

Extraction Parallèle des Dépendances Fonctionnelles et Dépendances Fonctionnelles Conditionnelles

Eve Garnaud

Université de Bordeaux, LaBRI

Motivations Générales

Nom	Prénom	Poste
Dupond	Jean	MDC
Dupond	Pierre	MDC
Durant	Jean	MDC
Martin	Pierre	Prof.
Petit	Pierre	Admin.

```
SELECT Nom, Prénom, Poste  
FROM Table  
WHERE Nom = 'Dupond' & Prénom = 'Pierre'
```

Motivations Générales

Nom	Prénom	Poste
Dupond	Jean	MDC
Dupond	Pierre	MDC
Durant	Jean	MDC
Martin	Pierre	Prof.
Petit	Pierre	Admin.

```
SELECT Nom, Prénom, Poste  
FROM Table  
WHERE Nom = 'Dupond' & Prénom = 'Pierre'
```

Si on dispose de l'information *Nom* → *Poste* alors on ne cherche qu'une seule ligne puisque toutes les personnes ayant le même nom occupent le même poste.

Enoncé du problème

Soit T une table définie sur $n + 1$ attributs.

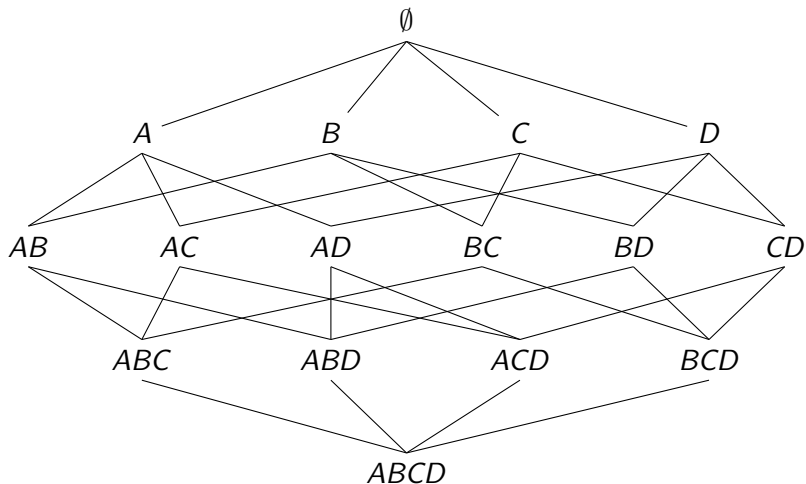
On cherche les dépendances fonctionnelles minimales satisfaites par T

Dépendances fonctionnelles minimales

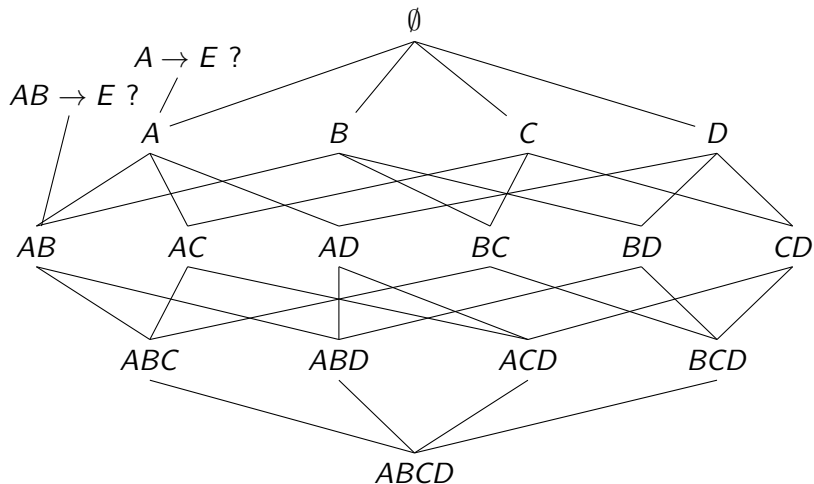
Une DF est minimale si elle est de la forme $X \rightarrow A_i$ avec

- $A_i \in attr(T) \setminus X$,
- $X \subset attr(T)$ et $\nexists X' \subset X$ tel que $X' \rightarrow A_i$.

