UFR Mathématiques et Informatique

LORIA

Master 2 Sciences Cognitives et Médias Numériques

Graphes médians, FCA et phylogénie

Étudiant :
Baptiste Mounier

Encadrants :
Miguel Couceiro
Alain Gely
Amedeo Napoli

Référent universitaire :
Christine Bourjot

Jury :
Christine Bourjot
Miguel Couceiro
Manuel Rebuschi
Azim Roussanaly

Date :
09/04/18 - 08/08/18

Résumé

L'étude de filiation des espèces, phylogénie est un domaine de recherche utilisant des arbres parcimonieux. Par le passé, H.J. Bandelt [1] a mis en avant l'avantage qu'il y a à gagner à encoder l'ensemble des informations dans un unique graphe médian regroupant l'ensemble de ces arbres en même temps. Puis cela à été au tour de U. Priss [2] de présenter l'avantage de cette vision qui permettrait de donner à la phylogénie accès aux méthodes de l'analyse de concepts formels. De plus elle donne un point de départ de méthode permettant afin de passer d'un treillis construit à partir de la matrice de données source des arbres, en un graphe médian. Ce travail a été poursuivis et explicité en un algorithme pour les cas simple par l'équipe ORPAILLEUR[1]. Mon travail présenté dans ce dossier va porter sur le prolongement de cet algorithme pour sa mise en place dans un programme et son amélioration pour la prise en compte de cas plus complexes.

Remerciements

Avant de débuter ce rapport, il me semble judicieux de remercier tous ceux et celles qui m'ont permis de réaliser ce stage dans ces conditions.

Je commencerai par Miguel Couceiro, Alain Gely et Amedeo Napoli, mes tuteurs qui m'ont accepté au sein du projet et qui m'ont accompagné tout au long de la réalisation de ma contribution.

Je remercierai également Mme. Bourjot, ma marraine qui a effectué le suivi et l'encadrement de mon stage.

De même que l'équipe Orpailleur dans son intégralité et l'ensemble des autres stagiaires qui m'ont accueilli dans les locaux dans une joie et une bonne humeur omniprésentes, éléments essentiels d'un travail efficace au quotidien.

Et par extension le LORIA dans son ensemble pour ces mêmes raisons.

Je n'oublierai pas l'UFR Mathématiques et Informatique et ses enseignants qui m'ont apporté, tout au long de cette année d'études, les capacités, les connaissances et le recul nécessaires pour réaliser ce genre de projet.

Table des matières

1 Contexte

1.1 Laboratoire

Le Loria, Laboratoire lorrain de Recherche en Informatique et ses Applications est une Unité Mixte de Recherche (UMR 7503), commune à plusieurs établissements : le CNRS, l'Université de Lorraine et Inria.

Le laboratoire a pour mission la recherche fondamentale et appliquée en sciences informatiques et ce, depuis sa création, en 1997.

Il est membre de la Fédération Charles Hermite qui regroupe les trois principaux laboratoires de recherche en mathématiques et STIC (Science et Technologies de l'Information et de la Communication) de Lorraine. Le laboratoire fait partie du pôle scientifique AM2I (Automatique, Mathématiques, Informatique et leurs interactions) de l'Université de Lorraine.

Les travaux scientifiques sont menés au sein de 28 équipes structurées en 5 départements, dont 15 sont communes avec Inria, représentant un total de plus de 400 personnes. Le Loria est un des plus grands laboratoires de la région lorraine.

1.2 Équipe ORPAILLEUR

Le stage s'est déroulé au sein de l'équipe ORPAILLEUR et plus précisément en collaboration directe avec Miguel Couceiro, Alain Gély et Amedeo Napoli qui la dirige. Elle fait partie du 4$^{\text{ème}}$ département « Traitement automatique des langues et des connaissances ».

Le domaine de recherche de l'équipe est l'exploration de connaissances dans les bases de données (knowledge discovery in databases). Cela consiste à traiter des données afin d'en ressortir des unités de connaissances utiles et réutilisables. Le processus se fait en trois mécanismes principaux : la préparation des données, leur exploitation et l'interprétation des unités extraites en unité de connaissance.

Son nom vient de la comparaison que nous pouvons faire entre son activité et la recherche d'or. En effet en associant la présence d'or en tant qu'unités de connaissances et les données du terrain en tant que base de données, on obtient l'activité de l'équipe sur les données.

2 Motivations

Il est important avant de commencer cette section de préciser le détails des définitions et propriété sera détaillé par la suite.

2.1 Phylogénie

La phylogénie est un domaine de la biologie ayant pour but l'étude des relations de parenté entre les êtres vivants. Nous nous concentrerons sur la partie du domaine qui traite des relations entre les espèces et leurs évolutions. Pour représenter les informations, on utilise des arbres montrant les différentes espèces avec leurs caractérisques communes comme en figure 2.1 sur laquelle chaque feuille, ici représentée en noir, correspond à une espèce actuelle. Sa lecture est très simple, on part du nœud le plus haut qui correspond à l'intégralité des espèces et on descend progressivement chaque ramification qui vient sélectionner une partie des espèces suivant certaines caractéristiques. On part également du principe que chaque nœud correspond à un ancêtre commun à toutes les espèces des feuilles sur lesquelles ce nœud débouche.

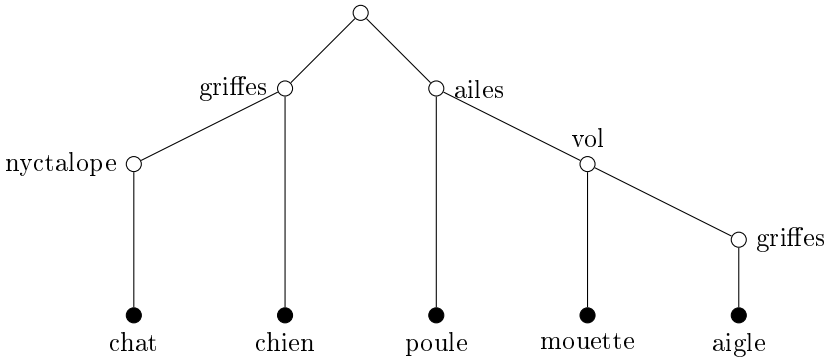


FIGURE 2.1 – Arbre phylogénétique

Ces arbres sont des représentations des données possédées et peuvent varier suivant sur quoi on cherche à mettre l'accent. Par exemple la figure 2.1 et la figure 2.2 représentent les mêmes espèces à partir des mêmes données mais avec un accent différent porté avant tout dans le cas de la figure 2.2 sur une catégorisation de la présence de griffes ou non. Ces données sont stockées dans des matrices binaires mettant en lien chaque espèce avec ses caractéristiques comme le montre la figure 2.3. Le problème est alors de choisir le bon arbre, pour diriger ce choix on utilise la notion de parcimonie qui consiste à minimiser les mutations nécessaires pour atteindre les espèces visées. Les arbres ainsi obtenus sont dit parcimonieux.

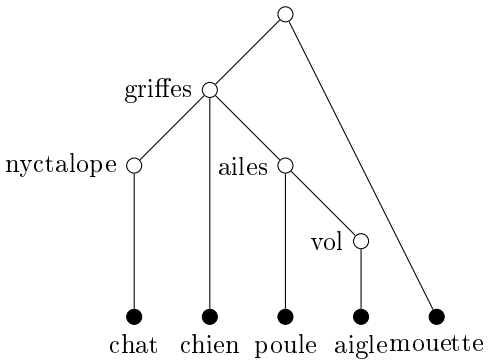


FIGURE 2.2 – Arbre phylogénétique, autre accent

	griffes	ailes	nyctalope	vol
chat	x		x	
chien	x			
aigle	x	x		x
mouette		x		x
poule	x	x		

FIGURE 2.3 – Matrice espèce/caractéristique

2.2 Outil : graphes médians

Le vivant et la multitude de caractéristiques de chaque espèce ne permet pas d'avoir un unique arbre parcimonieux pour représenter tous les accents possibles avec un jeu de données. Pour compenser cela la première solution est de donner tous les arbres possibles, solution qui est pour des raisons évidentes de place et de pertinence, pas viable. La seconde, proposée par Hans-Jürgen Bandelt dans [3], consiste à encoder l'ensemble de ces arbres dans un graphe médian. Cette méthode permet ainsi d'avoir l'intégralité de ces arbres phylogénétiques parcimonieux en une unique représentation au lieu de devoir faire un arbre par information qu'on souhaite mettre en avant. Suivant ce principe nous obtenons la figure 2.4 dans laquelle toutes les espèces, toujours sur les nœuds noirs, disposent des caractéristiques des nœuds qui lui sont au dessus et de celui où elles se trouvent [1]. Dans ce jeu de données l'espèce « chat » dispose des caractéristiques « nyctalope » et « griffes ». Ce graphe contient toutes les informations que les arbres précédents des figures 2.1 et 2.2 ainsi que les autres arbres qui peuvent être formé à partir de ce jeu de données, peuvent offrir.

définition succincte des graphes médians

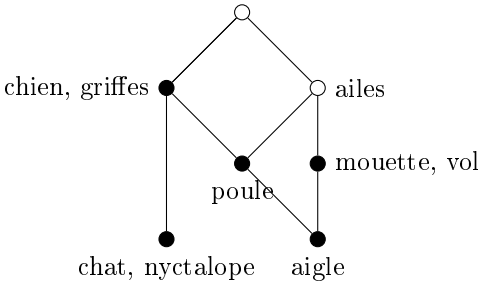


FIGURE 2.4 – Graphe médian

2.3 Approche : Analyse de Concepts Formels

Uta Priss quant à elle profite que les arbres proviennent de matrices binaires de données qui sont également à une place centrale au sein de la FCA [2] et ainsi créer des treillis qui sont des ensembles ordonnés. À partir de ces derniers nous pouvons représenter le vivant avec son propre système de classification. Elle propose dans [2] l'utilisation de treillis de concept à la place ou en complément des graphes médians et fait l'ébauche d'un algorithme pour convertir un treillis de concept en graphe médian. Créant par la même occasion entre ces deux domaines un lien qui a l'avantage d'offrir à la phylogénie les nombreux outils et la communauté relativement importante de la FCA.

Pour arriver à ce résultat, U. Priss expose une méthode consistant à rendre les treillis formés à partir des filtres des atomes [3] distributifs puis à supprimer le nœuds le plus bas, souvent noté $\perp$ dans ce rapport. Prenons en exemple un cas de base en figure 2.5 avec en nœuds noirs le sous treillis posant problème. Nous devons tout d'abord rendre les treillis formés à partir des filtres de ses atomes, distributifs, ce qui donne après opération manuelle le treillis de la figure 2.6. Puis nous supprimons le nœuds $\perp$ pour pour obtenir le graphe médian de la figure 2.7 qui valide toutes les conditions pour correspondre à un graphe médian.

