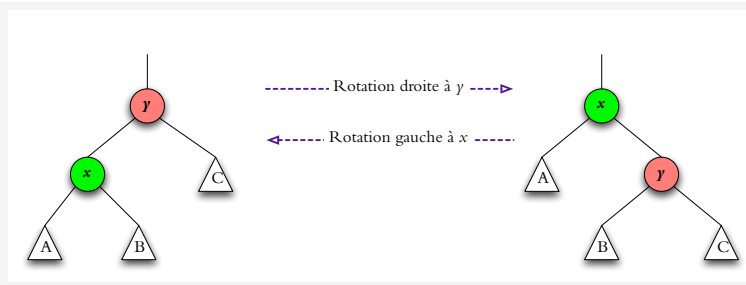


14 Arbres splay

14.1 Rotations

W_(en)

Arbres équilibrés : on maintient une condition qui assure que les sous-arbres ne sont trop différents à aucun nœud. Si l'on veut maintenir une condition d'équilibre, il faudra travailler un peu plus à chaque (ou quelques) opérations — mais on veut toujours maintenir $O(\log n)$ par opération.



La technique principale dans l'établissement de l'équilibre est la **rotation**. Les rotations (gauche ou droite) — préservent la propriété des arbres de recherche et prennent seulement $O(1)$.

14.2 Déploiement (splaying).

W_(en)

Idée principale : on fait des rotations sans tests spécifiques pour l'équilibre. Quand on accède à nœud x , on performe des rotations sur le chemin de la racine à x pour monter x à la racine.

```

    SPLAY( $x$ ) // déploiement du nœud  $x$ 
    S1  tandis que  $x$  n'est pas la racine faire
    S2    si parent( $x$ ) = racine alors zig ou zag
    S3    sinon
    S4      si  $x$  et parent( $x$ ) au même coté (gauche-gauche ou droit-droit) alors zig-zig ou zag-zag
    S5      sinon zig-zag ou zag-zig
  
```

Choix de x pour déploiement :

- ★ insert : x est le nouveau nœud
- ★ search : x est le nœud où on arrive à la fin de la recherche
- ★ delete : x est le parent du nœud *effectivement* supprimé. Attention : c'est le parent ancien du successeur (ou prédécesseur) si on doit supprimer un nœud à deux enfants (logique : échange de nœuds, suivi par la suppression du nœud sans enfant).

Théorème 14.1. *Le temps pour exécuter une série de m opérations avec search, insert et delete en commençant avec l'arbre vide est de $O(m \log n)$ où n est le nombre d'opérations d'insert dans la série.*

Il s'agit d'une structure d'auto-ajustement (*self-adjusting*). On a un temps de calcul efficace dans le sens amorti.

→ il peut arriver que l'exécution est très rapide au début et tout d'un coup une opération prend très long — ceci peut être problématique dans le contexte d'opérations online.

→ cela ne fait aucune différence pour le temps de calcul d'un **algorithme** qui utilise la structure

