Una actualización significa adicionar información en la base de datos. Esta información debe ser del tipo correcto y no perturbar la organización de la base de datos. La actualización se implementa normalmente como una transacción.

Bases de datos

Operaciones típicas sobre una base de datos relacional

- Una consulta es mucho más que una simple lectura, pues es una fórmula lógica cuyos elementos básicos son las relaciones en la base de datos. Es responsabilidad del SDBD el encontrar todas las tuplas que satisfacen la fórmula.
- Una actualización significa adicionar información en la base de datos. Esta información debe ser del tipo correcto y no perturbar la organización de la base de datos. La actualización se implementa normalmente como una transacción.

Bases de datos

Prog. relacional y BD relacionales

- La programación relacional no impone restricciones sobre la forma lógica de la consulta. Aún si la consulta es altamente disyuntiva (tiene muchas alternativas), ésta será calculada correctamente (aunque lentamente).
- La programación relacional permite la experimentación con bases de datos deductivas.

Una base de datos deductiva es una base de datos cuya implementación puede deducir tuplas adicionales que no están almacenadas explícitamente.

Bases de datos

Prog. relacional y BD relacionales

- La programación relacional no impone restricciones sobre la forma lógica de la consulta. Aún si la consulta es altamente disyuntiva (tiene muchas alternativas), ésta será calculada correctamente (aunque lentamente).

- La programación relacional permite la experimentación con bases de datos deductivas.

Una base de datos deductiva es una base de datos cuya implementación puede deducir tuplas adicionales que no están almacenadas explícitamente.

Bases de datos

Prog. relacional y BD relacionales

- La programación relacional no impone restricciones sobre la forma lógica de la consulta. Aún si la consulta es altamente disyuntiva (tiene muchas alternativas), ésta será calculada correctamente (aunque lentamente).
- La programación relacional permite la experimentación con bases de datos deductivas.

Una base de datos deductiva es una base de datos cuya implementación puede deducir tuplas adicionales que no están almacenadas explícitamente.

Plan

- 1 Bases de datos relacionales
 - La clase Relación
 - Construyendo grafos
 - Consultas sobre grafos
 - Implementación de la clase Relación

Bases de datos

Definiendo una relación

- Una relación nueva es una instancia de `ClaseRelación`, e.g., `Rel={New ClaseRelación inic}` crea la relación `Rel`, inicialmente vacía.
- Las operaciones posibles son las siguientes:
 - `{Rel afirmar(T)}` agrega la tupla `T` a `Rel`. Esta operación sólo puede realizarse por fuera de un programa relacional.
 - `{Rel afirmartodos(Ts)}` agrega la lista de tuplas `Ts` a `Rel`. Esta operación sólo puede realizarse por fuera de un programa relacional.
 - `{Rel consulta(X)}` liga `X` a una de la tuplas de `Rel`. `X` puede ser un valor parcial. Si más de una tupla es compatible con `X`, entonces la búsqueda puede enumerarlas a todas. Esta operación sólo se puede realizar dentro de un programa relacional.

Bases de datos

Definiendo una relación

- Una relación nueva es una instancia de `ClaseRelación`, e.g., `Rel={New ClaseRelación inic}` crea la relación `Rel`, inicialmente vacía.
- Las operaciones posibles son las siguientes:
 - `{Rel afirmar(T)}` agrega la tupla `T` a `Rel`. Esta operación sólo puede realizarse por fuera de un programa relacional.
 - `{Rel afirmartodos(Ts)}` agrega la lista de tuplas `Ts` a `Rel`. Esta operación sólo puede realizarse por fuera de un programa relacional.
 - `{Rel consulta(X)}` liga `X` a una de la tuplas de `Rel`. `X` puede ser un valor parcial. Si más de una tupla es compatible con `X`, entonces la búsqueda puede enumerarlas a todas. Esta operación sólo se puede realizar dentro de un programa relacional.

