

ENSEMBLES ET INTERSECTIONS DE FICHIERS

INFO0902 – Structures de données et
algorithmes

Projet 2

20 mars 2015

Aperçu

- **Ensembles**

- Deux implémentations de l'interface définie dans `Set.h`
 - `HashSet.c` (Table de hachage)
 - `TreeSet.c` (Arbre binaire de recherche)
- `ListSet.c` fourni
- Quelques questions d'analyse

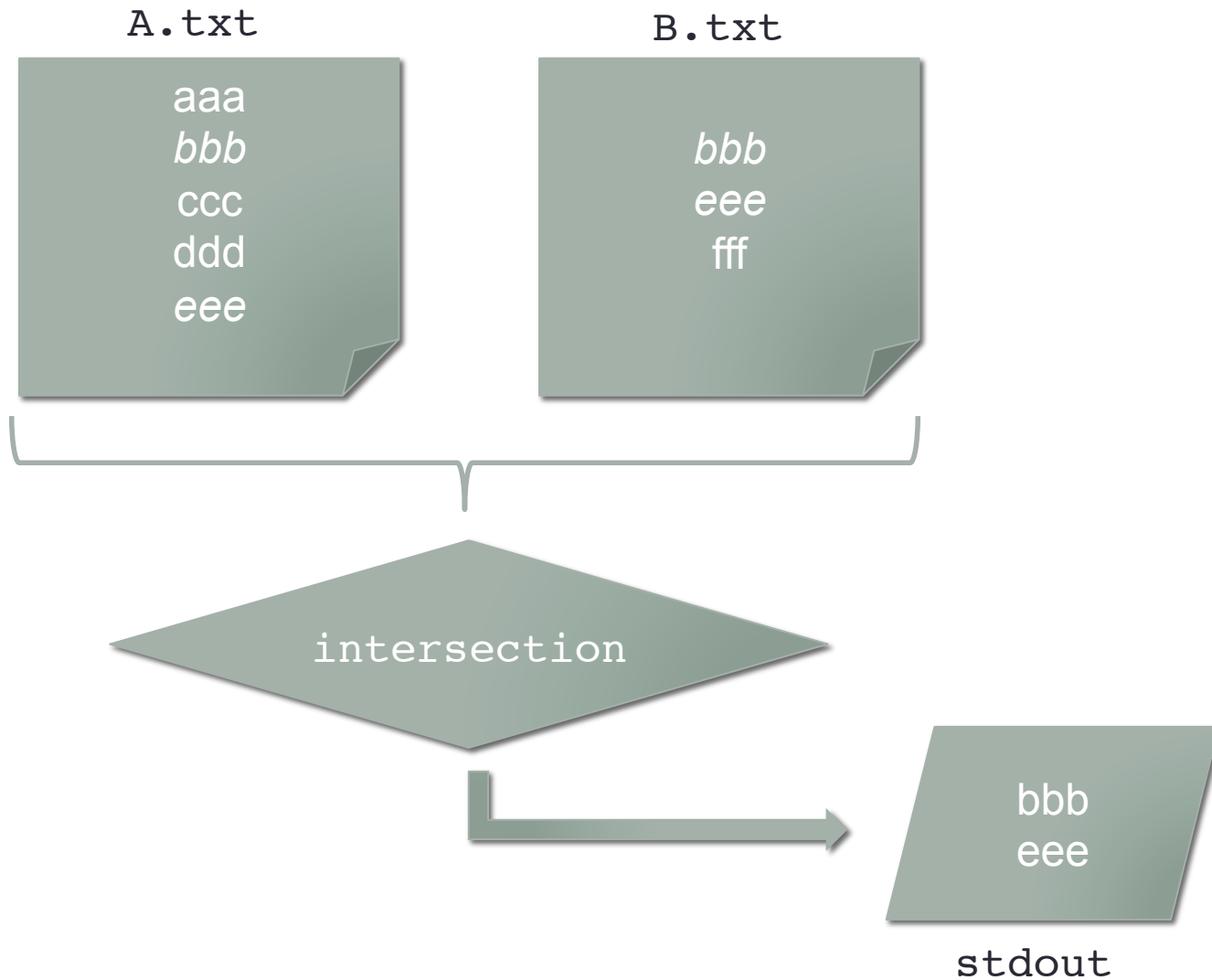
- **Intersection de deux fichiers**

- Implémentation de `getIntersection` déclaré dans `StringArray.h` dans un fichier `Intersection.c`
- Quelques questions d'analyse

Ensembles

- Interface :
 - `createEmptySet`
 - `freeSet`
 - `sizeofSet`
 - `insertInSet`
 - `contains`
 - `setIntersection`
- Remarque
 - Il n'est pas nécessaire de tester si les pointeurs donnés aux fonctions sont NULL (« *valid pointer* »)

