

Projet de Logique & Structures Informatiques

Arbre 2-3-4

À déposer sur la plateforme UniversiTICE avant le **dimanche 9 Mai 2021 au plus tard**.

À réaliser en groupe de deux étudiants au maximum.

Les dates des soutenances seront fixées ultérieurement.

1 But du projet et définition de base

Le but de ce projet est d'implanter les arbres 2-3-4 pour manipuler les parties d'un ensemble E fini et ordonné. Cette structure est définie comme suit:

Définition 1 (Arbre 2-3-4)

Les arbres 2-3-4 obéissent aux propriétés suivantes:

1. les nœuds sont étiquetés par 1, 2 ou 3 éléments de E distincts et ordonnés du plus petit au plus grand;
2. chaque nœud d'étiquettes $e_1 < \dots < e_k$ ($0 < k < 4$) admet $k + 1$ sous-arbres A_i ($0 < i < k + 2$) tels que tous les éléments de A_i sont plus petits que e_i ($0 < i < k + 1$) et tous les éléments de A_i ($1 < i < k + 2$) sont plus grands que e_{i-1} ;
3. tous les nœuds externes sont au même niveau.

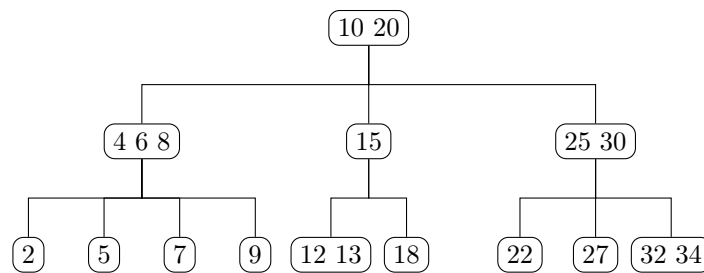


Figure 1: Un arbre 2-3-4.

Il résulte de la définition que les arbres 2-3-4 sont bien équilibrés. La recherche d'une étiquette est en $O(\log n)$ où n est le nombre d'étiquettes. À tout arbre 2-3-4, on peut faire correspondre un arbre bicolore. Il vous est demandé d'implanter les ensembles à l'aide des arbres 2-3-4. **Il ne faudra considérer que l'algorithmique sur les arbres 2-3-4.**

2 Définition du type et propriétés de base

Question 1 (Type de données)

En OCaml:

1. Proposer un type de données `arbre234` encodant les arbres 2-3-4.
2. Vérifier si une valeur de type `arbre234` respecte bien les propriétés de la définition 1.
3. Rechercher une valeur dans un arbre 2-3-4.

3 Conversion

On considère les deux fonctions suivantes:

- la fonction `a234_vers_abic` qui transforme un arbre 2-3-4 vers un arbre bicolore;
- la fonction `abic_vers_a234` qui transforme un arbre bicolore en un arbre 2-3-4.

Pour tout $i = 1, \dots, 4$, soient a_i un arbre 2-3-4 et b_i son image par la fonction `a234_vers_abic`. Le passage d'un arbre 2-3-4 en un arbre bicolore se réalise récursivement par fission selon le type de nœud (Figure 2).

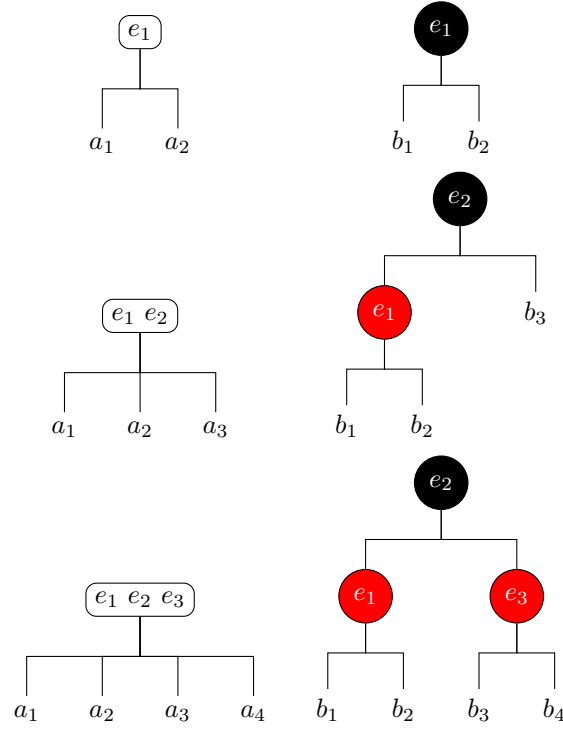


Figure 2: Les fissions de nœud d'arbres 2-3-4.