1. à confirmer, graphe médian n'ont pas réellement de sens
2. Formal Concept Analysis : Analyse de Concepts Formels
3. Les atomes sont les nœuds qui sont en lien direct avec le nœud le plus bas, souvent noté $\perp$ dans ce rapport

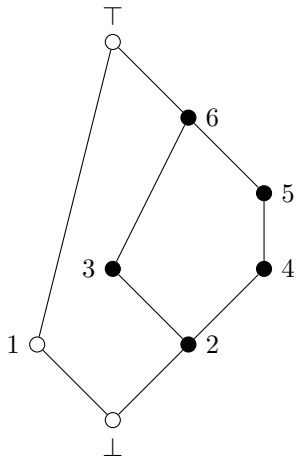


FIGURE 2.5 – Treillis de base

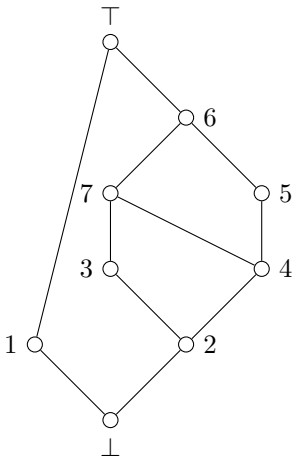


FIGURE 2.6 – Avec treillis des atomes distributifs

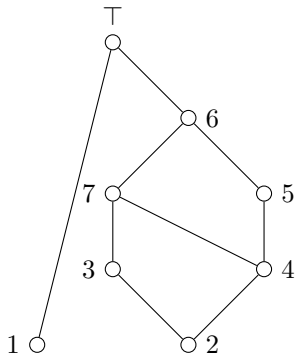


FIGURE 2.7 – Graphe médian résultat

2.4 Approche : limites

L'équipe ORPAILLEUR a poursuivi en ce sens à travers [1] et [4] en proposant une solution systématique pour la proposition de U. Priss avec ses limites. La résolution est en deux parties, tout d'abord il faut savoir transformer un treillis quelconque en treillis distributif. Pour ce faire, l'équipe a mis au point un algorithme qui permet de passer de la figure 2.8 à la figure 2.9.

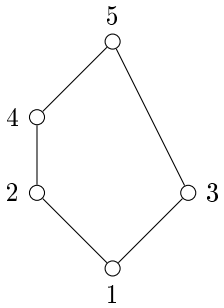


FIGURE 2.8 – N_5 avant transformation

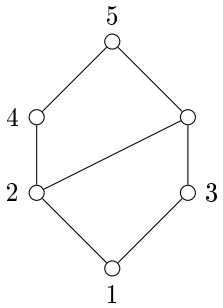


FIGURE 2.9 – N_5 après transformation

Ensuite il faut se servir de cet algorithme dans un système plus large qui doit être capable d'extraire les contextes correspondant aux treillis des atomes pour effectuer le traitement sur chacun d'eux avant d'assembler les contextes résultats en un unique contexte pour avoir un unique treillis résultat. Et une fois ces deux étapes réussies on supprime le nœud $\perp$ pour obtenir notre graphe médian.

Pour effectuer l'extraction du contexte d'un atome, il suffit d'extraire du contexte global tous les attributs qui sont en correspondance avec l'atome et de prendre tous les objets qui ont une correspondance avec l'un de

ces attributs. En prenant en exemple le contexte de la figure 2.10 dans lequel vous voulons extraire le contexte de l'atome 1 nous obtenons le contexte de la figure 2.11.

	A	B	C	D	E	F
1	x	x	x			
2	x		x			
3		x				
4			x			
5				x	x	x
6				x		x
7					x	
8						x

FIGURE 2.10 – Contexte global

C1	A	B	C
2	x		x
3		x	
4			x

FIGURE 2.11 – Contexte de l'atome 1 extrait

À l'inverse pour effectuer une remise en commun des contextes, on assemble tout simplement les contextes entre eux sans oublier d'ajouter la correspondance entre l'atome source du contexte extrait avec tous les attributs contenus dans ce contexte comme le montre les figures 2.12 et 2.13.

C1	A	B	C
2		x	x
3			x
4/7	x	x	
C4	D	E	F
5		x	x
6			x
1/7	x	x	
C7	G	G	I
8		x	x
9			x
1/4	x	x	

FIGURE 2.12 – Contextes extraits pour les atomes 1, 4 et 7

	A	B	C	D	E	F	G	H	I
1	x	x	x	x	x		x	x	
2		x	x						
3			x						
4	x	x		x	x	x	x	x	
5					x	x			
6						x			
7	x	x		x	x		x	x	x
8								x	x
9									x

FIGURE 2.13 – Contexte réassemblé

Une fois la méthode mise en place, on se rend compte qu'elle dispose de problèmes. Sur des cas moins triviaux il se pose la question de l'optimalité du treillis unique obtenu. Prenons en exemple la figure 2.14 avec en noir une partie des nœuds empêchant les treillis des atomes d'être distributif par la présence d'un sous treillis N_5. Si nous appliquons la méthode dessus nous obtenons la figure 2.15 avec également en noirs des nœuds posant le même problème.

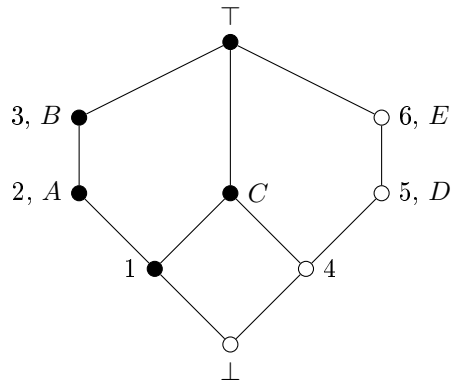


FIGURE 2.14 – Cas posant problème, situation de départ

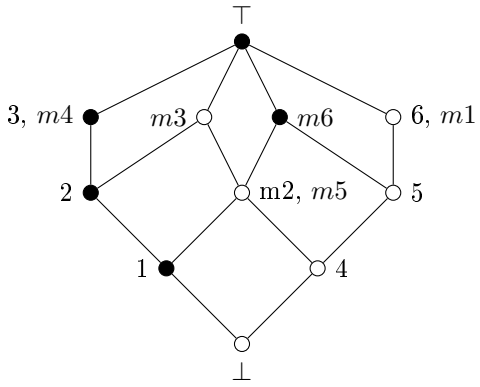


FIGURE 2.15 – Cas posant problème, situation après l'application de la méthode

La première contre mesure serait de refaire la méthode jusqu'à n'obtenir aucun sous treillis problèmatique. C'est une contre mesure fonctionnelle mais qui vient perturber l'utilité de la méthode. Nous avons besoin d'une représentation global la plus simple et proche du treillis d'origine possible. Cette méthode ne fait perdre aucune liaison et n'en ajoute uniquement pour rendre le treillis distributif, si nous la faisons en boucle, nous obtenons dans le pire des cas le treillis distributif le plus grand qui est en figure 2.16 avec en noirs les nœuds déjà présent dans le treillis d'origine. Nous pouvons voir par execution manuelle qu'une solution existe en figure 2.17 mais que la méthode ne parvient pas à atteindre. Le stage a consisté en la modification de la méthode afin de trouver cette solution optimale et de rechercher d'autres potentiels cas problématiques pour les prendre en considération.

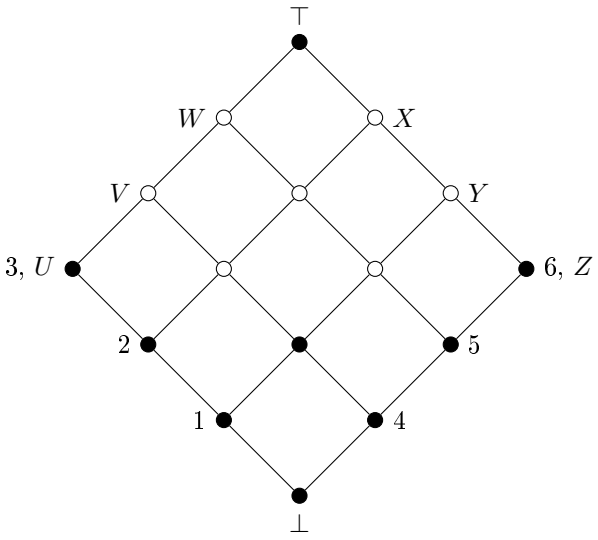


FIGURE 2.16 – Cas posant problème, situation maximale

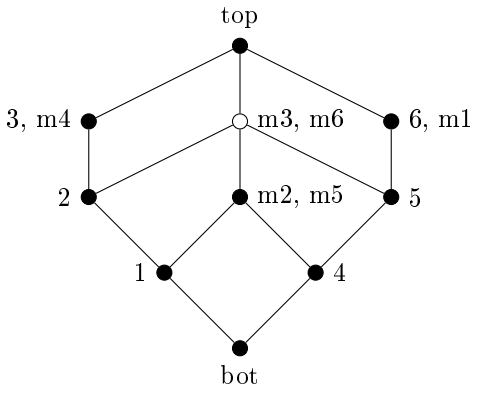


FIGURE 2.17 – Cas posant problème, situation souhaitée

Nous allons poursuivre ce document par un chapitre de définitions sur les différents éléments qui sont nécessaires à la compréhension de la problèmatique. À la suite duquel nous développeront ma contribution à la résolution de cette problèmatique.

3 Notions nécessaires

3.1 Analyse de Concepts Formels

L'Analyse Concepts Formels utilise des matrices binaires mettant en relation des objets avec des attributs, une matrice porte le nom de contexte et est défini par $C(O, A, I)$ où O est l'ensemble des objets, A l'ensemble des attributs et I l'ensemble des relations entre les objets et les attributs.

Définition 1 (Contexte) *Soit un contexte $C(O, A, I)$ dans lequel O est l'ensemble des objets, A l'ensemble des attributs et I l'ensemble des relations entre les objets et les attributs.*

Nous définitions une opération, « prime » pour les objets et attributs, notée « ' ». Elle permet de passer des objets aux attributs et inversement. On lui donne un objet (resp. attribut) ou un ensemble d'objets (resp. attributs) et elle nous renvois l'ensemble des attributs (resp. objets) qui sont en correspondance.

