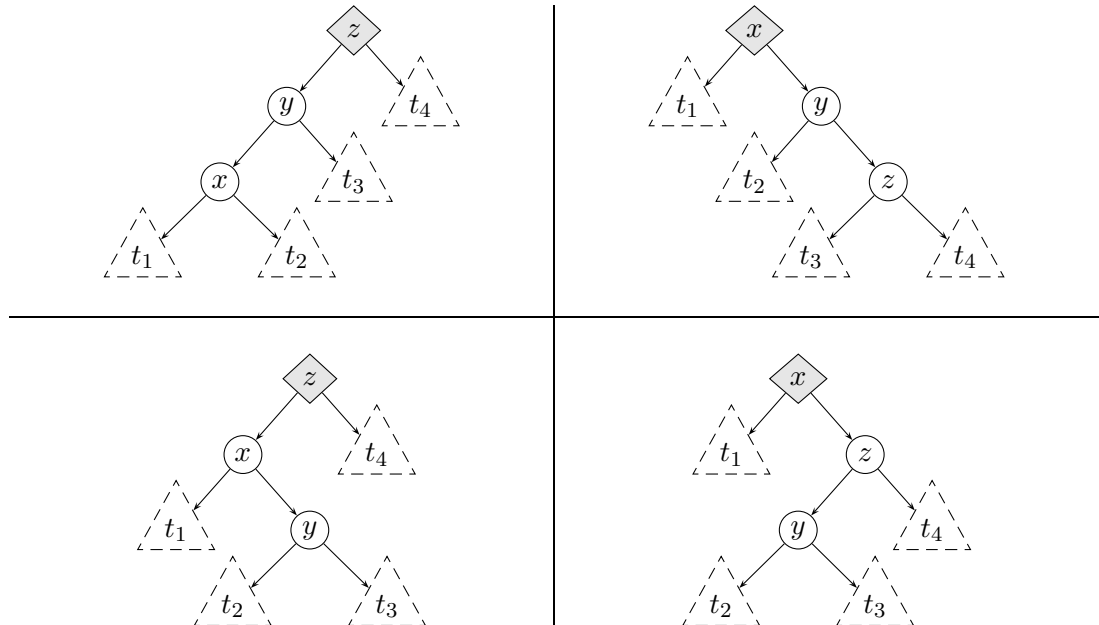
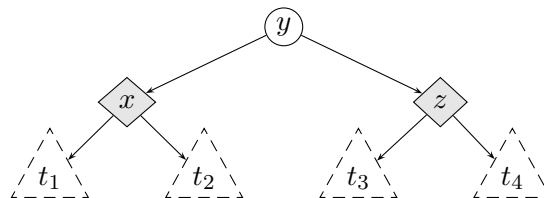


□ **Exercice 1** : *Équilibrage d'un arbre rouge-noir*

1. Pour insérer un noeud dans un arbre rouge-noir, on commence par utiliser l'algorithme d'insertion usuel dans un ABR et on attribut au nouveau noeud la couleur *rouge*. Quel est alors le seul conflit possible ? (on appellera un tel conflit un *conflit rouge-rouge*).
2. Si le conflit rouge-rouge se situe à la racine, donner une méthode simple pour le résoudre.
3. Si le conflit n'est pas situé à la racine, justifier qu'on se trouve dans l'un des quatre cas suivants où les noeuds rouges sont représentés dans un cercle et les noeuds noirs dans un losange grisé :

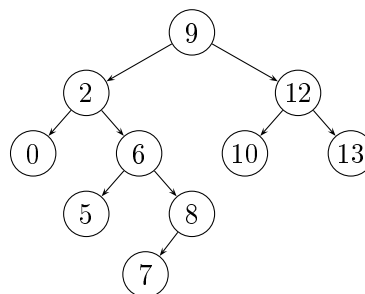


4. Montrer qu'en effectuant une ou plusieurs rotations, ces arbres se ramènent à

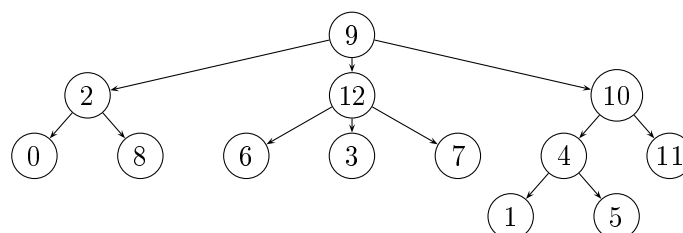


□ **Exercice 2** : *Conversion entre arbre binaire et arbre généralisé*

1. Convertir en arbre généralisé l'arbre binaire suivant

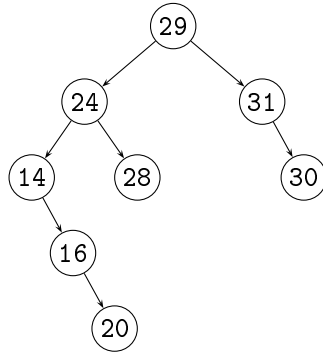


2. Convertir en arbre binaire l'arbre généralisé suivant



□ **Exercice 3** : *Sérialisation*

1. Donner l'ordre des noeuds lors des parcours prefixe, infixe et postfixe de l'arbre suivant :



2. On cherche à présent à reconstruire un arbre en connaissant un ou plusieurs de ses parcours. Montrer sur un exemple que deux arbres ayant les mêmes parcours prefixe et postfixe peuvent être différents.
3. Montrer (en le construisant) qu'un seul arbre a pour préfixe [0 ; 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7] et pour parcours infixe [1 ; 2 ; 0 ; 5 ; 4 ; 6 ; 3 ; 7]
4. Comment savoir si deux listes de valeurs correspondent aux parcours prefixe et infixe d'un arbre binaire ?
5. Ecrire une fonction qui prend en argument deux listes (le parcours prefixe et le parcours infixe) et qui renvoie l'arbre binaire correspondant. On supposera que les étiquettes de l'arbre sont des entiers tous différents.

☛ On pourra commencer par écrire :

- Une fonction `separe_valeur` de signature `int -> int list -> int list * int list` qui prend en argument un entier, une liste contenant cet entier et renvoie un couple de liste : les éléments situés avant (resp après) cette valeur
- Une fonction `separe_nb` de signature `int -> int list -> int list * int list` qui prend en argument un entier `n` et une liste et renvoie un couple de listes : les `n` éléments situés après le premier puis le reste de la liste.