Pour tout $i = 1, \dots, 4$, soient b_i un arbre bicolore et a_i son image par la fonction `abic_vers_a234`. Le passage d'un arbre bicolore en un arbre 2-3-4 se réalise par fusion de nœuds selon la couleur (Figure 3).

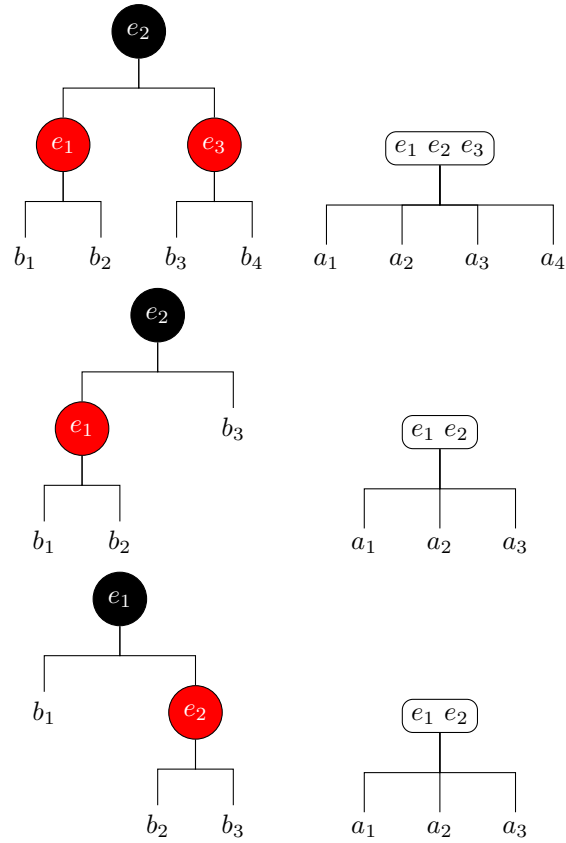


Figure 3: Les fusions de nœuds d'arbres bicolores.

Un nœud noir qui n'est pas suivi d'un nœud rouge est transformé en un nœud à une seule étiquette (Figure 4).

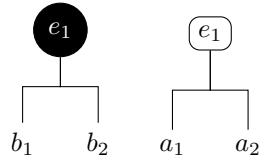


Figure 4: Le cas d'un nœud noir sans descendant rouge.

Question 2 (Conversions)

1. Montrer que les transformations de la Figure 2 convertissent un arbre 2-3-4 en un arbre bicolore.
2. Montrer que les transformations des figures 3 et 4 convertissent un arbre bicolore en un arbre 2-3-4.
3. Écrire les fonctions `a234_vers_abic` et `abic_vers_a234` en OCaml.

4 Algorithmique des arbres 2–3–4

4.1 Insertion dans un arbre 2–3–4

Pour insérer une étiquette, il faut parcourir l'arbre 2–3–4 comme on le ferait pour un arbre binaire de recherche pour déterminer le nœud externe concerné pour l'insertion. Si ce nœud contient une ou deux étiquettes, il reste un emplacement. Si ce nœud a trois étiquettes, il faut le casser et rééquilibrer. Pour cela, on peut prévoir un éclatement anticipé des nœuds à trois étiquettes situées sur le chemin de la racine au nœud externe lors de la descente dans l'arbre pour toujours se retrouver dans le cas d'un nœud à une ou deux étiquettes. Dans les exemples suivants, la valeur e_1 est une des étiquettes du nœud considéré lors du parcours de la racine vers l'emplacement où il faudra insérer (Figure 5).

