Lifbdw2 – Bases de données avancées

Calcul de fermetures et de couvertures

Ce programme permet de calculer des fermetures, une couverture minimum et une couverture minimum réduite d'un ensemble de dépendances fonctionelles ainsi que le schéma normalisé.

Installation

Une version 3 minimum de python est nécessaire pour exécuter le programme. Installez tous les modules python requis comme suit :

```
python -m pip install --upgrade pip
pip install -r requirements.txt
```

Utilisation

1. ouvrir un terminal
2. dirigez-vous dans le dossier qui doit contenir io.py, calculs.py, dependency_tools.py, schema_tools.py, requirements.txt et au moins le fichier df_ex_0.txt ou plusieurs examples des ensembles de DFs et DMVs (cd /mon/projet/répertoire)
3. Pour exécuter le script, utilisez :

`python io.py [args]`

Tapez `python io.py -h` pour obtenir la sortie ci-dessous pour les arguments possibles :

```
usage: io.py [-h] [-f [FERMETURE [FERMETURE ...]]] [-fl [FERMETURE_LINEAIRE [FERMETURE_LINEAIRE ...]]] [-m] [
                    [-n NORMALISATION]
                    [DFs [DFs ...]]

positional arguments:
  DFs                   Le chemin d'accès au fichier .txt qui contient un ensemble de DFs. Il y a un ensemble
                        defaut. Vous pouvez donner plusieurs fichiers.

optional arguments:
  -h, --help            show this help message and exit
  -f [FERMETURE [FERMETURE ...]], --fermeture [FERMETURE [FERMETURE ...]]
                        la fermeture des attributs données de l'ensemble de DFs donné est calculée et affiché
  -fl [FERMETURE_LINEAIRE [FERMETURE_LINEAIRE ...]], --fermeture_lineaire [FERMETURE_LINEAIRE [FERMETURE_LINEA
                        la fermeture des attributs données de l'ensemble de DFs donné est calculée et affiché
                        l'algorithme linéaire
  -m, --couverture_minimal
                        la couverture minimal de l'ensemble de DFs donné est calculée et affichée
  -r, --couverture_minimal_reduite
                        la couverture minimal et réduite de l'ensemble de DFs donné est calculée et affichée
  -n NORMALISATION, --normalisation NORMALISATION
                        Le chemin d'accès au fichier .txt qui contient un ensemble de DMVs. Seulement s'il n'
                        ensemble de DFs donné
```

Arguments expliqués

`DFs`

exemple : `</chemin/vers/>myDfs.txt`

Les fichiers d'entrée spécifient votre ensemble de DFs. Chaque fichier correspond à un ensemble et
doit avoir un format spécifique pour que le programme fonctionne correctement.
Bien qu'il fonctionne tant que l'entrée est un fichier txt, il ne fera pas ce qu'il est censé faire si le fichier ne ressemble pas à ce qui suit :

Pour un ensemble de DF par exemple :
`{ AB -> B, C -> DF, F -> D }`

Votre fichier d'entrée doit ressembler à ceci :

```
AB B
C DF
F D
```

Il n'est pas nécessaire de spécifier ce fichier si vous voulez utiliser le jeu de paramètres par défaut du
fichier df_ex_0.txt qui contient l'ensemble:

`{ A -> B, A-> C, A -> D, CD -> E, BE -> F, ABE -> G, EG -> ABD,ABE -> G, EG -> ABD, FG -> AE, AB -> B, A -> E }`

`-f <string>` **exemple :** `-f 'AB'`

Si vous spécifiez -f, vous devez spécifier une ou plusieurs chaînes de caractères comme paramètres.
Ces chaînes représentent chacune un ensemble d'attributs. Le programme calculera la fermeture de chaque
ensemble de DFs pour chacun des ensembles d'attributs spécifiés.

`-fl <string>` **exemple :** `-fl 'AB'`

L'argument -fl calculera également la fermeture comme -f et a besoin d'au moins une chaîne de caractères
comme paramètre. Cependant, un algorithme différent est utilisé. C'est la version linéaire de l'algorithme de fermeture.

`-m`

-m n'a besoin d'aucun paramètre et calcule la couverture minimale de l'ensemble de DFs.

`-r`

-r calcule d'abord la couverture minimale et la réduit ensuite sur les côtés gauche et droit de chaque DF.

`-n` **exemple :** `-n </chemin/vers/>myDMVs.txt`

-n constitue la couverture minimale et réduite et suit ensuite les étapes nécessaires à la construction d'un schéma qui est en forme normale de Boyce Codd ou en troisième forme normale.
À cette fin, un ensemble de dépendances multi-valuées peut être spécifié en passant le fichier txt (au même format que les DF) comme paramètre. Toutefois, cela est facultatif.
Il n'est pas possible de spécifier plusieurs fichiers et il ne peut y avoir qu'un seul fichier spécifié avec les DF. Sinon, il n'est pas clair à quel ensemble de DFs les DMVs appartiennent et le programme émettra un message d'erreur.

Exemples d'exécution du programme

`python3 io.py`

Cet appel affichera les dépendances fonctionnelles qui se trouvent dans le fichier par défaut df_ex_0.txt.

`python3 io.py -f 'A' 'BEC' 'AB' 'FG'`

Cet appel calcule les fermetures A+ BEC+ AB+ et FG+ pour l'ensemble de DFs par défaut.

`python3 io.py df_ex_0.txt df_ex_1.txt -fl 'AC' -m`

Ici, 2 ensembles différentes de DF sont spécifiées. Pour les deux, on calcule la fermeture de `AC` et la couverture minimale.

`python3 io.py df_ex_2.txt -m -r -n dmv_ex_0.txt`
Pour l'ensemble de DFs dans df_ex_2.txt, la couverture minimale et la couverture minimale réduite sont affichées. En outre, le schéma normalisé est calculé avec l'ensemble de DMVs dans dmv_ex_0.txt

Affichage des résultats

En dessous se trouve le résultat produit par:

`python3 io.py df_ex_2.txt -f 'A' 'B' 'H' -m -r -n dmv_ex_0.txt`

```
_____________________________________
1. ENSEMBLE DE DFs DONNEES EN ENTREE
A -> BC
B -> C
BC -> E
D -> A
E -> AF
AEF -> D
H -> I
_____________________________________
FERMETURE(S)
A⁺ = ABCDEF
B⁺ = ABCDEF
H⁺ = HI
_____________________________________
COUVERTURE MINIMAL
A -> ABCDEF
B -> ABCDEF
D -> ABCDEF
E -> ABCDEF
H -> HI
_____________________________________
COUVERTURE MINIMAL REDUITE
A -> E
B -> A
D -> A
E -> BCDF
H -> I


_____________________________________
NORMALISATION

Ensemble de DMVs:
G ->-> HI

Schéma normalisé:
R1(A B)
R2(B D)
R3(D E)
R4(E FAC)
R5(H I)
R6(GH )
R7(GA )
```