Définition 2 (Connexion de Gallois) *Pour $X \subseteq O$ et $Y \subseteq A$, on définie :*
— $X' = \{y \in A : (x, y) \in I, \forall x \in X\}$
— $Y' = \{x \in O : (x, y) \in I, \forall y \in Y\}$

Dans la figure 3.1, nous avons $O = \{chat, chien, aigle, mouette, poule\}$, $A = \{griffes, ailes, nyclatope, vol\}$ et $I = \{(chat, griffes), (chat, nyctalope), (chien, griffes), (aigle, griffes), ...\}$. Nous avons entre autres pour les objets $chat' = \{griffes, nyctalope\}$ et $\{chat, aigle\}' = \{griffes\}$ et inversement pour les attributs $griffes' = \{chat, chien, aigle, poule\}$ et $\{griffes, ailes\}' = \{aigle, poule\}$.

	griffes	*ailes*	*nyctalope*	*vol*
chat	x		x	
chien	x			
aigle	x	x		x
mouette		x		x
poule	x	x		

FIGURE 3.1 – Contexte

Nous pouvont effectuer plusieurs opérations sur les contextes sans réelles pertes d'informations sur la structure. La première est la clarification, cela consiste à ne garder que les lignes et les colonnes qui ne sont pas en doublon.

Définition 3 (Contexte clarifié) *Soit un contexte clarifié $C(O, A, I)$:*
— $\forall x1, x2 \in O$ *si* $x1' = x2'$ *alors* $x1 = x2$
— $\forall y1, y2 \in A$ *si* $y1' = y2'$ *alors* $y1 = y2$

À partir du contexte non clarifié de la figure 3.2 nous obtenu le contexte clarifié de la figure 3.3 avec la suppression de la ligne 5 qui est le doublon de la ligne 2 et la colonne e qui est le doublon de la b. Nous pouvons le voir à travers les opérations de prime, deux objets (resp. attributs) sont en doublon losqu'ils obtiennent le même résultat. Nous obtenons $2' = \{a\}$ et $5' = \{a\}$. Il est important de bien comprendre que malgré la suppression de lignes ou de colonnes dans cette opération, nous ne perdons aucune données.

	a	b	c	d	e
1	x			x	
2	x				
3	x	x	x		x
4		x	x		x
5	x				
6	x		x		

FIGURE 3.2 – Contexte non clarifié

	a	b	c	d
1	x			x
2	x			
3	x	x	x	
4		x	x	
6	x		x	

FIGURE 3.3 – Contexte clarifié

La seconde opération est la plus utilisée et va plus loin dans la réduction de la taille du contexte sans perte de données, les contextes ainsi obtenus sont dit « contexte réduit » [1]. Le principe est toujours de supprimer les lignes et les colonnes dont on peut se passer, celles qui sont recalculables à partir de celles qu'on garde. On ne garde que les lignes et les colonnes qui ne sont pas l'intersection d'une ou plusieurs autres.

Définition 4 (Contexte réduit) *Un contexte clarifié est réduit $C(J, M, I)$ si et seulement si :*
— $\forall x \in J, \forall X \subseteq J$, *si* $x' = X'$ *alors* $x \in X$
— $\forall x \in M, \forall X \subseteq M$, *si* $x' = X'$ *alors* $x \in X$

Dans la figure 3.4 la ligne $2' = \{a\}$ est l'intersection des lignes $1' = \{a, d\}$ et $3' = \{a, b, c\}$ ou $1' = \{a, d\}$ et $6' = \{a, c\}$, on peut donc la supprimer. En revanche, nous n'avons aucune colonne dans ce cas ici. Lorsqu'on effectue la réduction d'un contexte, il est systématiquement sous entendu que l'opération de clarification est également effectuée.

	a	b	c	d
1	x			x
2	x			
3	x	x	x	
4		x	x	
6	x		x	

FIGURE 3.4 – Contexte non réduit

	a	b	c	d
1	x			x
3	x	x	x	
4		x	x	
6	x		x	

FIGURE 3.5 – Contexte réduit

Nous pouvons maintenant définir la notion de concept. Cela correspond aux rectangles maximaux qui se trouvent dans le contexte, regroupant tous les objets qui font partis du concept et tous les attributs qu'ils ont en commun. Un concept est l'ensemble des objets et attributs tel que tous les objets du concepts partagent les attributs du concepts et qu'il n'existe pas d'autre objet partageant les attributs du concept et pas d'autre attributs qui sont partagés par tous les objets du concept.

Définition 5 (Concept) *Soit un concept c,* $X \subseteq O$ *et* $Y \subseteq A$ *:*
— $c = \{X, Y\}$
— $\forall x \in O$, $x \in X \Leftrightarrow x' \subseteq Y$
— $\forall y \in A$, $y \in Y \Leftrightarrow y' \subseteq X$

Sur le contexte de la figure 3.5 nous avons entre autre le concept contenant les objets 3 et 4 et les attributs b et c dont les correspondances forment un rectangle plein.

Maintenant que nous avons défini la notion de concept, nous pouvons définir les treillis de concepts, également appelés treillis de Galois. Ce treillis se base sur une relation d'ordre entre les concepts exprimée sur un diagramme de Hasse. C'est un diagramme orientés du bas vers le haut où chaque élément, ici il s'agit du concept, est un nœud et chaque relation d'ordre est un arc.

1. « standard context » pour la documentation anglaise

Définition 6 (Treillis de concepts) *Soit deux concepts $c_1 = (O_1, A_1)$ et $c_2 = (O_2, A_2)$, on a $c_1 \leq c_2$ si et seulement si $O_1 \subseteq O_2$ ou $A_1 \supseteq A_2$.*

En plus des propriétés précédentes, un contexte peut contenir ce qu'on appelle des relations fléches qui ne viennent pas ajouter de l'information mais permettre une extraction plus rapide à porté humaine. Il en existe de trois types $\uparrow$, $\downarrow$, et $\updownarrow$.

Définition 7 (Relations flèches) *Soit un contexte $C(O, A, I)$, $o \in O$, $a \in A$:*
— *$o \uparrow a$ ssi $(o, a) \notin I$ et si $\exists x \in A$, $a' \subset x'$, $(o, x) \in I$: $\forall y \in A$, $a' \subset y' \Rightarrow y = x$*
— *$o \downarrow a$ ssi $(o, a) \notin I$ et si $\exists x \in O$, $o' \subset x'$, $(x, a) \in I$: $\forall y \in O$, $o' \subset y' \Rightarrow y = x$*
— *$o \updownarrow a$ ssi $o \uparrow a$ et $o \downarrow a$.*

	A	B	C	D	E
1	x	x	x	x	$\updownarrow$
2	x	$\updownarrow$	x	x	x
3	x	x	x	$\updownarrow$	
4	x	$\uparrow$	$\updownarrow$	x	x
5	x	x	$\updownarrow$	$\uparrow$	
6	$\updownarrow$	x			

FIGURE 3.6 – Contexte avec relations fléches

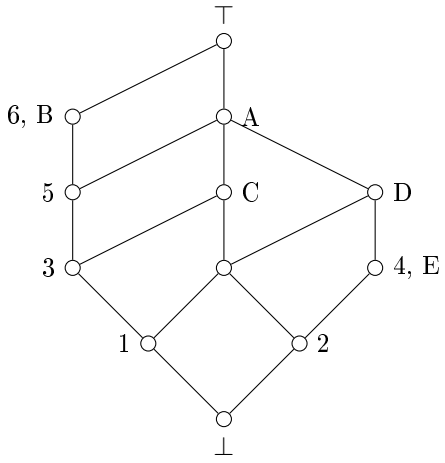


FIGURE 3.7 – Treillis associé

3.2 Ensemble ordonné

Un ensemble ordonné est un ensemble P d'éléments avec une relation d'ordre $\leq$, qu'on note $(P, \leq)$. Pour le représenter, on utilise des diagrammes de Hasse. Pour un ensemble $X \subseteq P$ on note $\uparrow X$ le filtre (resp. $\downarrow X$ l'idéal) de X. Pour un élément $x \in P$, on note $\uparrow x$ le filtre principal (resp. $\downarrow x$ l'idéal principal) de x tel que pour $\uparrow x = y$ (resp. $\downarrow x = y$) on a $x \leq y$ (resp. $y \leq x$).

Définition 8 (Ensemble ordonné) *Soit un ensemble ordonné $(P, \leq)$, $X \subseteq P$ et $Y \subseteq P$:*
— *$\forall x, y \in P$, on a $x \leq y$ ou $y \leq x$*
— *$\uparrow X = Y \Rightarrow x \leq y \ \forall x \in X, \forall y \in Y$*
— *$\downarrow X = Y \Rightarrow y \leq x \ \forall x \in X, \forall y \in Y$*

Dans la figure 3.8 nous avons $\uparrow a = a, b, c$ et $\downarrow b = a, b$.

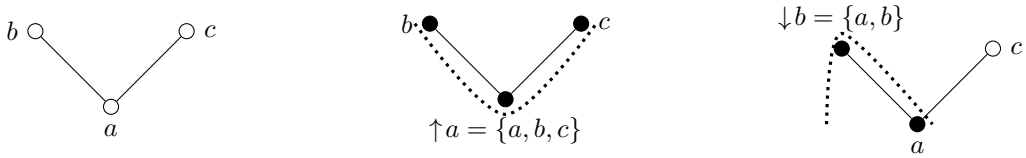


FIGURE 3.8 – Diagrammes de Hasse avec filtres et idéaux

3.3 Treillis

Un treillis est un ensemble ordonné $(T, \leq)$ sur lequel on définit pour chaque tuple un supremum ou borne supérieure noté $\vee$ et un infimum ou borne inférieure noté $\wedge$, on le note $(T, \leq, \vee, \wedge)$. Pour trois éléments $x, y, z \in P$ tel que $z \leq x$ et $z \leq y$ on peut dire que $x \vee y = z$. Le second est celui d'infimum ou borne inférieure noté $\wedge$. Et de la même façon pour trois éléments $x, y, z \in P$ tel que $x \leq z$ et $y \leq z$ on peut dire que $x \wedge y = z$. Lorsqu'un élément est la borne supérieure (resp. inférieure) de tous les autres éléments on dit qu'il ferme le treillis par les supremums (resp. infimums), on note cet élément $\top$ (resp.$\bot$).

Comment définir les notions de borne sup et borne inf ?