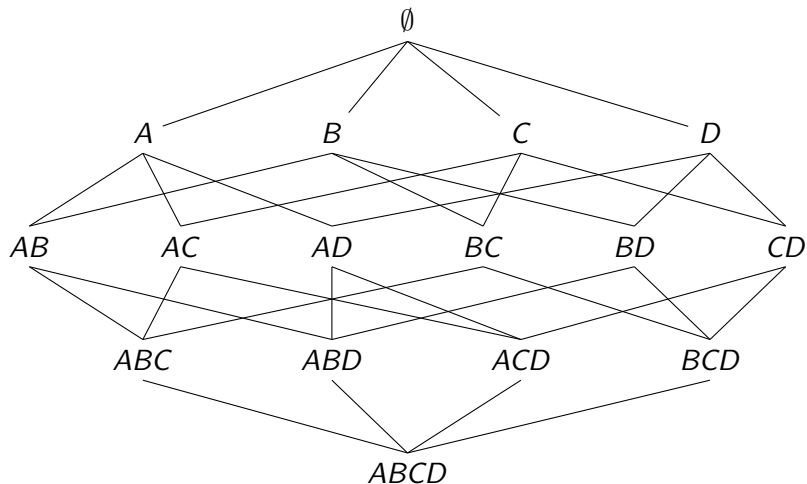
Espace de recherche construit sur n attributs



Espace de recherche construit sur n attributs



Espace de recherche construit sur n attributs



Complexité en $O(2^n)$

Rapide état de l'art

Avec pré-calcul

- ⇒ TANE, Dep_Miner, FastFDs, FD_Mine
- + pas d'accès aux données durant le calcul des DFs
- complexité des pré-calculs en $O(m^2)$ (m le nombre de tuples)

Sans pré-calcul

- ⇒ FUN
- + pas de pré-calcul
- parcours en largeur du treillis

Stratégie d'élagage

Elagage vers le bas

$$X \rightarrow A_i \Rightarrow X \cup Z \rightarrow A_i.$$

Elagage vers le haut

$$X \not\rightarrow A_i \Rightarrow X \setminus Z \not\rightarrow A_i.$$

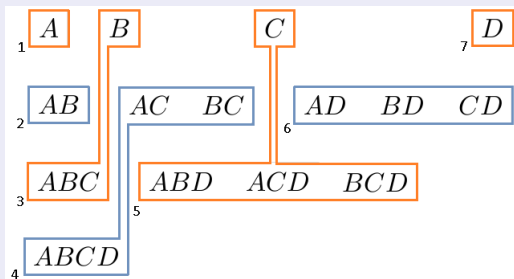
⇒ Parcours en profondeur à privilégier pour élaguer dans les deux sens.

Parallélisation du DFS

Deux candidats X et Y peuvent être évalués en même temps ssi :

- ils sont incomparables ($X \not\subseteq Y$ et $Y \not\subseteq X$)
- on dispose des informations concernant leurs sur-ensembles et sous-ensembles de taille ± 1 (ils déterminent ou non l'attribut cible).

Partitionnement



Les grandes lignes de l'algorithme (inspiré de MineWithRound)

Pour chaque $A_i \in attr(T)$

$r = 1$

$\mathcal{R}_r \leftarrow Child(\emptyset)$

Tant que les 2 prochaines rondes ne sont pas vides

Pour chaque $X \in \mathcal{R}_r$ *boucle parallèle*

Si $T \models (X \rightarrow A_i)$

$\Sigma[i] \leftarrow \Sigma[i] \cup \{X\}$

supprimer de $\Sigma[i]$ tous les Y tq $X \subset Y$

$\mathcal{R}_{r+1} \leftarrow \mathcal{R}_{r+1} \cup RightParents(X)$

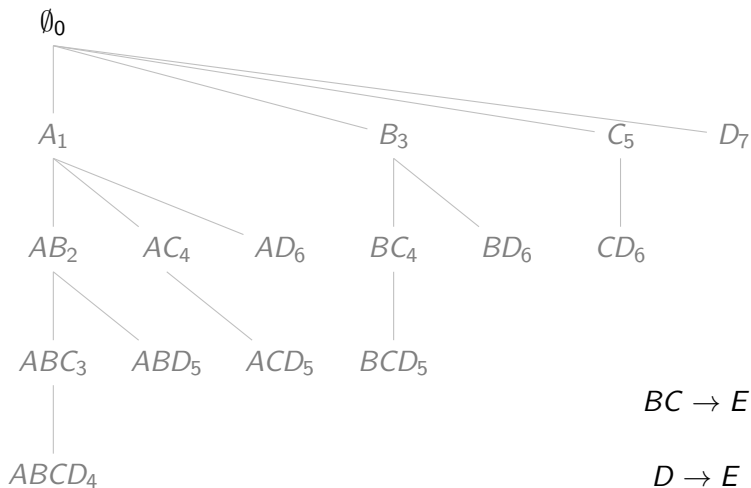
$\mathcal{R}_{r+2} \leftarrow \mathcal{R}_{r+2} \cup \{Sibling(X)\}$

