



Rapport TP d'introduction à l'analyse formelle de concepts

Meryam RHADI

16 octobre 2022

Master 2
Génie logiciel
UE IA pour le génie logiciel

Responsable
Alexandre Bazin
Marianne Huchard

Sommaire

Titre	1
Sommaire	2
1 Création de données	3
2 Construction du treillis de concepts et la base d'implications	4
2.1 Construction du treillis de concepts	4
2.2 Construction de la base d'implications	6
3 Interprétation des résultats	7
3.1 Interprétation du résultats de treillis de concepts	7
3.2 Interprétation du résultats de la base d'implications	8

1 Création de données

Pour la création du jeu de données, j'ai choisi de travailler sur le thème du génie logiciel avec des données qui permet de faire la comparaison des logiciels wiki [2].

J'ai créé un fichier .csv, avec comme Extension **E** (*ensemble des objets appartenant au concept*) les wikis software, et comme intention **I** (*Description du concept*) des différents concepts. Tous dont la forme correspond à celle d'un contexte formel, et d'une matrice binaire.

Mon fichier est une description de 10 logiciels wiki par 7 attributs, comme montre la figure 1

	A	B	C	D	E	F	G	H	I
1		editing	export/import	Selectable	Spam	Personal	Open source	access control	
2	BlueSpice	1	1	1	1	0	1	1	
3	BookStack	1	0	0	1	1	1	1	
4	BrainKeeper	1	0	0	1	1	0	0	
5	Central Desktop	1	0	0	1	1	0	0	
6	Confluence	1	1	0	1	1	0	1	
7	ConnectedText	1	1	0	0	1	0	0	
8	DokuWiki	1	1	1	1	1	1	1	
9	FlexWiki	1	0	0	1	1	1	1	
10	FosWiki	1	1	1	1	1	1	1	
11	Gitit	1	1	1	1	1	1	1	

Figure 1. Le jeu de données

2 Construction du treillis de concepts et la base d'implications

Pour la construction du treillis de concepts et la base d'implications, j'ai utilisé le logiciel **FCA4J** [1], qui est une boîte à outils Java à des fins d'analyse de concept formelle.

2.1 Construction du treillis de concepts

Pour la construction du treillis de concepts, après avoir téléchargé FCA4J et construit mon jeu de données, j'ai exécuté les 2 commandes suivantes :

- **java -jar fca4j-cli-0.4.3.jar LATTICE monRepertoire/monFichier.csv -i CSV -s SEMICOLON -g monRepertoire/Lattice/resultatD.dot** : qui permet de construire une vue graphique textuelle du réseau de concepts simplifié, comme montre la figure 2.
- **dot -Tpdf monRepertoire/Lattice/resultatD.dot -o monRepertoire/Lattice/resultatPdf.pdf** : qui permet de produire une vue pdf du réseau, et donc notre treillis de concept, comme montre la figure 3

```
digraph G { |
    rankdir=BT;
    0 [shape=record,style=filled,fillcolor=lightblue,label="{0 (I: 1, E:
10) |editing\n|}"];
    1 [shape=record,style=filled,fillcolor=lightblue,label="{1 (I: 2, E:
6) |export/import\n|}"];
    2 [shape=record,style=filled,label="{2 (I: 6, E:
4) |Selectable\n|BlueSpice\n|}"];
    3 [shape=record,style=filled,fillcolor=lightblue,label="{3 (I: 4, E:
```