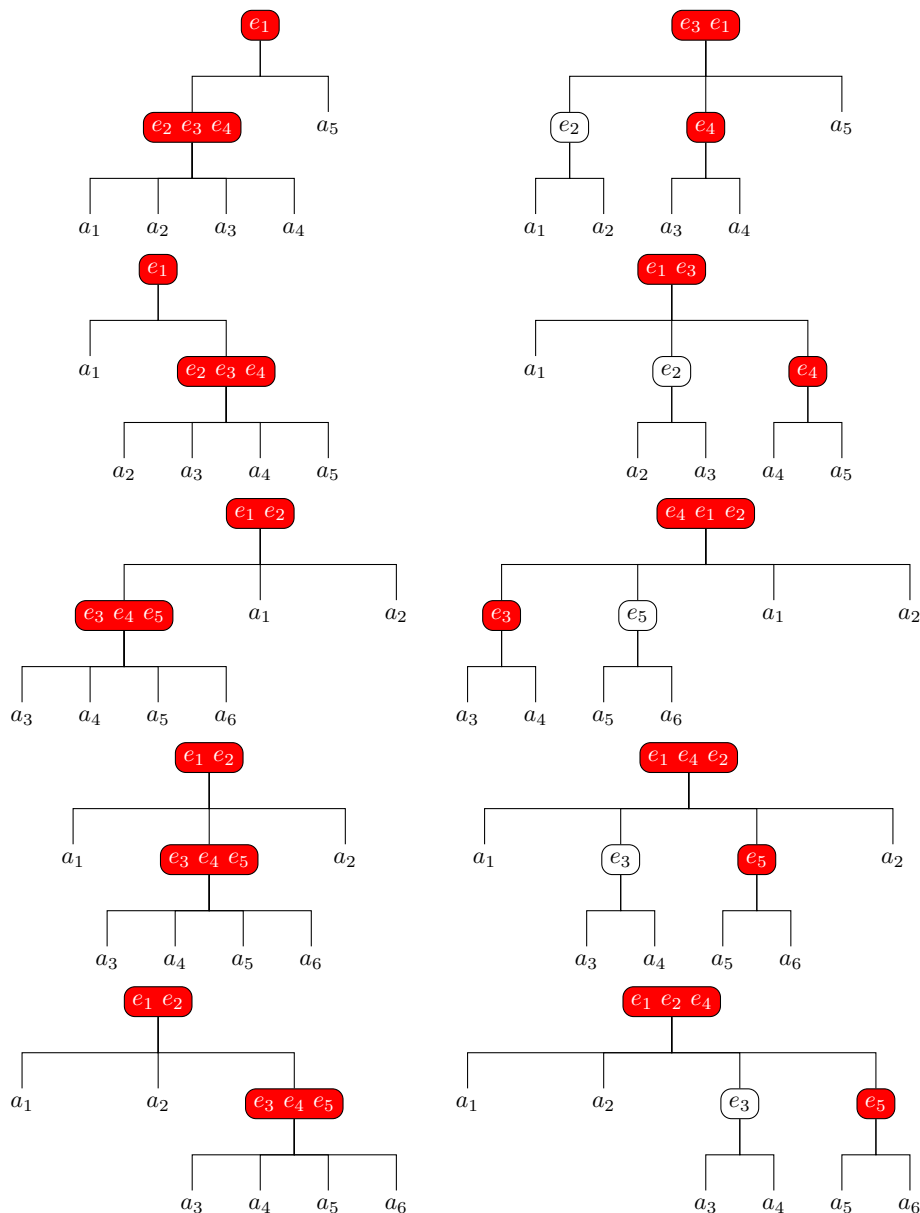


Figure 5: L'éclatement de nœuds d'arbre 2–3–4.

Un traitement spécifique est réalisé pour une racine étiquetée par 3 valeurs (Figure 6).

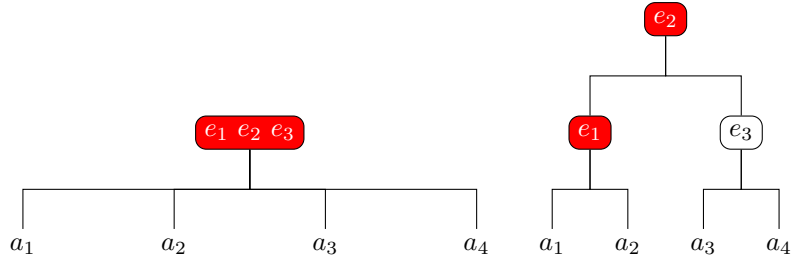


Figure 6: L'éclatement de la racine.

Question 3 (Insertion)

1. Étudier la correction et la complexité de l'algorithme d'insertion dans un arbre 2-3-4.
2. Implanter cet algorithme en OCaml.
3. Créer un arbre 2-3-4 en déterminant de façon aléatoire des entiers naturels.

4.2 Suppression dans un arbre 2-3-4

Une première étape de l'algorithme de suppression, comme pour les arbres binaires de recherche, consiste à localiser le nœud étiqueté par la valeur e à supprimer (s'il existe) puis à substituer à e la plus petite valeur x supérieure à e qui se trouve sur un nœud externe. Tout au long de ce parcours, on transformera tout nœud à une seule étiquette (sauf éventuellement la racine) en un nœud à plusieurs étiquettes tout en restructurant l'arbre. Une fois que ce traitement est effectué, la suppression de x , qui étiquette alors un nœud externe avec au moins deux valeurs, est immédiate (Figure 7).