Définition 9 (Treillis) *Soit un treillis* $(T, \leq, \vee, \wedge)$ *:*
— $\forall x, y \in T,\ \exists z \in T : x \vee y = z$
— $\forall x, y \in T,\ \exists z \in T : x \wedge y = z$

Définition 10 (Sous treillis) *Soit un treillis* $(T, \leq, \vee, \wedge)$ *et son sous treillis* $(T_S, \leq_S, \vee, \wedge)$ *si et seulement si :*
— $T_S \subseteq T$
— $\forall x, y \in T_S,\ x \leq y \Rightarrow x \leq_S y$

Sur notre exemple en figure 3.9 nous avons $a = b \vee c$, $d = b \wedge c$, $\top = d$ et $\bot = a$.

FIGURE 3.9 – Treillis avec supremum et infimum

Un élément qui est pas le supremum (resp. infimum) d'autres éléments est un $\vee$-irréductible (resp. $\wedge$-irréductible) et on note l'ensemble de ces élément $J(T)$ (resp. $M(T)$). Bien que le $\top$ et le $\bot$ soient des irréductibles, il est commun de ne pas toujours les prendre en considération, nous ne les prendront pas non plus dans ce rapport, il sera précisé lorsqu'ils le seront.

Définition 11 ($\vee$-irreductible et $\wedge$-irreductible) *Soit un treillis* $(T, \leq, \vee, \wedge)$, $x, y, z \in T$ *:*
— x *est* $\vee$*-irreductible ssi* $x \vee y = z \Rightarrow x = z$
— x *est* $\wedge$*-irreductible ssi* $x \wedge y = z \Rightarrow x = z$

Jusqu'à présent nous utilisions des labels sur la totalité des nœuds, par la suite nous en metteront uniquement sur ceux importants dont les irréductibles. Dans cet objectif, nous utiliserons des lettres (resp. chiffres) pour les $\vee$-irréductibles (resp. $\wedge$-irréductible), de même que les labels $\top$ et $\bot$. Sur l'exemple de la figure 3.10 les irréductibles sont représentés en noir, d'abord les $\vee$-irréductibles puis les $\wedge$-irréductibles.

FIGURE 3.10 – Diagrammes de Hasse avec irréductibles

Il existe une catégorie de treillis appelée treillis distributif dont la particularité d'avoir la distributivité des opérations $\vee$ et $\wedge$. Pour tout $x, y, z \in T$ on a $x \vee (y \wedge z) = (x \wedge y) \vee (x \wedge z)$ et $x \wedge (y \vee z) = (x \vee y) \wedge (x \vee z)$. Cette propriété transparait à travers trois conditions équivalentes qui permettent d'attester ou non de la distributivité du treillis permettant ainsi d'utiliser la plus adaptée dans une situation donnée.

Définition 12 (Treillis distributif) *Un treillis* $(T, \leq, \vee, \wedge)$ *est distributif si et seulement si au moins l'une des trois conditions équivalentes suivantes est vérifiée :*
— $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \wedge (y \vee z)$
— *Le treillis ne possède ni* N_5 *ni* M_3 *en guise de sous treillis*
— *Le contexte réduit contient une seule relation flèche double par ligne et par colonne*

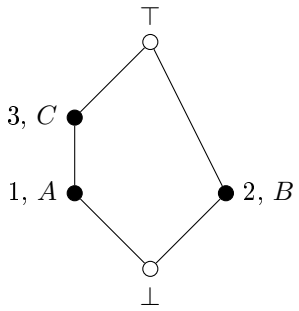


FIGURE 3.11 – N_5 (à déplacer)

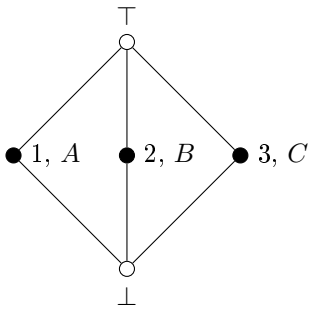


FIGURE 3.12 – M_3 (à déplacer)

3.4 Graphe médian

Un graphe est un diagramme composé de nœuds et d'arcs. Pour chaque triplets de nœuds, on définie l'ensemble des nœuds formant sur les plus courts chemins entre les éléments du triplet. Si pour tous les triplets du graphe cet ensemble ne contient qu'un unique nœud, alors on dit que ce graphe est médian.

4 Contribution

4.1 Implémentation

Mon stage s'est déroulé en deux phases. La première consiste à implémenter une méthode existante afin de la tester sur des cas de façon automatique. La seconde partie du stage a pour but d'améliorer la méthode précédente afin de prendre en compte le problème de fusion des nœuds. Avec en parallèle un renforcement de la compréhension du domaine à travers des articles sur l'existant et des discussions avec l'équipe.

La toute première question à se poser pour l'implémentation est celle du langage. Le problème n'ayant pas de contraintes particulières je suis partis sur un programme orienté objet en Python. C'est un langage assez répandu et très malléable, ce qui fait de lui un choix basique mais fiable. Choix également renforcé par mon expérience principalement tournée sur du langage objet permettant de m'adapter plus facilement. Une fois ce choix fait, le plus long a été d'adapter l'algorithme pour sa mise en fonction, cela m'a également permis de comprendre de mieux en mieux les mécanismes mis en jeu. À ce stade nous obtenons le workflow en figure 4.1. Le programme commence par effectuer l'importation des données d'entrée. Il extrait un contexte pour chaque treillis formé par les filtres des atomes et effectue l'algorithme vu précédemment en figure **??**. Une fois fait, le programme regroupe les contextes en un unique et exporte le résultat sous forme graphique à l'aide de Graphviz[5] à travers sa bibliothèque[6]. Nous avons toujours dans cet état le problème sur les cas non triviaux développé plus tôt.

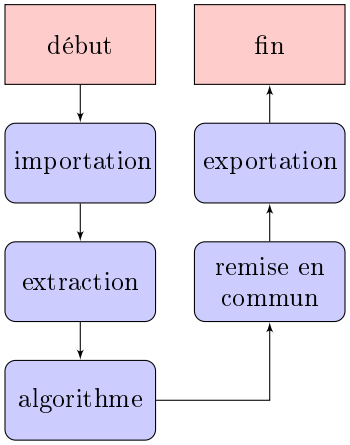


FIGURE 4.1 – Worflow de la première version du programme

Il est question à présent de la fusion des nœuds dans le but d'obtenir le treillis et des conditions de cette fusion pour en ressortir des règles de fusions. Nous allons nous baser sur le cas mis en avant dans [1] avec en figure 4.2 le treillis de départ. La figure 4.3 est un rappel du résultat dans l'état actuel et la figure 4.4 est le résultat optimal. On remarque que les points noirs 10 et 12 du résultat obtenu peuvent ne faire qu'un pour obtenir le point 8 du résultat désiré. En regardant également le treillis distributif maximum en figure 4.5 on a l'intuition qu'il suffit de remplacer les points qu'on souhaite fusionner par leur supremum, cela à été notre point de départ.

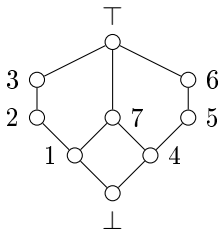


FIGURE 4.2 – Treillis de départ

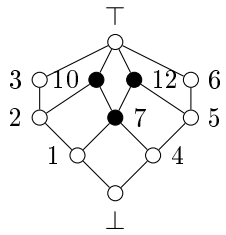


FIGURE 4.3 – Treillis obtenu

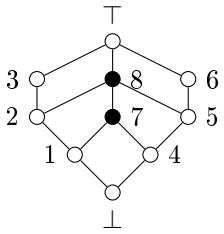


FIGURE 4.4 – Treillis désirée

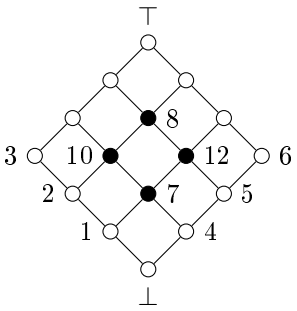


FIGURE 4.5 – Treillis distributif maximum

Nous avons ensuite travaillé sur cette piste à partir de toute une série de treillis qui se trouve en annexe 2 [1] ayant chacun ses spécificités. À chaque résultat déviant du résultat souhaité sur l'un des cas, nous avons effectué un ajustement des conditions de fusion des nœuds tout en vérifiant le bon fonctionnement sur les cas déjà traités. Nous sommes arrivés à trois conditions.

Conditions de fusion

— Les deux nœuds à fusionner ne doivent pas faire partis des nœuds du treillis de départ.
Les points 10 et 12 de la figure 4.3 ne sont pas présent dans la figure 4.2.

— Les deux nœuds à fusionner doivent être en couverture d'un nœud présent dans au moins deux filtres d'atomes.
Les nœuds 10 et 12 sont supremum directs du nœud 7 qui est présent dans les filtres de l'atome 1 et de l'atome 4.

— Les deux nœuds à fusionner ne doivent pas avoir de nœuds en commun dans leurs idéaux une fois que les idéaux du nœud en commun (celui dont il est question dans la condition précédente) leur sont retirés.
Nous avons $\downarrow 10 \setminus \downarrow 7 = 2$ et $\downarrow 12 \setminus \downarrow 7 = 5$.

Avec la mise en place de la fusion des nœuds, nous obtenons un nouveau workflow en figure 4.6.

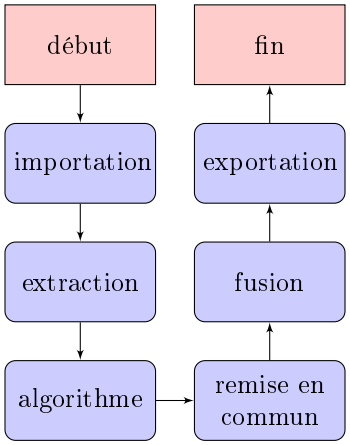


FIGURE 4.6 – Worflow de la seconde version du programme, avec la fusion des nœuds

1. revoir la réf aux annexes

4.2 Problèmes et difficultés

Nous avons rencontré un problème qui porte sur des cas où un passage complet de la méthode avec post traitement de fusion ne suffit pas. La fusion recréant une situation avec des treillis d'atomes non distributif et demandant donc de refaire un passage dans tout le processus.