Figure 2. Extrait du fichier .dot

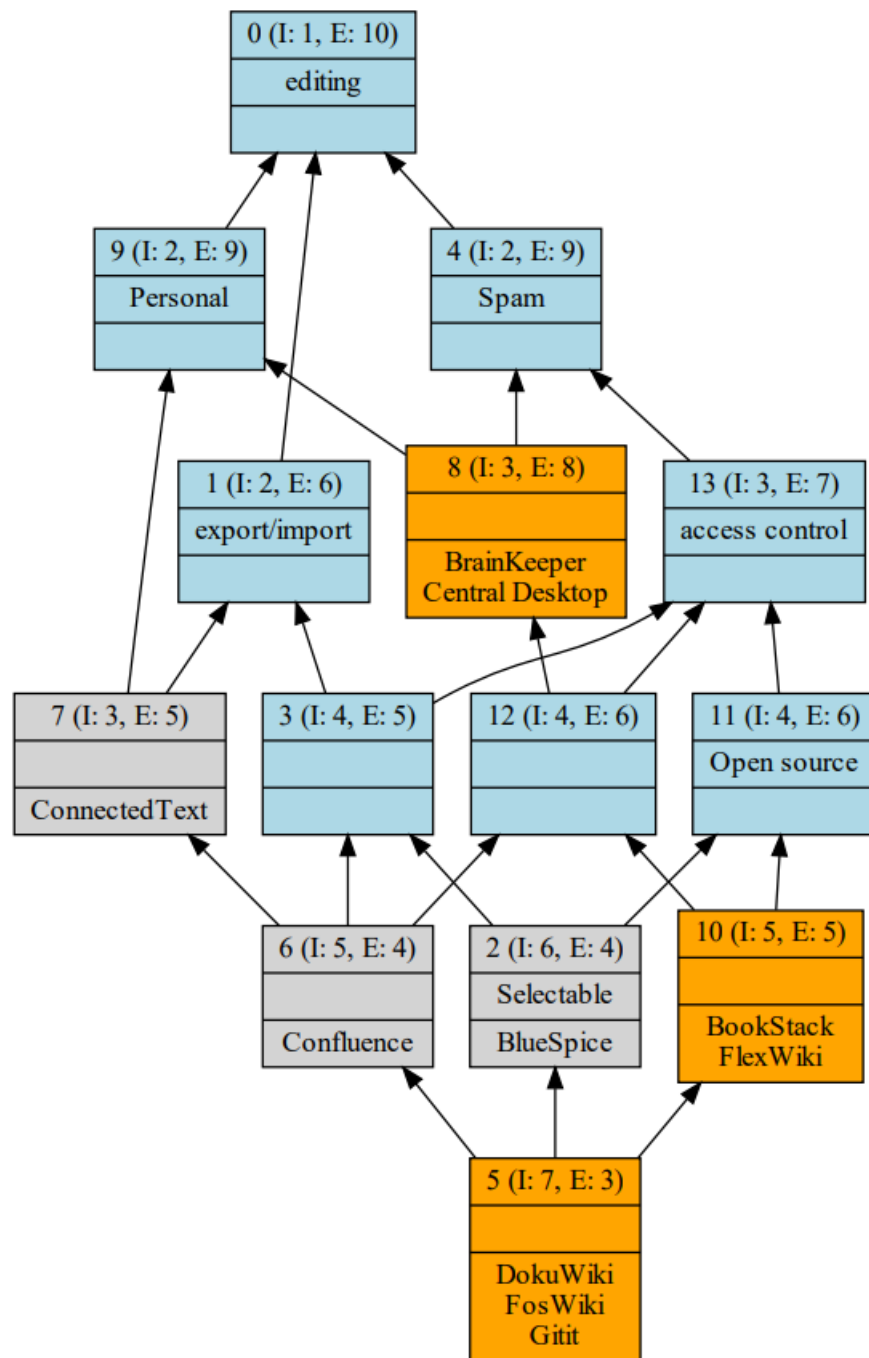


Figure 3. Treillis de conception

2.2 Construction de la base d'implications

Pour la construction de la base d'implications, j'ai exécuté la commande suivante :

- **java -jar fca4j-cli-0.4.3.jar RULEBASIS monRepertoire/monFichier.csv -i CSV -s SEMICOLON -folder ./monRepertoire/DGBI/** : qui permet de calculer la base d'implications Duquennes-Guigues (DGBI), et les règles sont stockées par le support dans différents des fichiers TXT, comme montre la figure 4, et la figure 5 montre les règles générées (la base d'implications).