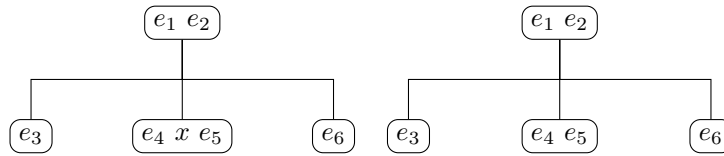


Figure 7: Un cas de suppression immédiate.

Dans le parcours, on va s'arranger à ce que la racine du sous-arbre considéré soit étiquetée par au moins deux valeurs. Cependant, un traitement spécifique doit être fait lorsque, en début de parcours, la racine de l'arbre n'admet qu'une étiquette comme dans les exemples des figures 8 et 9. Pour l'exemple de la Figure 8, la racine elle-même est transformée en un nœud contenant plusieurs étiquettes. Pour l'exemple de la Figure 9, c'est le descendant sur le parcours menant à x qui est transformé en un nœud à plusieurs étiquettes.

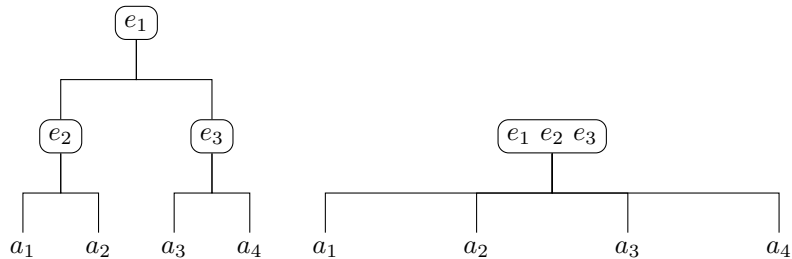


Figure 8: Un exemple pour la racine.

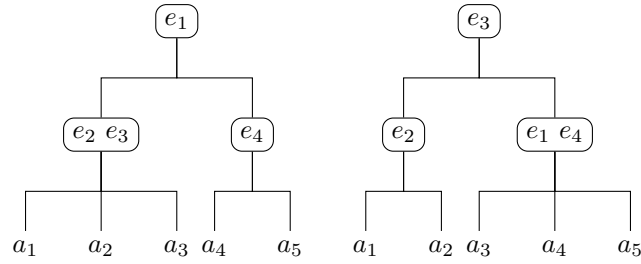


Figure 9: Un exemple pour la racine.

Ensuite, sur le parcours, il suffit de transformer tout descendant immédiat, étiqueté par une seule valeur, d'un nœud à plusieurs étiquettes. La Figure 10 liste quelques cas possibles.