sinon $\mathcal{R}_{r+1} \leftarrow \mathcal{R}_{r+1} \cup \{Child(X)\}$

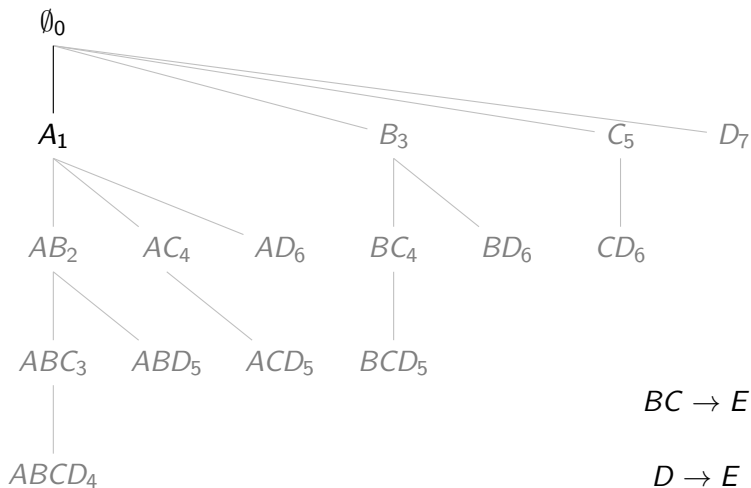
$r \leftarrow r + 1$

Retourner $\Sigma[i]$

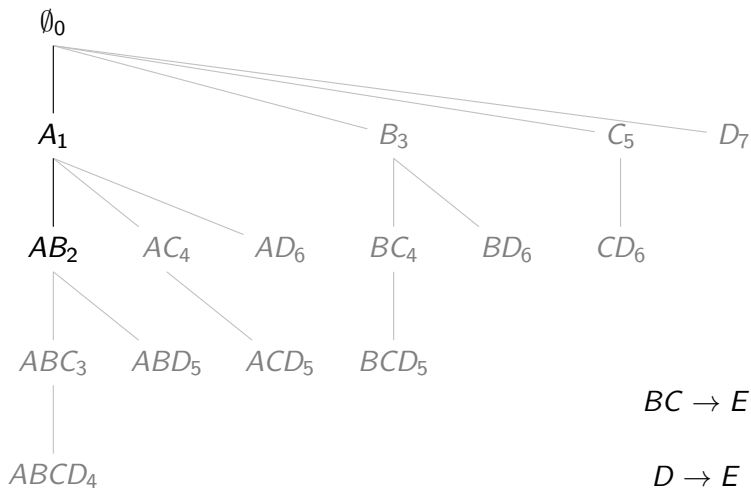
Exemple



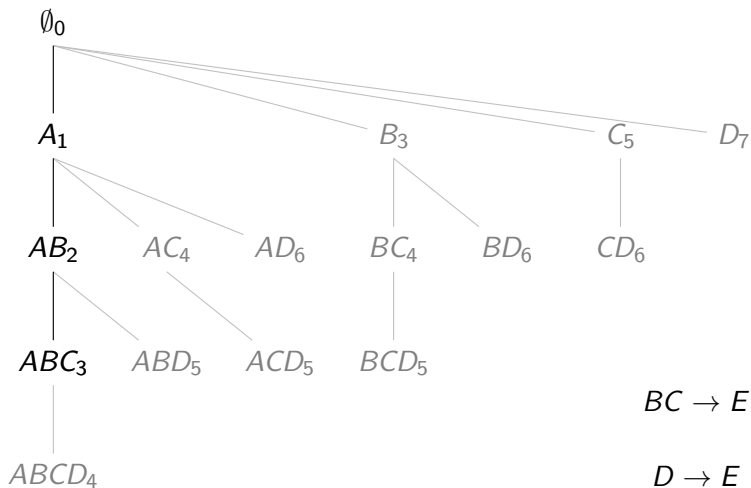
Exemple



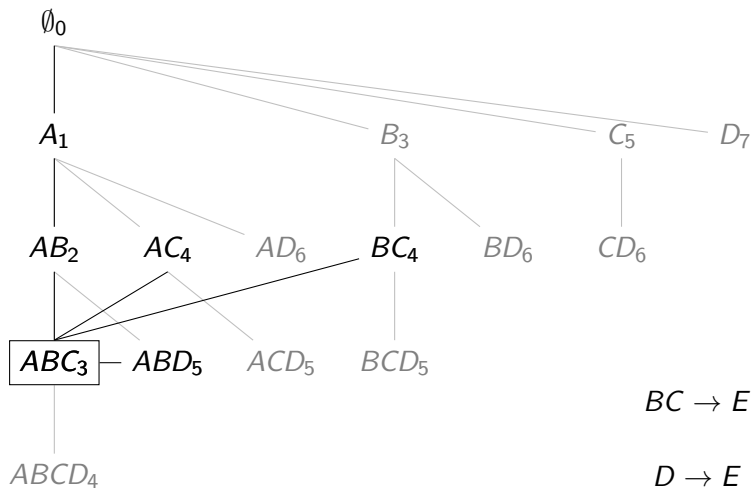
