Programmation fonctionnelle

ensIIE

Semestre 3 – 2026/27

Institut Mines-Télécom
ensIIE école associée
université PARIS-SACLAY

ensIIE

www.ensiie.fr

Ensembles sur les types composés

chaînes de caractères, listes, ...

Possibilité de faire mieux que les AVL

- utiliser la structure des types

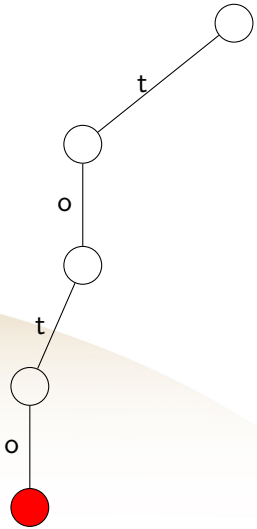
Solution : arbres à fils multiples sur les éléments internes

Arbre préfixe (trie)



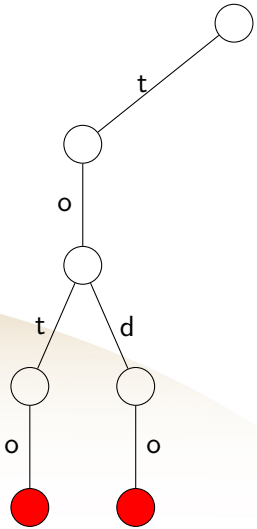
Ensemble vide

Arbre préfixe (trie)



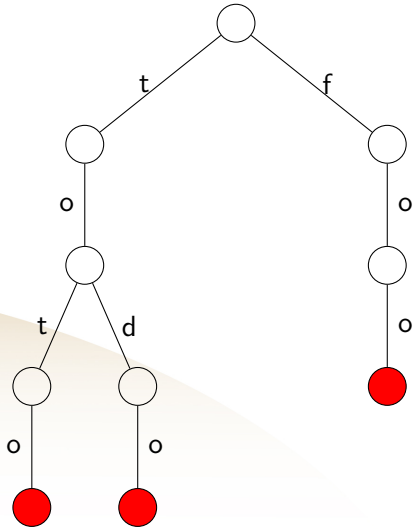
Ajout de “toto”

Arbre préfixe (trie)



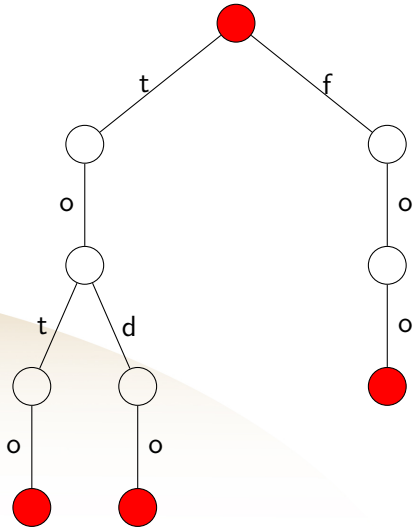
Ajout de “todo”

Arbre préfixe (trie)



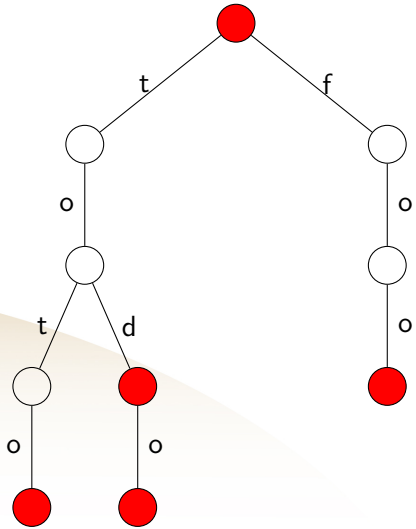
Ajout de “foo”

Arbre préfixe (trie)



Ajout de “”

Arbre préfixe (trie)



Ajout de “tod”

Implémentation

Fils étiquetés par des “lettres”

- utilisation d’un dictionnaire
  - clés : “lettres”
  - valeurs : arbres

Booléen pour indiquer présence dans l’ensemble

```
type tree =
  { present : bool; children : tree Map.t }
```

Les “lettres” doivent être ordonnées pour pouvoir utiliser Map.

Complexité

ℓ longueur de la liste à rechercher/ajouter/supprimer

- vide : $\mathcal{O}(1)$
- appartenance : $\mathcal{O}(\ell)$
- ajout : $\mathcal{O}(\ell)$
- cardinal : $\mathcal{O}(n)$
- union : $\mathcal{O}(n_1 + n_2)$
- intersection : $\mathcal{O}(n_1 + n_2)$
- retrait : parcours $\mathcal{O}(\ell)$
- agrégation $\mathcal{O}(n)$
- égalité $\mathcal{O}(n_1 + n_2)$

Dictionnaires

On remplace le champ `present : bool`
par `value : 'a option`