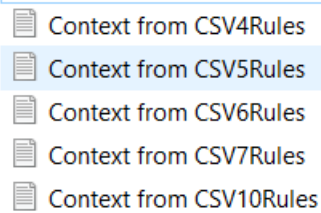


Figure 4. Les fichiers générés

```
C:\Users\SCD-UM\Downloads>java -jar fca4j-cli-0.4.3.jar RULEBASIS jeuDonne.csv -i CSV -s SEMICOLON -folder ./monRepertoire/DGBI/
running LINCOPRUNING (BITSET/BASIC/MONO) data: jeuDonne.csv ( 10 x 7 )
<10> [] => [0]
<7> [0,6] => [3]
<6> [0,5] => [3,6]
<4> [0,2] => [1,3,5,6]
<5> [0,1,3] => [6]
<4> [0,1,3,5,6] => [2]
duration: 27 ms
<10> => editing
<7> editing,access control => Spam
<6> editing,Open source => Spam ,access control
<4> editing,Selectable => export/import,Spam ,Open source,access control
<5> editing,export/import,Spam => access control
<4> editing,export/import,Spam ,Open source,access control => Selectable
```

Figure 5. la base d'implications

3 Interprétation des résultats

3.1 Interprétation du résultats de treillis de concepts

À partir du graphe généré 3, on peut déduire les interprétations suivantes :

- On a dans le premier niveau du treillis l'intention editing (I : 1), qui apparaît dans toutes nos 10 extensions (E : 10), c'est-à-dire que le concept editing est le plus grand car tout le wiki-software ont ce concept.
- Dans le deuxième niveau du treillis, pour les deux cases on a les caractéristiques (Personal et Spam), qui sont présents dans 9 de nos extensions (E:9). Donc pour nos 9 extensions, on a absolument 2 caractéristiques en commun qui sont editing + Personal ou editing + spam.
- Dans le troisième niveau du treillis, on a une case orange, donc les 3 intentions sont en commun entre les 8 extensions, et on a l'intention access control est présente dans 7 extensions , et export/import présente dans 6 extensions.
- Dans le quatrième niveau du treillis, on a 4 cases , l'intention open source est présente exactement dans les 6 extensions (Gitit, FosWiki, Doku wiki, bookStack, Flex Wiki et Blue Spice). Et les 2 autres cas bleus à côté rassemblent les cases dans le niveau bas d'eux en suivant les flèches, et la case violet à 5 extensions dont connectedText, et 3 intentions (export/import, personal, editing) .
- Dans le cinquième niveau du treillis, on a une case qui est orange, ce qui signifie que ses extensions E : 2 (BookStack et Flex-wiki) + 3(qui sont à la fin du treillis) ont les mêmes caractéristiques. Dans le même niveau on a deux cases violettes, sur une des deux cases, on a l'extension blue-space qui est caractérisé par 6 intentions parmi eux on trouve l'intention Selectable, qui existe seulement dans 4 extensions (bleu-spice et les 3 qui sont dans le niveau le plus bas). Et l'autre case violet à 5 intentions.
- Dans le dernier niveau du treillis, on a la case à une couleur orange, car les extensions de cette dernière ont les mêmes caractéristiques. Cette case est à la fin car ces extensions ont tous les caractéristiques disponibles au même temps, si on vérifie notre jeu de données on va trouver que ces 3 extensions ont 1 dans tous les cases.

3.2 Interprétation du résultats de la base d'implications

A partir de la base d'implications généré 5 , on peut déduire les interprétation suivant :

- **(=> editing)** : Rien n'implique editing, signifie que tous contient editing.
- **(editing , access control => Spam)** : Si on a editing et access control donc on a forcément spam.
- **(editing, open source => spam, access control)** : signifie que si un logiciel wiki a les caractéristiques editing et open source, alors il a inévitablement les intentions spam et access control.
- **(editing, selectable => export/import, spam, open source, access control)** : On déduit que si on a un logiciel qui possède les intentions editing et selectable, donc il a impérativement export/import, spam, open source et access control aussi.
- **(editing, export/import, spam => access control)** : On déduit que si un logiciel possède les caractéristiques editing, export/import et spam, alors il a obligatoirement access control.
- **(editing, export/import, spam, open source, access control => selectable)** : signifie que si on a editing, export/import, spam, open source et access control, donc on a forcément selectable aussi.

Bibliographie

- [1] Marianne Huchard. *FCA4J*. Lirmm. 2021. url : <https://www.lirmm.fr/fca4j/>.
- [2] Wikipedia. *Comparaison des logiciels wiki*. Wiki. 2019. url : https://en.wikipedia.org/wiki/Comparison_of_wiki_software.