Exemple



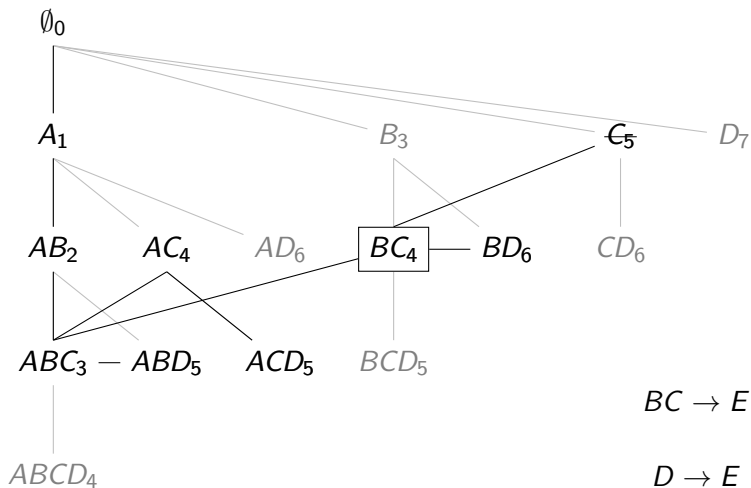
Exemple



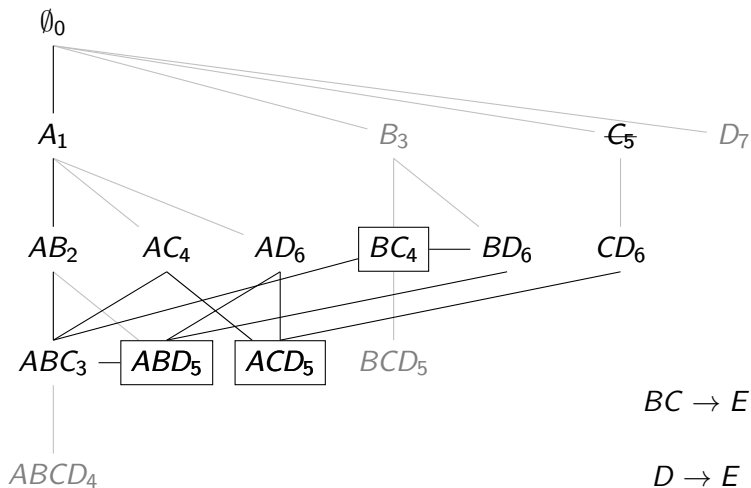
Exemple



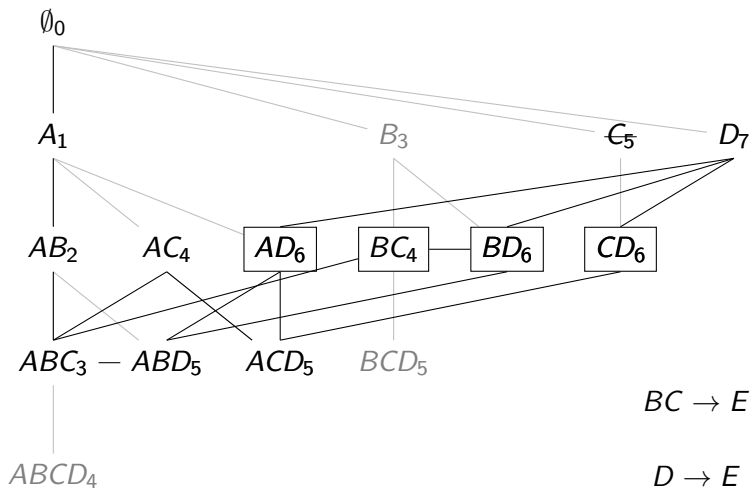
Exemple



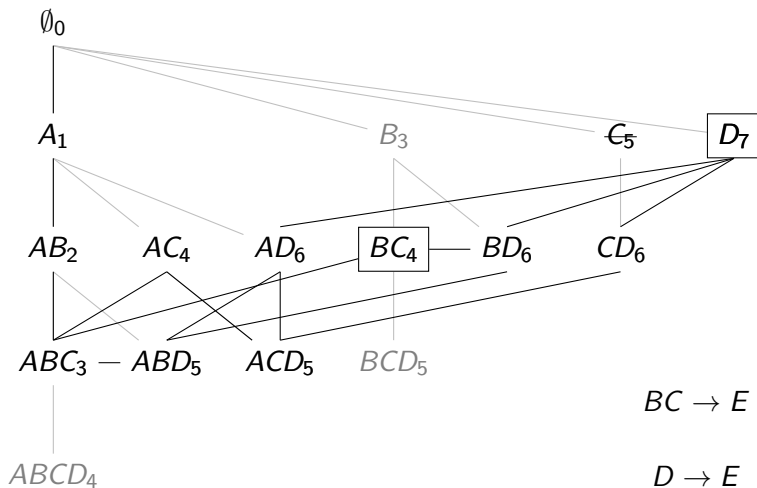
Exemple



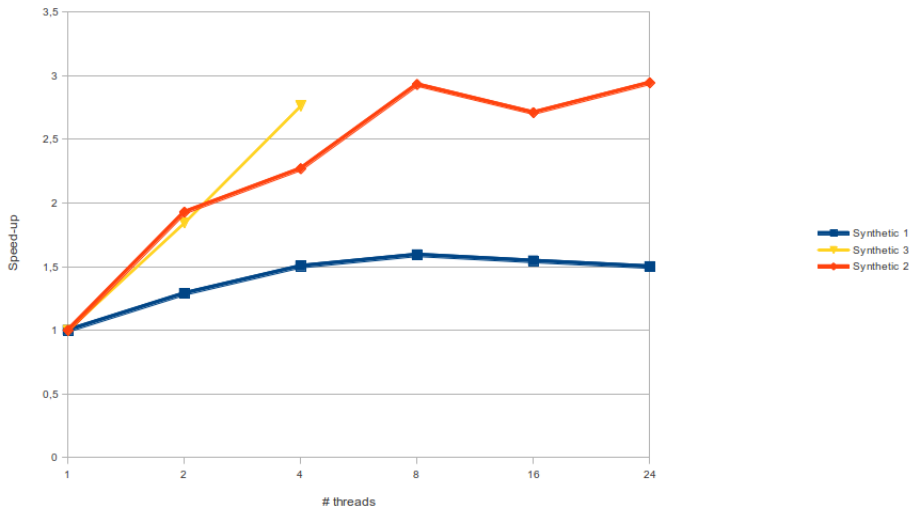
Exemple



Exemple



Accélération des temps de calcul



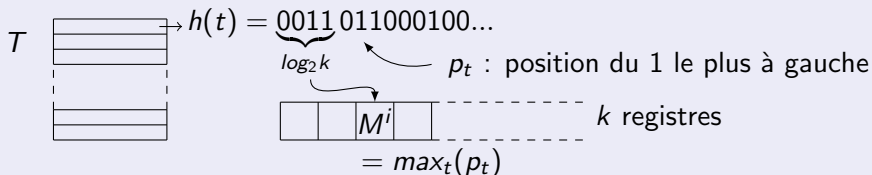
Problème de la mémoire

Pour vérifier la validité des DFs, on utilise la propriété :

$$X \rightarrow A_i \Leftrightarrow |\pi_X| = |\pi_{X \cup \{A_i\}}|$$

C'est l'étape la plus coûteuse (tant en mémoire qu'en temps) de notre algorithme.