Bases de datos

Definiendo una relación

- Una relación nueva es una instancia de `ClaseRelación`, e.g., `Rel={New ClaseRelación inic}` crea la relación `Rel`, inicialmente vacía.
- Las operaciones posibles son las siguientes:
 - `{Rel afirmar(T)}` agrega la tupla `T` a `Rel`. Esta operación sólo puede realizarse por fuera de un programa relacional.
 - `{Rel afirmartodos(Ts)}` agrega la lista de tuplas `Ts` a `Rel`. Esta operación sólo puede realizarse por fuera de un programa relacional.
 - `{Rel consulta(X)}` liga `X` a una de la tuplas de `Rel`. `X` puede ser un valor parcial. Si más de una tupla es compatible con `X`, entonces la búsqueda puede enumerarlas a todas. Esta operación sólo se puede realizar dentro de un programa relacional.

Bases de datos

Definiendo una relación

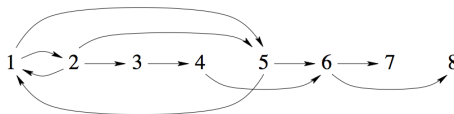
- Una relación nueva es una instancia de `ClaseRelación`, e.g., `Rel={New ClaseRelación inic}` crea la relación `Rel`, inicialmente vacía.
- Las operaciones posibles son las siguientes:
 - `{Rel afirmar(T)}` agrega la tupla `T` a `Rel`. Esta operación sólo puede realizarse por fuera de un programa relacional.
 - `{Rel afirmartodos(Ts)}` agrega la lista de tuplas `Ts` a `Rel`. Esta operación sólo puede realizarse por fuera de un programa relacional.
 - `{Rel consulta(X)}` liga `X` a una de la tuplas de `Rel`. `X` puede ser un valor parcial. Si más de una tupla es compatible con `X`, entonces la búsqueda puede enumerarlas a todas. Esta operación sólo se puede realizar dentro de un programa relacional.

Plan

- 1 Bases de datos relacionales
 - La clase Relación
 - Construyendo grafos
 - Consultas sobre grafos
 - Implementación de la clase Relación

Bases de datos

Un ejemplo: el grafo



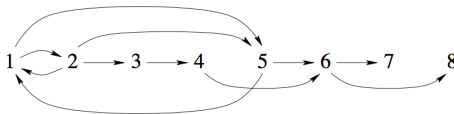
Cómo construirlo

```

RelNodos={New ClaseRelación inic}
{RelNodos
  afirmartodos([nodo(1) nodo(2) nodo(3) nodo(4)
                nodo(5) nodo(6) nodo(7) nodo(8)])}
RelAristas={New ClaseRelación inic}
{RelAristas
  afirmartodos([arista(1 2) arista(2 1) arista(2 3)
                arista(3 4) arista(2 5) arista(5 6)
                arista(4 6) arista(6 7) arista(6 8)
                arista(1 5) arista(5 1)])}
  
```

Bases de datos

Un ejemplo: el grafo



Cómo construirlo

```

RelNodos={New ClaseRelación inic}
{RelNodos
  afirmartodos([nodo(1) nodo(2) nodo(3) nodo(4)
                nodo(5) nodo(6) nodo(7) nodo(8)])}
RelAristas={New ClaseRelación inic}
{RelAristas
  afirmartodos([arista(1 2) arista(2 1) arista(2 3)
                arista(3 4) arista(2 5) arista(5 6)
                arista(4 6) arista(6 7) arista(6 8)
                arista(1 5) arista(5 1)])}
  
```

Plan

- 1 Bases de datos relacionales
 - La clase Relación
 - Construyendo grafos
 - Consultas sobre grafos
 - Implementación de la clase Relación

Bases de datos

Para consultar...

```
proc {PNode A} {RelNodos consulta(nodo(A))} end
proc {PArista A B} {RelAristas consulta(arista(A B))} end
```

Algunas consultas ...

```
proc {Q ?X} {PArista 1 X} end

proc {Q2 ?X} A B C D in
  {PArista A B} A<B=true
  {PArista B C} B<C=true
  {PArista C D} C<D=true
  X=camino(A B C D)
end
```

Bases de datos

Para consultar...

```
proc {PNode A} {RelNodos consulta(nodo(A))} end  
proc {PArista A B} {RelAristas consulta(arista(A B))} end
```

Algunas consultas ...

```
proc {Q ?X} {PArista 1 X} end  
  
proc {Q2 ?X} A B C D in  
  {PArista A B} A<B=true  
  {PArista B C} B<C=true  
  {PArista C D} C<D=true  
  X=camino(A B C D)  
end
```

Bases de datos

Consultar los caminos ...