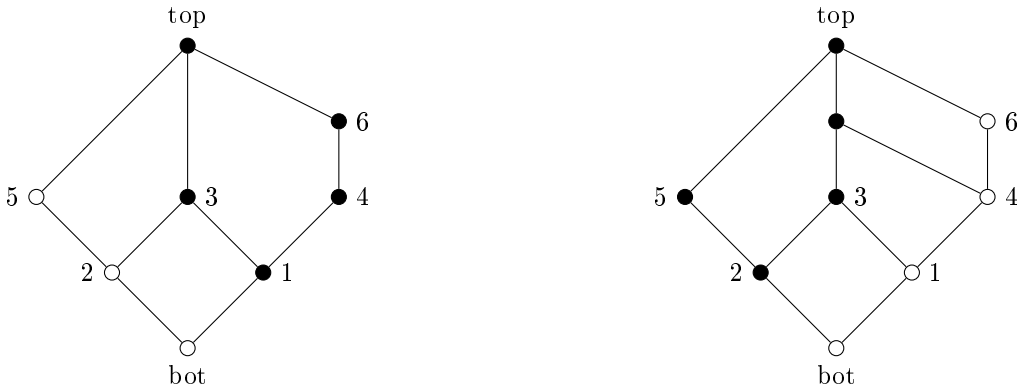


FIGURE 4.7 – Cas cla_v7 : première itération

À ce stade, plusieurs correctifs ont été envisagé. Le premier est de faire une boucle de traitement où on refait tout le processus tant que tous les treillis d'atome ne sont pas disctributif, c'est la solution utilisé à l'heure actuelle. Le second est de faire une boucle sur tout le cœur du processus et de faire les fusions de nœuds à la fin. Cette solution contrairement à la première empêche toute possibilité de cycle dû à la fusion mais donne la plupart du temps le treillis distributif maximal pour chaque atome et la fusion devient par conséquent très compliquée, ce qui donne des treillis bien plus grand que ceux visés. La troisième solution encore théorique consiste à faire les fusions pendant le processus en lui même et de boucler sur l'ensemble tant que les sous treillis formés par les filtres des atomes ne sont pas distributif.

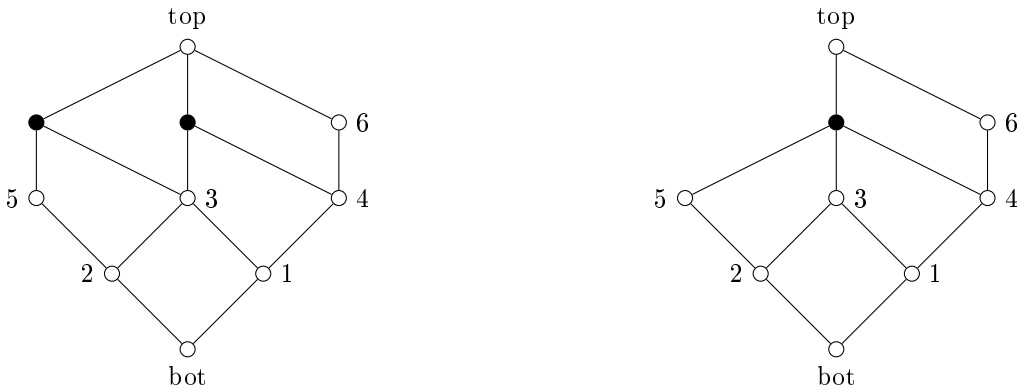


FIGURE 4.8 – Cas cla_v7 : seconde itération

4.3 Ouvertures

4.3.1 Système de boucle

Comme nous l'avons vu précédemment, nous devons boucler sur l'algorithme afin de prendre en considération les cas qui génèrent un nouvel N_5 ou M_3 lors de leur première résolution. Cette méthode présentée ici n'est pas utilisé dans le programme mais est une piste intéressante à garder à l'esprit en cas de problèmes futurs. Il s'agit de faire la fusion pas à pas. On execute la méthode cla différemment. On extrait les contextes des treillis des atomes puis on effectue pour chacun d'eux la transformation en contexte de treillis distributif et on l'assemble avec le contexte global et c'est à ce moment où on fait les fusions si besoin avant de passer au contexte du treillis de l'atome suivant. Cette façon de faire peut permettre de gagner du temps sur la phase de la fusion mais en ferra perdre pour l'autre. Je ne peux pas déterminer laquelle est la meilleure à l'heure actuelle, reste à voir si à l'avenir de nouveaux problmes se posent et si cette nouvelle méthode peut apporter des réponses.

4.3.2 Le cas des chaines

Je ne l'ai pas encore détaillé en dehors du workflow mais le programme effectue également une épuration des chaînes qu'il rencontre lorsque le concept courant n'est pas un concept présent dans le treillis d'origine et qu'il ne possède pas plus de un sup et pas plus de un inf. Nous pouvons résumer les conditions de cette façon :

Condition d'épuration Soit a le nœud qu'on souhaite épurer :

- $a \in J \setminus J_{origine}$
- $\forall x_1, x_2 \in J, b, c \in J$
  $\uparrow b \not\subset \uparrow x_1 \not\subset \uparrow a$
  $\uparrow a \not\subset \uparrow x_2 \not\subset \uparrow c$
- $\forall y_1, y_2 \in J, b, c \in J$
  $b \neq y_1, c \neq y_2$
  $\uparrow b \not\subset \uparrow x_1 \not\subset \uparrow a$
  $\uparrow a \not\subset \uparrow x_2 \not\subset \uparrow c$

TODO : Trouver notation pour il existe un unique b, c

Malheuresement, cette partie n'est pas fiable et résulte principalement d'une intuition basé sur un objectif à atteindre et donnant lieu à un raccourci par rapport à une propriété ou à un traitement non mis à découvert. Ce cas s'est présenté une seule fois jusqu'à présent, sur le cas « cla_v6 ». L'épuration est effectué au même moment que la fusion. Si on ne l'effectue pas pour ce cas, nous allons boucler et obtenir une solution assez loin de ce qu'on cherche du fait qu'on recommence toute la procédure à cause de la présence d'un N_5. C'est sous doute le 1er aspect du programme à garder sous surveillance.

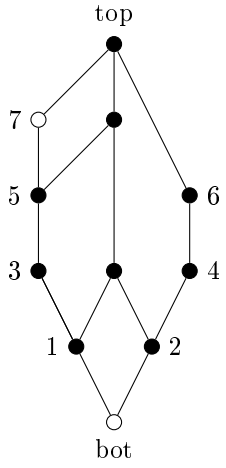


FIGURE 4.9 – Cas cla_v6

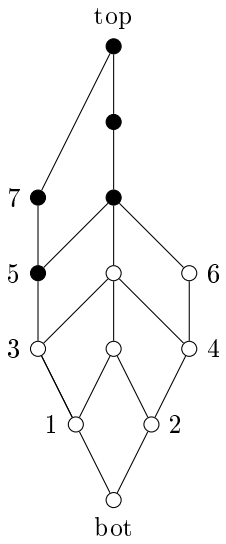


FIGURE 4.10 – Cas cla_v6, avant épuration

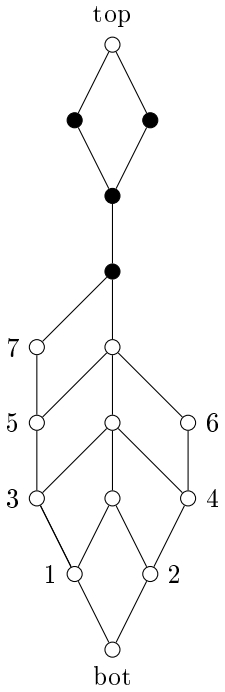


FIGURE 4.11 – Cas cla_v6, sans épuration

4.3.3 Optimalité du treillis

4.3.4 Optimisation du programme

5 Bilan

à revoir

5.1 Contribution

Nous cherchions à produire des treillis utilisables sous forme de graphes médians afin d'être utilisable dans la phylogénie en encodant l'ensemble des arbres parcimonieux dans un unique graphes. Pour se faire nous nous sommes basés sur les travaux de H.J. Bandelt, U. Priss et de l'équipe, en recherchant à générer des treillis distributifs sur leurs atomes à partir de treillis quelconque.

Nous avons obtenu un programme permettant de le faire. Il commence par extraire le contexte initial d'un fichier pour ensuite regarder si tous les treillis des atomes sont distributifs en calculant les relations flèches pour utiliser la propriété que le contexte réduit d'un treillis distributif ne possède uniquement une relation fléche double par ligne et par colonne et aucune autre relations fléches. Si ce n'est pas le cas, nous appliquons une méthode de transformation sur chacun d'eux avant de les remettre en commun dans un unique treillis. À partir de là, nous effectuons la fusion deux à deux de tous les couples de nœuds remplissant les conditions suivantes (avec a et b les deux nœuds en question) :

— $a, b \in J \setminus J_{origine}$
  $a \neq b$
— $\forall x_1, x_2 \in J, c \in J$
  $\uparrow a \not\subset \uparrow x_1 \not\subset \uparrow c$
  $\uparrow b \not\subset \uparrow x_2 \not\subset \uparrow c$
— $\forall z_a \in \downarrow a \setminus \downarrow c \setminus a$
  $\forall z_b \in \downarrow b \setminus \downarrow c \setminus b$
  $z_a \neq z_b$

Une fois les fusions effectuées nous recommencons à l'étape de vérification de la distributivité des treillis des atomes. Une fois que nous l'atteignons le programme s'arrête et génére le treillis obtenu sous forme pdf et un fichier texte du contexte réduit exploitable par ConExp. Vous pourrez trouver en Annexe 1 le workflow complet du programme et en Annexe 2 tous les cas qui ont été traité et testé pendant le développement. Tandis que l'Annexe 3 présente le formatage du fichier source et du fichier d'exportage pour ConExp.

5.2 Stage

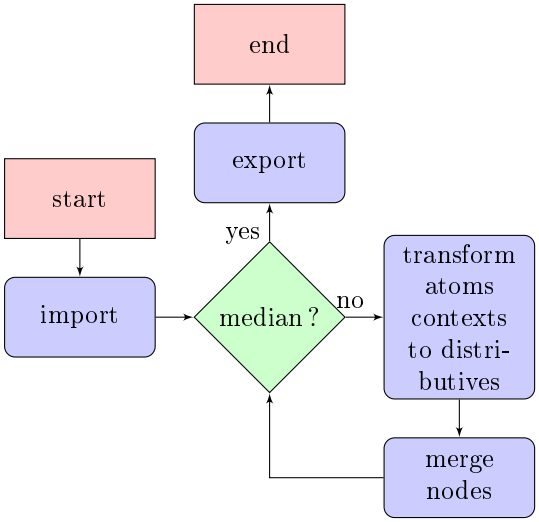
Ce stage a été bénéfique sur le plan technique. Il m'a permit de confirmer et d'approfondir des notions vu en cours tout en leur offrant un cas concret d'utilisation possible. J'ai également d'améliorer mes compétences en Python et Latex.