La méthode HyperLoglog (Flajolet *et al.* 2007)



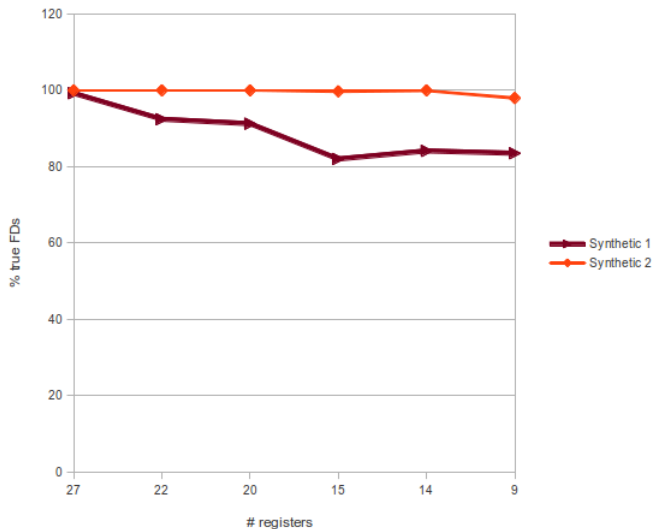
$$|T| = \frac{\alpha_k \times k^2}{\sum_{i=1}^k 2^{-M^i}}$$

HyperLogLog en pratique

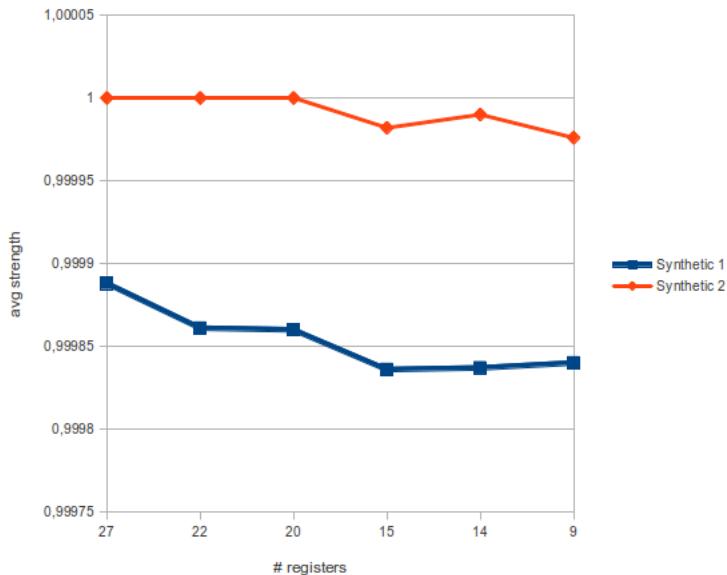
Sur une table à $m = 3 * 10^8$ tuples

	calcul exact	méthode hyperloglog
Complexité en temps	$O(m \log(m))$	$O(m)$
Mémoire utilisée	552 Mo	12 ko
Temps d'exécution	5 min 52	1 min 31
# Tuples différents	6 040 316	6 057 554 (erreur = 0.28 %)