Intersection de fichiers



Fichiers fournis

`main.c`

1. Fichiers textes → tableaux
2. Lance l'intersection
3. Affiche le résultat

`StringArray.h`

Bibliothèque:
Tableaux de
`String`

`StringArray.c`

`Set.h`

Interface générique d'un
ensemble de `String`

`ListSet.c`

Implémentation d'un
ensemble de `String`
par liste liée

`A.txt`

`B.txt`

Fichiers de tests
de l'exemple

`English.txt`

`French.txt`

Fichiers de tests
réels

Fichiers à rendre

`Intersection.c`

Algorithme d'intersection de deux tableaux
(`getIntersection` définie dans `stringArray.h`)

`TreeSet.c`

Implémentation de l'ensemble par un arbre
binaire de recherche

`HashSet.c`

Implémentation de l'ensemble par une
table de hachage

`*.pdf`

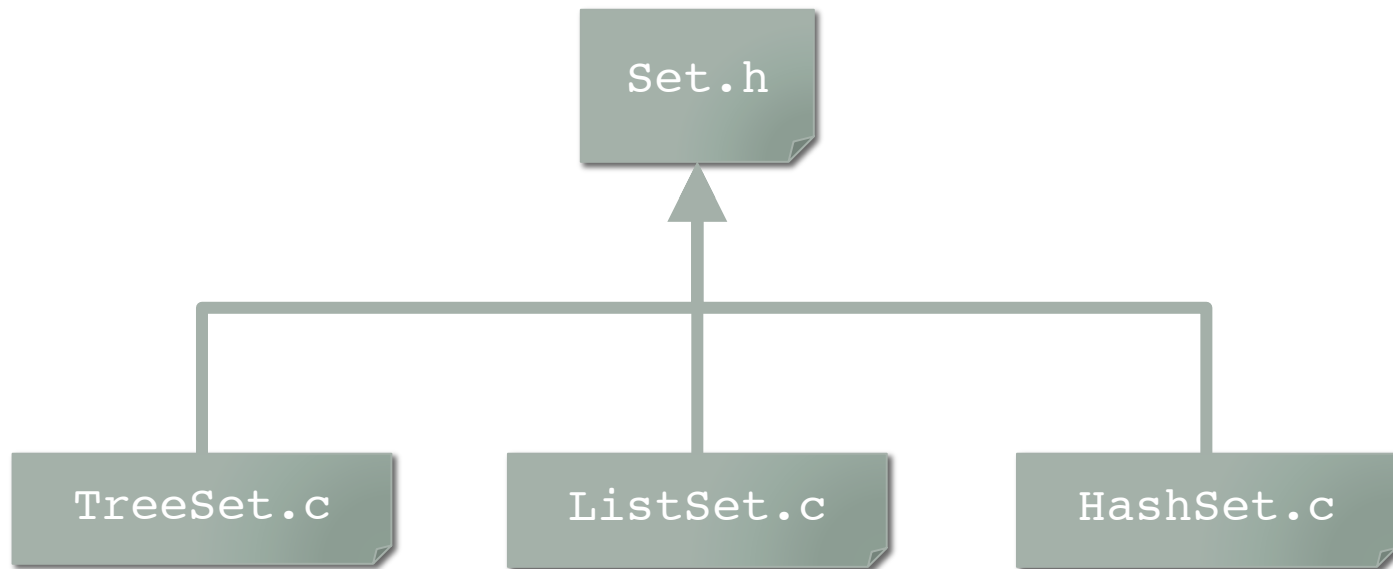
Rapport

Dans une archive `tar.gz`

Remarques

- Il y a deux fonctions liées à l'intersection
 - `setIntersection` (`Set.h` ; pour les ensembles)
 - `getIntersection` (`StringArray.h` ; pour les vecteurs)
- Analyse des ensembles
 - Les simulations peuvent se faire sur base des fichiers `main.c`, `French.txt` et `English.txt` fournis (légères modifications)
 - → Besoin « d'implémenter » la fonction `getIntersection`
- Le choix d'implémentation de l'ensemble se fait à la compilation

Choix statique de l'implémentation



Choix statique de l'implémentation

TreeSet.c

Intersection.c

Main.c

StringArray.c

ListSet.c

HashSet.c

```
gcc main.c StringArray.c TreeSet.c  
Intersection.c --std=c99 -o intersection
```



intersection

Par arbre binaire de recherche

Choix statique de l'implémentation

TreeSet.c

Intersection.c

Main.c

StringArray.c

ListSet.c

HashSet.c

```
gcc main.c StringArray.c ListSet.c  
Intersection.c --std=c99 -o intersection
```

intersection

Par liste chaînée

Choix statique de l'implémentation

TreeSet.c

Intersection.c

Main.c

StringArray.c

ListSet.c

HashSet.c

```
gcc main.c StringArray.c HashSet.c  
Intersection.c --std=c99 -o intersection
```

intersection

Par table de hachage