Un camino es una secuencia de nodos tales que existe una arista entre cada nodo y su sucesor y ningún nodo aparece más de una vez.

[1 2 5 6] es un camino

Relación $PCamino$

$\{PCamino\ A\ B\ Camino\}$ es cierta si $Camino$ es un camino de A a B.

Relación $PCamino2$

$\{PCamino2\ A\ B\ Rastro\ Camino\}$ es cierta si $Camino$ es un camino de A a B y $Rastro$ es la lista invertida de nodos del camino.

Bases de datos

Consultar los caminos ...

Un camino es una secuencia de nodos tales que existe una arista entre cada nodo y su sucesor y ningún nodo aparece más de una vez.

[1 2 5 6] es un camino

Relación $PCamino$

$\{PCamino \ A \ B \ Camino\}$ es cierta si $Camino$ es un camino de A a B .

Relación $PCamino2$

$\{PCamino2 \ A \ B \ Rastro \ Camino\}$ es cierta si $Camino$ es un camino de A a B y $Rastro$ es la lista invertida de nodos del camino.

Bases de datos

Consultar los caminos ...

Un camino es una secuencia de nodos tales que existe una arista entre cada nodo y su sucesor y ningún nodo aparece más de una vez.

[1 2 5 6] es un camino

Relación $PCamino$

$\{PCamino \ A \ B \ Camino\}$ es cierta si $Camino$ es un camino de A a B .

Relación $PCamino2$

$\{PCamino2 \ A \ B \ Rastro \ Camino\}$ es cierta si $Camino$ es un camino de A a B y $Rastro$ es la lista invertida de nodos del camino.

Bases de datos

Implementación PCamino

```

proc {PCamino ?A ?B ?Camino}
  {PNode A}
  {PCamino2 A B [A] Camino}
end

proc {PCamino2 ?A ?B Rastro ?Camino}
  choice
    A=B
    Camino={Reverse Rastro}
  [] C in
    {PArista A C}
    {Member C Rastro}=false
    {PCamino2 C B C|Rastro Camino}
  end
end

```

Plan

- 1 Bases de datos relacionales
 - La clase Relación
 - Construyendo grafos
 - Consultas sobre grafos
 - Implementación de la clase Relación

Bases de datos

Implementación relaciones (1)

- Se utiliza un diccionario para almacenar las tuplas.
- Se hace indexación por el primer argumento la cual se implementa utilizando una operación nueva, `IsDet`.
- La declaración `choice` se utiliza en el procedimiento `Escoger`, el cual escoge consecutivamente todos los elementos de una lista.

Bases de datos

Implementación relaciones (1)

- Se utiliza un diccionario para almacenar las tuplas.
- Se hace indexación por el primer argumento la cual se implementa utilizando una operación nueva, `IsDet`.
- La declaración `choice` se utiliza en el procedimiento `Escoger`, el cual escoge consecutivamente todos los elementos de una lista.

Bases de datos

Implementación relaciones (1)

- Se utiliza un diccionario para almacenar las tuplas.
- Se hace indexación por el primer argumento la cual se implementa utilizando una operación nueva, `IsDet`.
- La declaración `choice` se utiliza en el procedimiento `Escoger`, el cual escoge consecutivamente todos los elementos de una lista.

Bases de datos

Implementación relaciones (2)

```

class ClaseRelación
  attr d
  meth inic
    d:={NewDictionary}
  end
  meth afirmartodos(Is)
    for I in Is do {self afirmar(I)} end
  end
  meth afirmar(I)
    if {IsDet I.1} then
      Is={Dictionary.condGet @d I.1 nil} in
      {Dictionary.put @d I.1 {Append Is [I]}}
    else
      raise databaseError(nonground(I)) end
    end
  end
  ...
end

```

Bases de datos

Implementación relaciones (3)

```

class ClaseRelación
...
  meth consulta(I)
    if {IsDet I} andthen {IsDet I.1} then
      {Escoger I {Dictionary.condGet @d
                                     I.1 nil}}
    else
      {Escoger I {Flatten
                  {Dictionary.items @d}}}
    end
  end
end
proc {Escoger ?X Ys}
  choice Ys=X|_
  [] Yr in Ys=_|Yr {Escoger X Yr} end
end

```