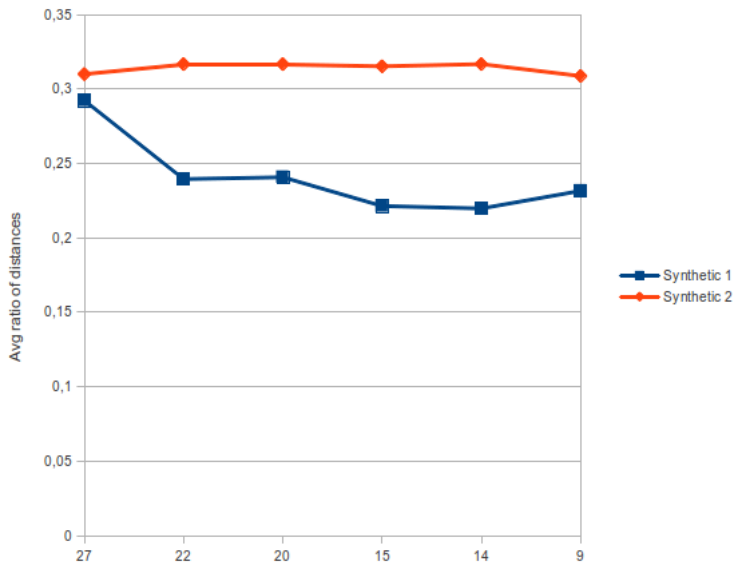
Qualité de l'estimation



Qualité de l'estimation



Qualité de l'estimation



Les dépendances fonctionnelles conditionnelles

Motivation

Si une DF $X \rightarrow A$ n'est pas vérifiée sur l'ensemble de la table, alors il est possible que seules certaines valeurs de X déterminent l'attribut A .

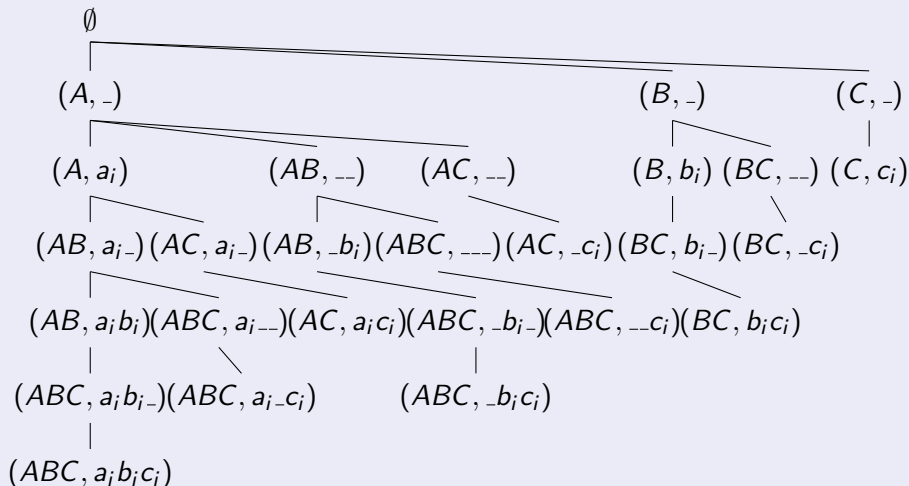
Exemple

Nom	Prénom	Poste
Dupond	Jean	MDC
Dupond	Pierre	MDC
Durant	Jean	MDC
Martin	Pierre	Prof.
Petit	Pierre	Admin.

On a $\text{Prénom} \not\rightarrow \text{Poste}$ mais on a $\text{Prénom} \rightarrow \text{Poste}$ (Jean)

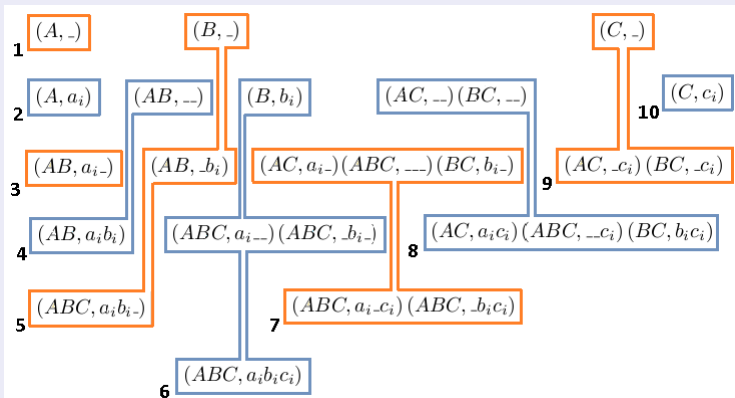
Adaptation aux DFCs

Espace de recherche



Adaptation aux DFCs

Partitionnement



Extensions en cours

Recherche des DF(C)s approximatives

Une DF approximative $X \rightarrow A$ est valide si $\text{confiance}(X \rightarrow A) \geq s$ avec $\text{confiance}(X \rightarrow A) = \sum_{x \in \text{Dom}(X)} \max_{a \in \text{Dom}(A)} (\text{frequence}(X_x A_a))$.

Calcul approché

HyperLogLog + fréquence des k plus fréquents

Conclusion

- La parallélisation permet de **gagner du temps**
- On peut faire une estimation des tailles et accélérer encore les calculs tout en utilisant **moins de mémoire** que dans la version exacte
- On obtient un ensemble de DF(C)s de "bonne" qualité