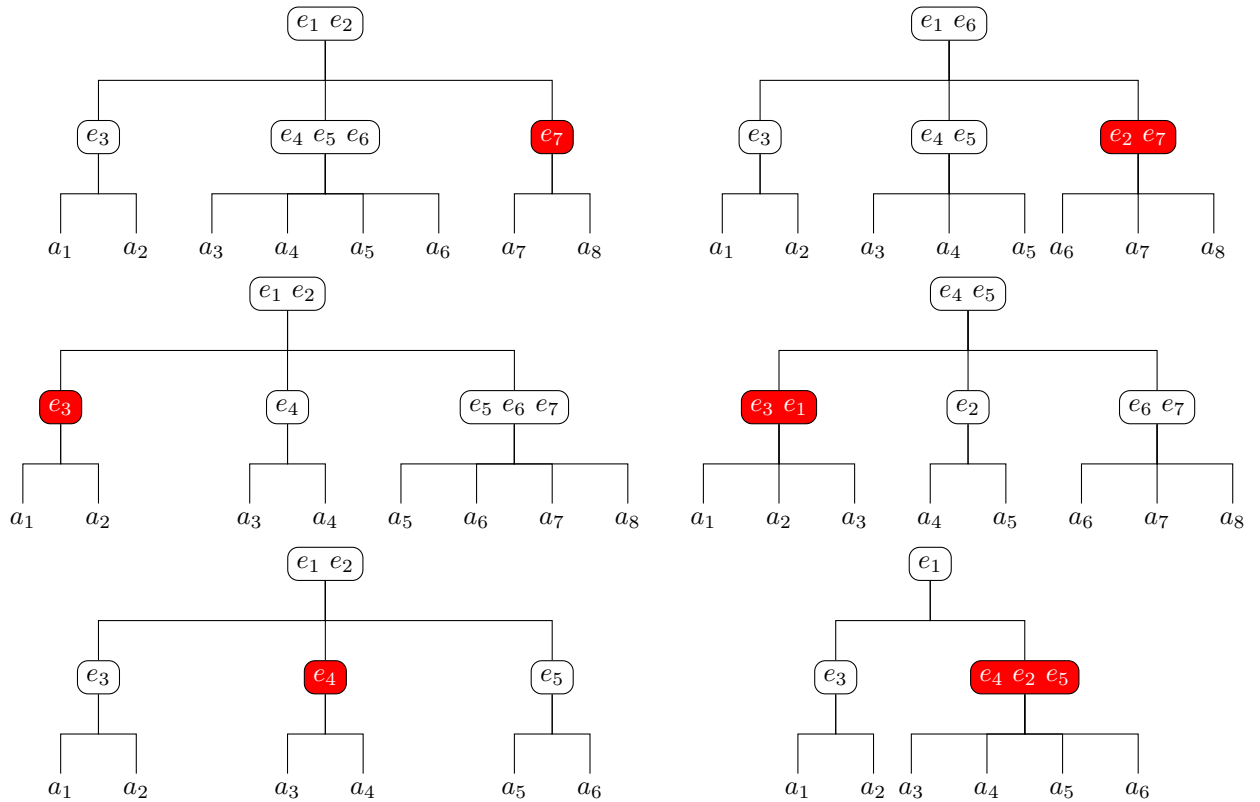


Figure 10: Des cas de transformation.

Question 4 (Suppression)

1. Proposer un algorithme de suppression de valeurs pour un arbre 2-3-4 basé sur la méthode précédente.
2. Étudier la correction et la complexité de l'algorithme de suppression dans un arbre 2-3-4.
3. Implanter cet algorithme en OCaml.

5 Manipulation d'ensembles

Question 5 (Fonctions ensemblistes)

À l'aide de l'implantation précédente des arbres 2-3-4, proposer une implantation des fonctions suivantes et montrer leurs corrections:

1. union de deux ensembles;
2. intersection de deux ensembles;
3. différence de deux ensembles;
4. différence symétrique de deux ensembles;
5. test d'égalité de deux ensembles;
6. test d'inclusion d'un ensemble dans un autre.

Question 6 (Fonctions de haut niveau)

- À l'aide de l'implantation précédente des arbres 2-3-4, proposer des équivalents aux fonctions `List.fold_left` et `List.fold_right`;
- utiliser ces fonctions pour calculer:
 - le cardinal d'un ensemble;
 - pour un élément x donné, un couple (E_1, E_2) d'ensembles contenant respectivement les éléments plus petits que x et les éléments plus grands que x (fonction de séparation);
 - pour une fonction Booléenne f , l'ensemble des éléments x d'un ensemble tels que $f(x)$ soit vrai (fonction de filtre).

6 Déroulement de la soutenance

Vous disposerez de 10 minutes pour présenter votre projet. S'ensuivra une séance de 5 minutes de questions.

7 Barème indicatif

- Code: 24 points
 - Question 1:
 - * type: 1 point
 - * propriétés: 2 points
 - * recherche: 1 point
 - Question 2:
 - * preuves: 2 points
 - * fonctions de conversion: 2 points
 - Question 3:
 - * étude de la correction et de la complexité: 2 points
 - * insertion: 3 points
 - * création d'arbres : 1 point
 - Question 4:

- * algorithme de suppression: 3 points
- * étude de la correction et de la complexité: 2 points
- * implantation: 1 point
- Question 5:
 - * union, intersection, différence, différence symétrique: 1 point
 - * égalité, inclusion: 1 point
- Question 6:
 - * équivalents fold: 1 point
 - * séparation, filtre: 1 point
- Qualité du code (paradigme fonctionnel, commentaires, *etc.*): 3 points
- Rapport et soutenance: 3 points

8 Documents à déposer

Seule une archive dans un format interopérable sera déposée sur la plateforme. Elle contiendra:

1. les sources de votre implantation;
2. un ensemble d'exemples judicieusement choisi;
3. un rapport avec une notice d'utilisation, des explications sur les choix d'implantation, le détail des problèmes rencontrés ainsi que les preuves des algorithmes utilisés.