Mais également sur le plan personnel, le Loria reste un lieu des plus intéressant pour échanger sur un sujet afin de faire avancer la problèmatique avec des personnes ayant des approches différentes pour s'approcher au mieux d'une solution fiable.

Annexes

6 Annexe : workflows

Cette annexe présente le workflow du programme final.



import Lecture du fichier et création du contexte et du treillis de base

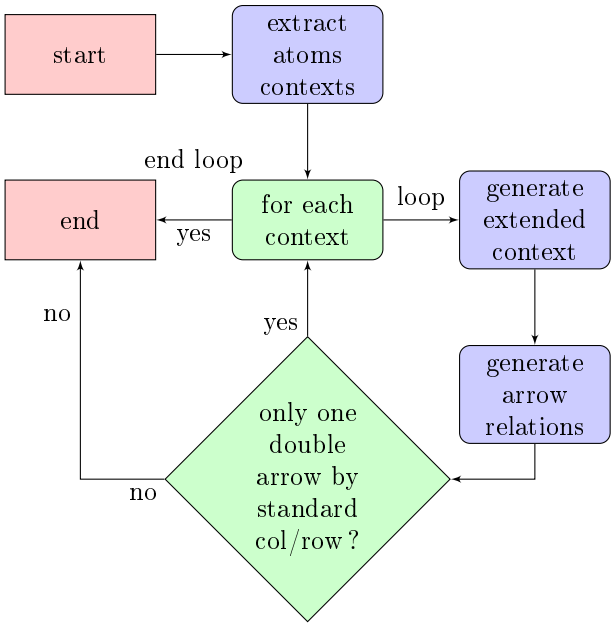
median ? Vérification si le treillis est "médian" (treillis des atoms distributif) : détaillé par la suite

transform atoms... Effectue l'algorithme de cla sur chaque contexte des atoms et les assemble : détaillé par la suite

merge nodes Fusionne les concepts : détaillé par la suite

export Création du fichier pdf de représentation du treillis après tout le traitement et génération du fichier d'exportation à destination de ConExp

FIGURE 6.1 – Global



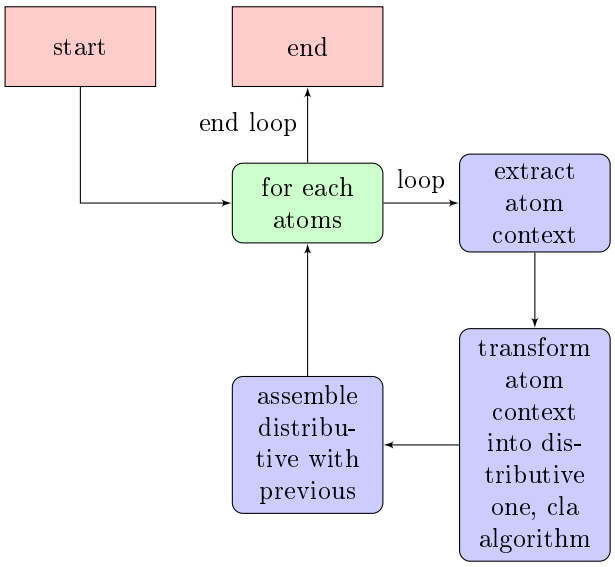
extract atoms contexts Extrait le contexte du sous treillis de l'atome

generate extended context Génère le contexte du treillis

generate arrow relations Génère les relations fléches

only one double... Test s'il y a une seule relation fléche par ligne et colonne du contexte standardisé et qu'elle soit double

FIGURE 6.2 – Block « Median ? »

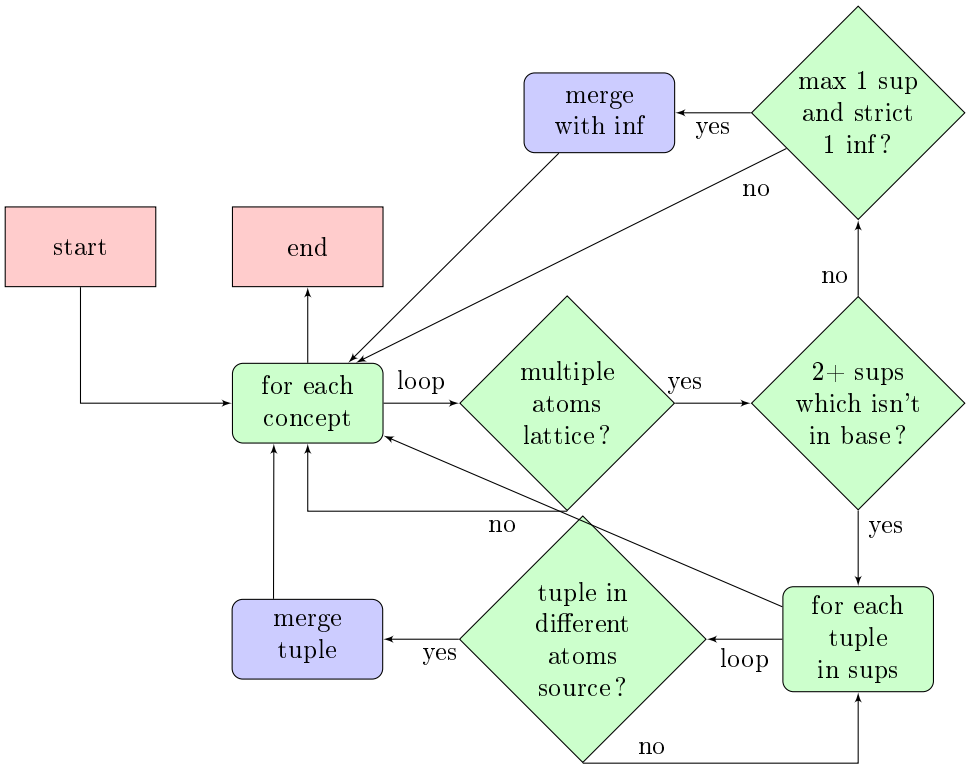


extract atoms contexts Extrait le contexte du sous treillis de l'atome (peut être déplacé hors de la boucle pour réduire le temps de traitement, dans la théorie du moins, nécessite des tests)

transform atoms... Effectue l'algorithme de cla avec quelques ajouts : si le contexte obtenu ne dispose pas de relation ou pas d'attribut on crée un nouvel attribut en relation avec tous les objets du contexte (potentielle simplification)

assemble distributive... Ajoute le contexte obtenu avec l'ensemble en créant de nouveau attributs et sans oublier d'ajouter une relation entre chaque nouvel attribut et l'atome

FIGURE 6.3 – Block « Transform atoms contexts to median »



multiple atoms lattice ? Regarde si le concept appartient au treillis de plusieurs atomes

2+ sups Regarde si le concept dispose de 2 sups directs

max 1 sup... Regarde si le concept a maximum 1 sup et strictement 1 inf

tuple in different... Regarde si les deux éléments du tuple appartiennent au treillis d'atome différent sans passer par le concept

merge tuple On fusionne les deux éléments du tuple

FIGURE 6.4 – Block « Merge nodes »

7 Annexe : Exemples de cas

Cette annexe regroupes tous les cas utilisés pour les divers essais de cas particuliers avec l'état initial et l'état final pour chacun. Ils considèrent tous le programme final du stage.

1 Dérivé de N_5

Tout d'abord, une série dérivé du cas cla.

