



Trabajo Práctico 4

Algoritmos de Normalización

Cierre de un conjunto de dependencias funcionales F^+

Sea F un conjunto de dependencias funcionales. El cierre de F , denotado por F^+ , es el conjunto de todas las dependencias funcionales implicadas lógicamente en F .

Procedimiento para calcular F^+ escrito en pseudocódigo.

```
F+ = F
repeat
  for each dependencia funcional f de F+
    aplicar las reglas de reflexividad y de aumentatividad a f
    añadir las dependencias funcionales resultantes a F+

  for each pareja de dependencias funcionales f1 y f2 de F+
    if f1 y f2 pueden combinarse mediante la transitividad
      añadir la dependencia funcional resultante a F+

until F+ no cambie más
```

1) Implemente este algoritmo, de modo que cuando reciba una relación R y un conjunto F de dependencias funcionales, calcule y devuelva F^+ .

Cierre de conjuntos de atributos α^+

Sea α un conjunto de atributos. Al conjunto de todos los atributos determinados funcionalmente por α bajo un conjunto F de dependencias funcionales se le denomina cierre de α bajo F ; se denota mediante α^+ .

A continuación se muestra un algoritmo, escrito en pseudocódigo para calcular α^+ .

```
resultado := alfa;
while (cambios en resultado) do
  for each dependencia funcional b-> c in F do
    begin
      if b  $\subseteq$  resultado then resultado := resultado  $\cup$  c;
    end
```

Resulta que, en el peor de los casos, este algoritmo puede tardar un tiempo proporcional al cuadrado del tamaño de F .

2) Implemente este algoritmo, de modo que cuando reciba una relación R , un conjunto F de dependencias funcionales y un conjunto de atributos A , calcule y devuelva A^+ .

Conjunto de claves candidatas

3) Diseñe un algoritmo que dada una relación R y un conjunto de dependencias funcionales F devuelva el conjunto de todas sus claves candidatas, para esto puede utilizar el cierre de un conjunto de atributos que haya implementado en el punto anterior. **Importante:** no será evaluada la optimalidad de la solución propuesta (complejidad temporal, espacial, etc), pero sí que funcione correctamente en todos los casos.

Debe presentar tanto su algoritmo en pseudo-código junto con una breve explicación dentro del informe, tanto su implementación en C++.

Consideraciones generales

Para realizar este tp debe descargar el archivo “Norm.rar” de la página de la materia. En el se encuentra el código base C++ para resolver el tp, tanto como 5 sets de datos para probar los algoritmos implementados. Tener en cuenta que:

- Los archivos provistos pueden compilarse (en Linux) con `>make` y con `>make clean` se borrarán los archivos `.o` y el ejecutable creados.
- En el informe se debe copiar la salida del programa, donde se utilizarán los sets 1 y 2 para probar F^+ y los sets 3, 4 y 5 para probar A^+ y el cálculo de claves.
- Dentro del archivo `main.cpp` implementar las funciones “`calculateFplus`” que corresponde al Ej 1, “`calculateAlphaPlus`” al Ej2 y “`calculateCandidates`” al Ej3.
- Debe quedar claro que lo que se implementa es en esencia lo mismo que lo presentado aquí en pseudocódigo, de ser necesario puede incluir las aclaraciones que considere necesarias en el informe

Los algoritmos tomarán los datos de entrada desde los archivos `R.data`, donde se definirá el esquema relacional, y `F.data` que contendrá las dependencias funcionales.

Ejemplos:

```
A
B
C
D
```

`R.data`

```
A->B
C, B->A
B->A, D
```

`F.data`

El trabajo se realiza en grupos de a lo sumo tres alumnos.

Documentación a entregar:

1. Un pequeño informe donde consten las aclaraciones que considere necesarias y la explicación del algoritmo propuesto en el ejercicio 3. Además debe incluirse la salida del programa para los sets de datos propuestos.
2. Los archivos de código fuente de los algoritmos implementados.

Forma de Entrega: Entregar el informe en papel y enviar un archivo comprimido con el informe y demás archivos por Email a: diana@fceia.unr.edu.ar