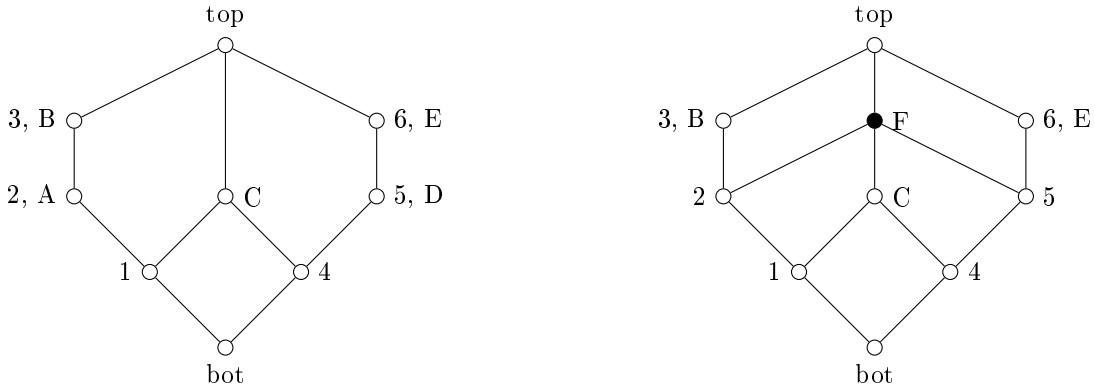


FIGURE 7.1 – Cas cla_v1

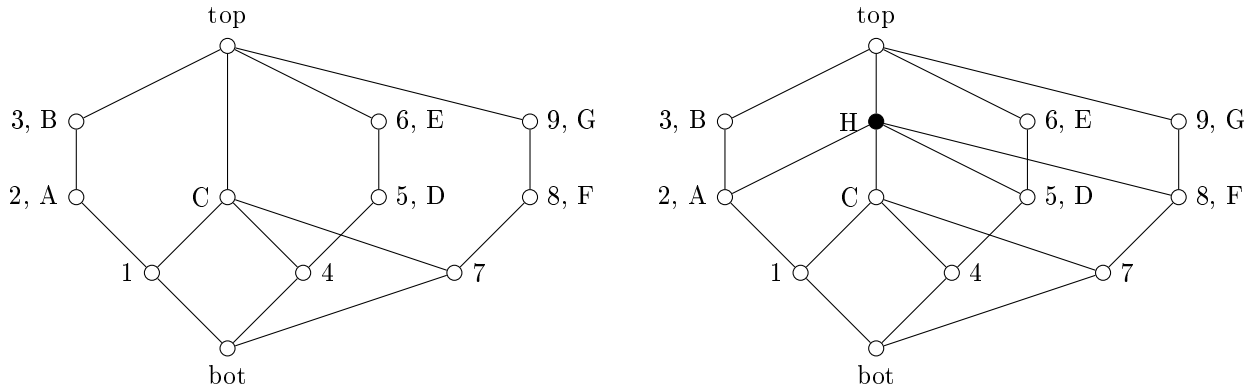


FIGURE 7.2 – Cas cla_v2

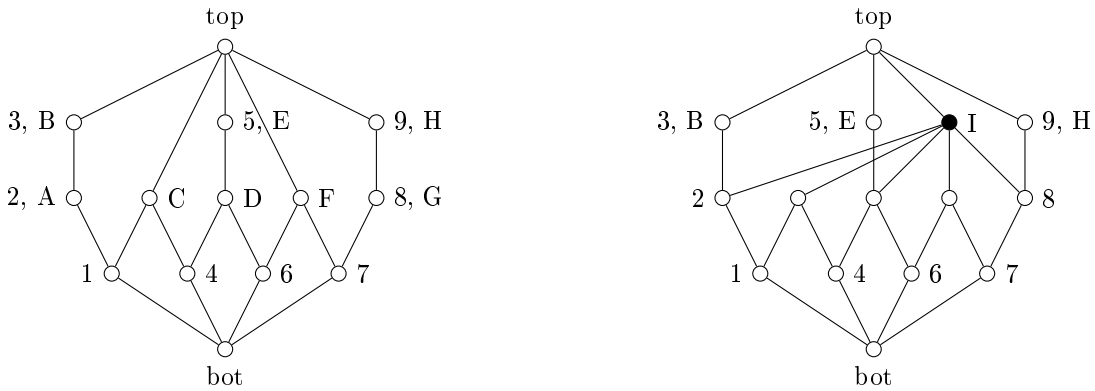


FIGURE 7.3 – Cas cla_v3

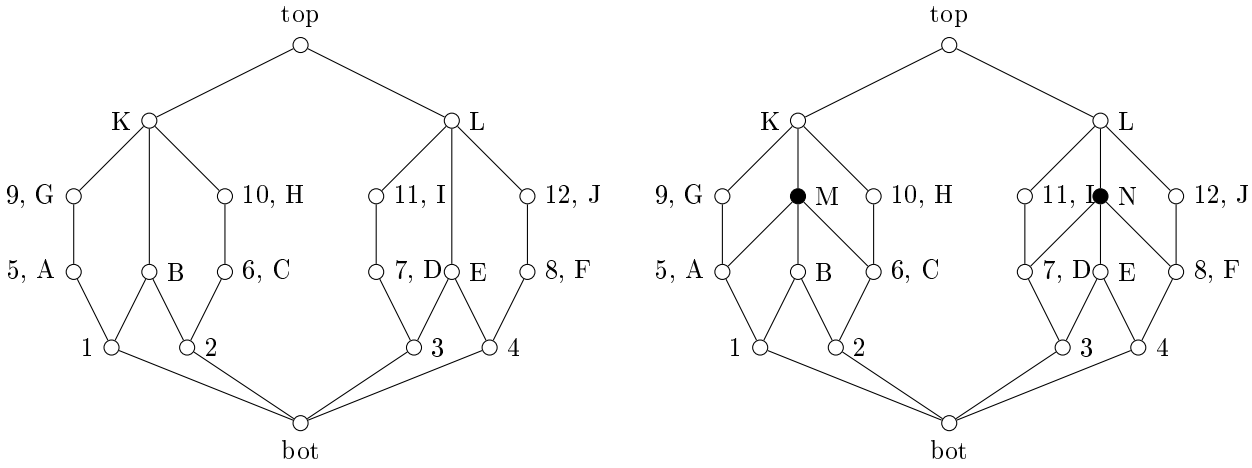


FIGURE 7.4 – Cas cla_v4

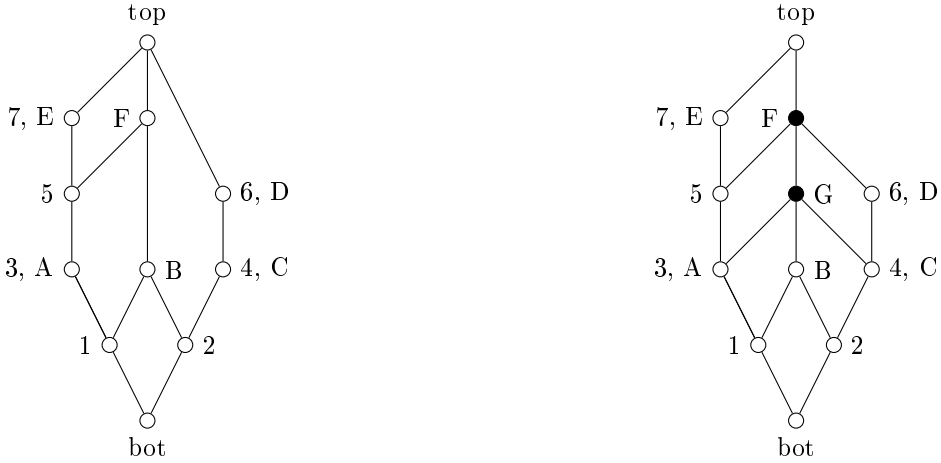


FIGURE 7.5 – Cas cla_v6

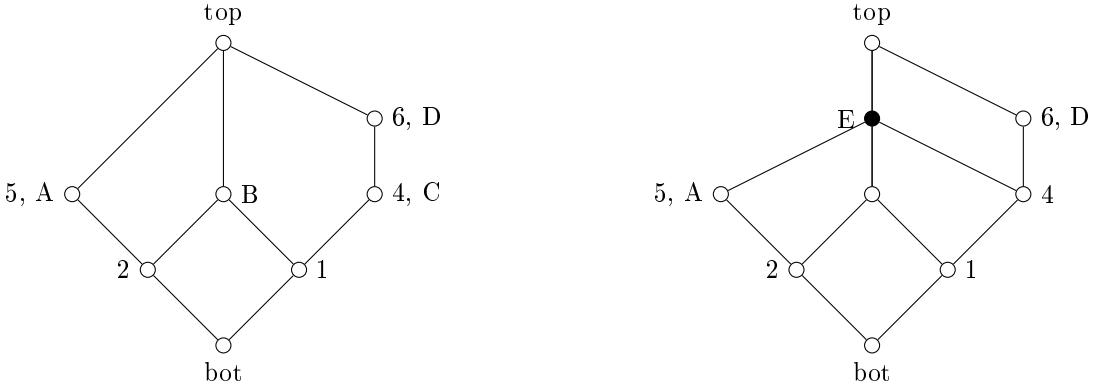


FIGURE 7.6 – Cas cla_v7

2 Dérivé de M_3

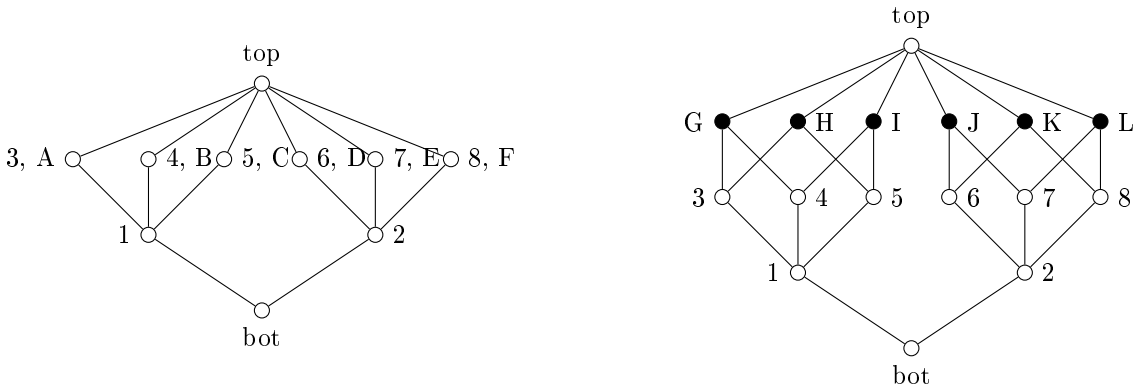


FIGURE 7.7 – Cas dm3_v0

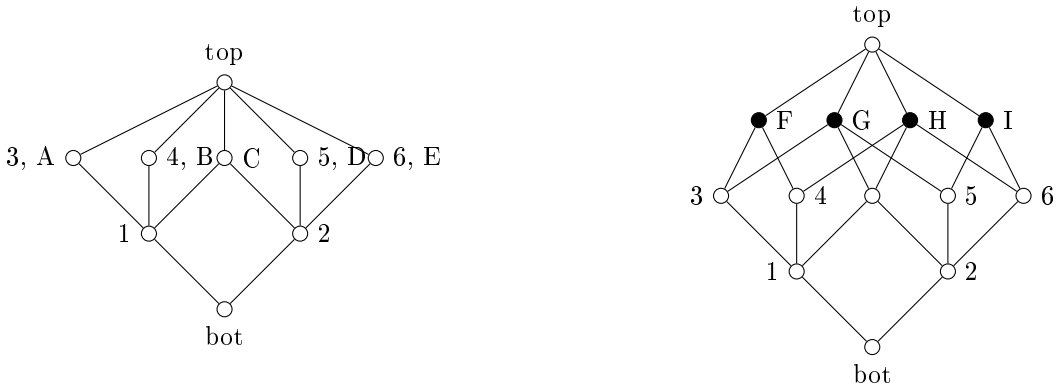


FIGURE 7.8 – Cas dm3_v1

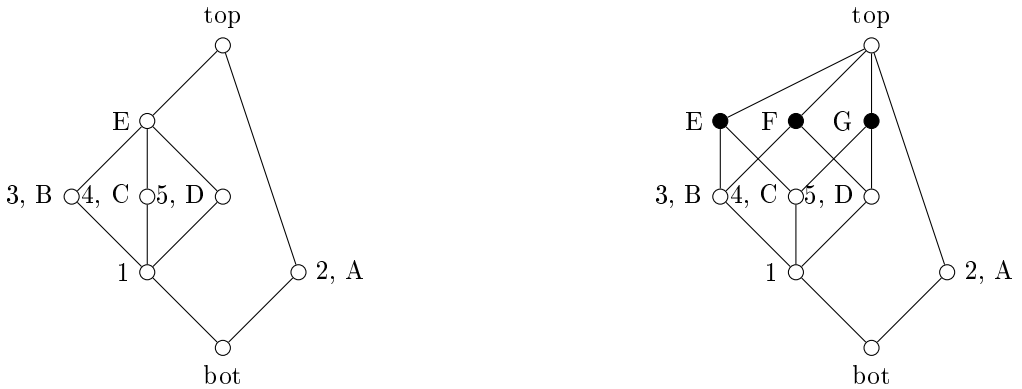


FIGURE 7.9 – Cas dm3_v2

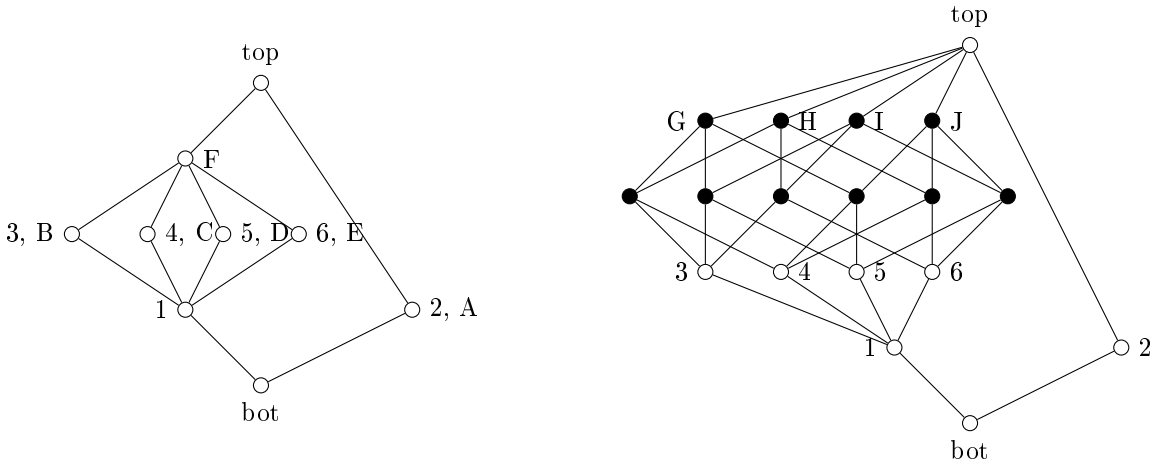


FIGURE 7.10 – Cas dm3_v3

3 Cas de H.J. Bandelt

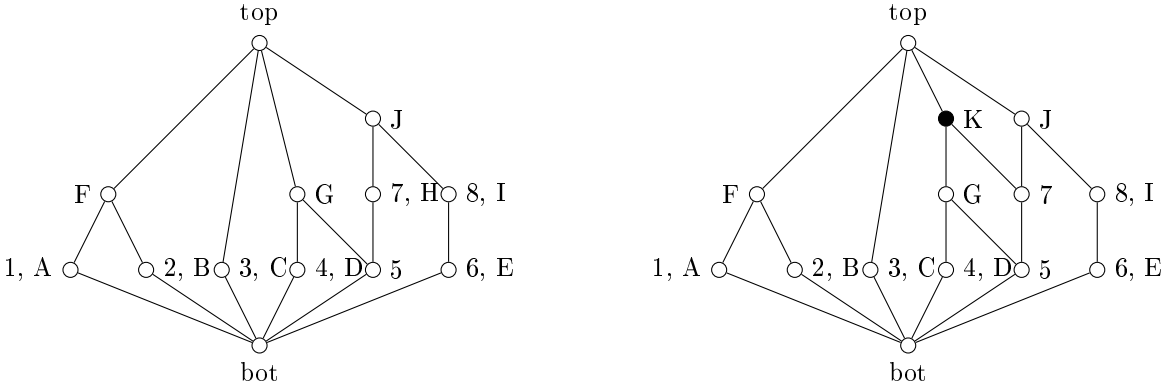


FIGURE 7.11 – Cas Bandelt table 1 ([7])

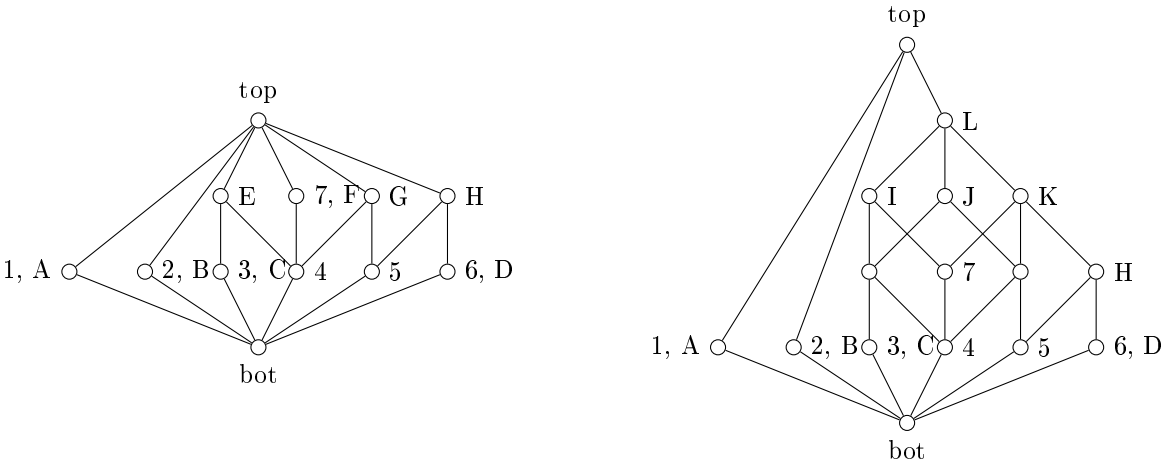


FIGURE 7.12 – Cas Bandelt table 2 ([7])

4 Cas de U. Priss

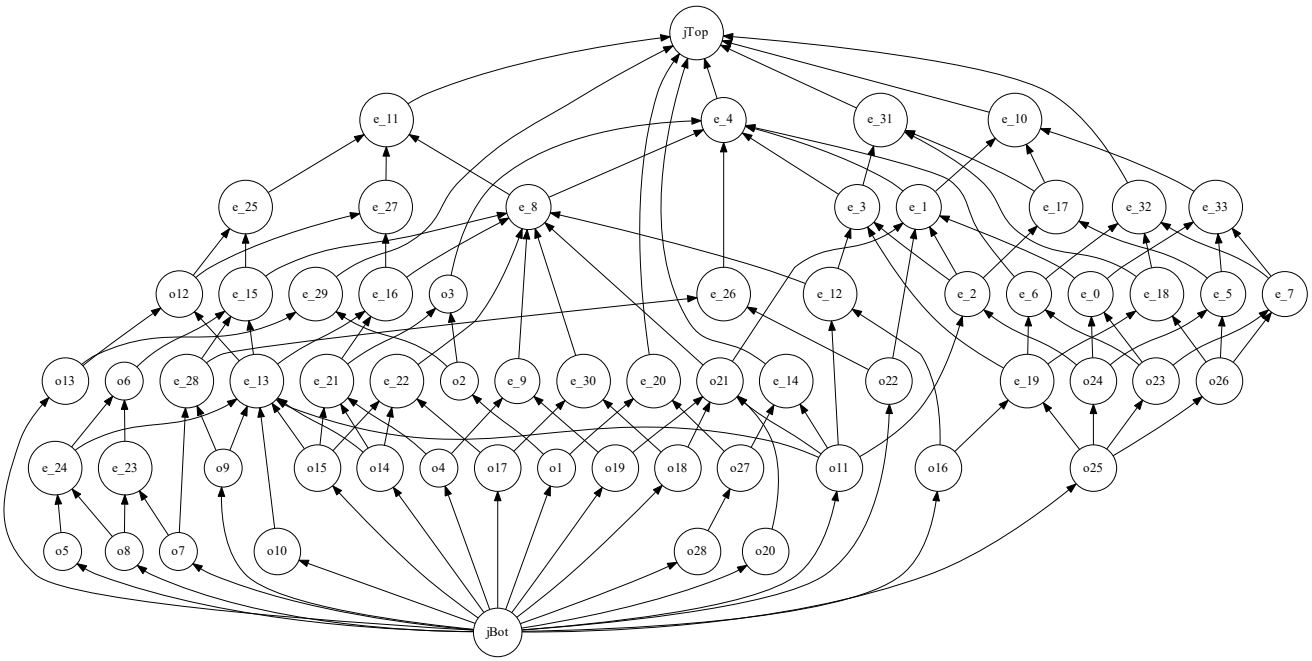


FIGURE 7.13 – Cas Priss ([2])

8 Annexe : Fichiers d'entré et de sortie

Cette annexe présente le fichier texte d'entré contenant le contexte de base et le fichier texte de sortie pour l'utilisation du résultat sous ConExp.

Input

Le fichier d'entré peut prendre de multiples formes. Il demande néanmoins d'avoir en première ligne les attributs séparé par une tabulation. Chaque ligne suivant correspondra à un nouvel objet sans avoir à indiquer son labe et dont chaque correspondance ou absence de correspondance séparées par un tabulation. Une non correspondace se marque avec un « 0 » ou l'absence de caractère. La correspondance est signalé par tout autre caractère.

a1	a2	a3	a4	a5
x	x		x	
	x	x		x
x			x	
		x		x
				x

a1	a2	a3	a4	a5
1	1	0	1	0
0	1	1	0	1
1	0	0	1	0
0	0	1	0	1
0	0	0	1	0
0	0	0	0	1

FIGURE 8.1 – Fichier source de cla_v1

Output

Le fichier de sortie quant à lui respecte la mise en forme d'importation de ConExp. Les attributs sur la première ligne séparé par une tabulation suivis d'une ligne vide et enfin des correspondances avec un objet par ligne sans indiquer le label et une notation binaire « 0 » et « 1 » indiquant respectivement la non correspondance et la correspondance.

a1	a2	a3	a4	a5
1	1	0	1	0
0	1	1	0	1
1	0	0	1	0
0	0	1	0	1
0	0	0	1	0
0	0	0	0	1

FIGURE 8.2 – Exportation de cla_v1 pour ConExp

Bibliographie

[1] Alain Gély, Miguel Couceiro, and Amedeo Napoli. Achieving distributivity in formal concept analysis. *journal name*, page 12, 2018.

[2] Uta Priss. Representing median networks with concept lattices. *journal name*, page 11, 2013.

[3] Hans-Jürgen Bandelt and Jarmila Hedlíková. Median algebras. *Discrete Mathematics*, page 30, 1980.

[4] Alain Gély, Miguel Couceiro, Yassine Namir, and Amedeo Napoli. Contribution à l'étude de la distributivité d'un treillis de concepts. *journal name*, page 12, 2018.

[5] John Ellson, Emden Gansner, Yifan Hu, Erwin Janssen, and Stephen North. Documentation graphviz pour python. Disponible sur : `https://www.graphviz.org/`. Consulté de avril à août 2018.

[6] Sebastian Bank. Documentation graphviz pour python. Disponible sur : `https://pypi.org/project/graphviz/`. Consulté de avril à août 2018.

[7] Hans-Jürgen Bandelt, Peter Forster, and Röhl Arne. Median-joining networks for inferring intraspecific phylogenies. *Molecular Biology and Evolution*, page 12, 1998.

Todo list