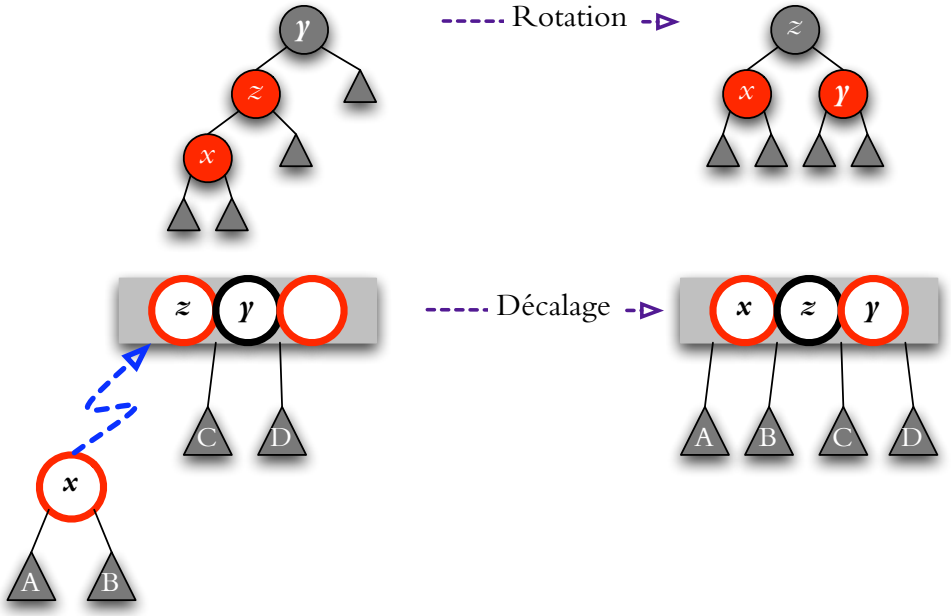
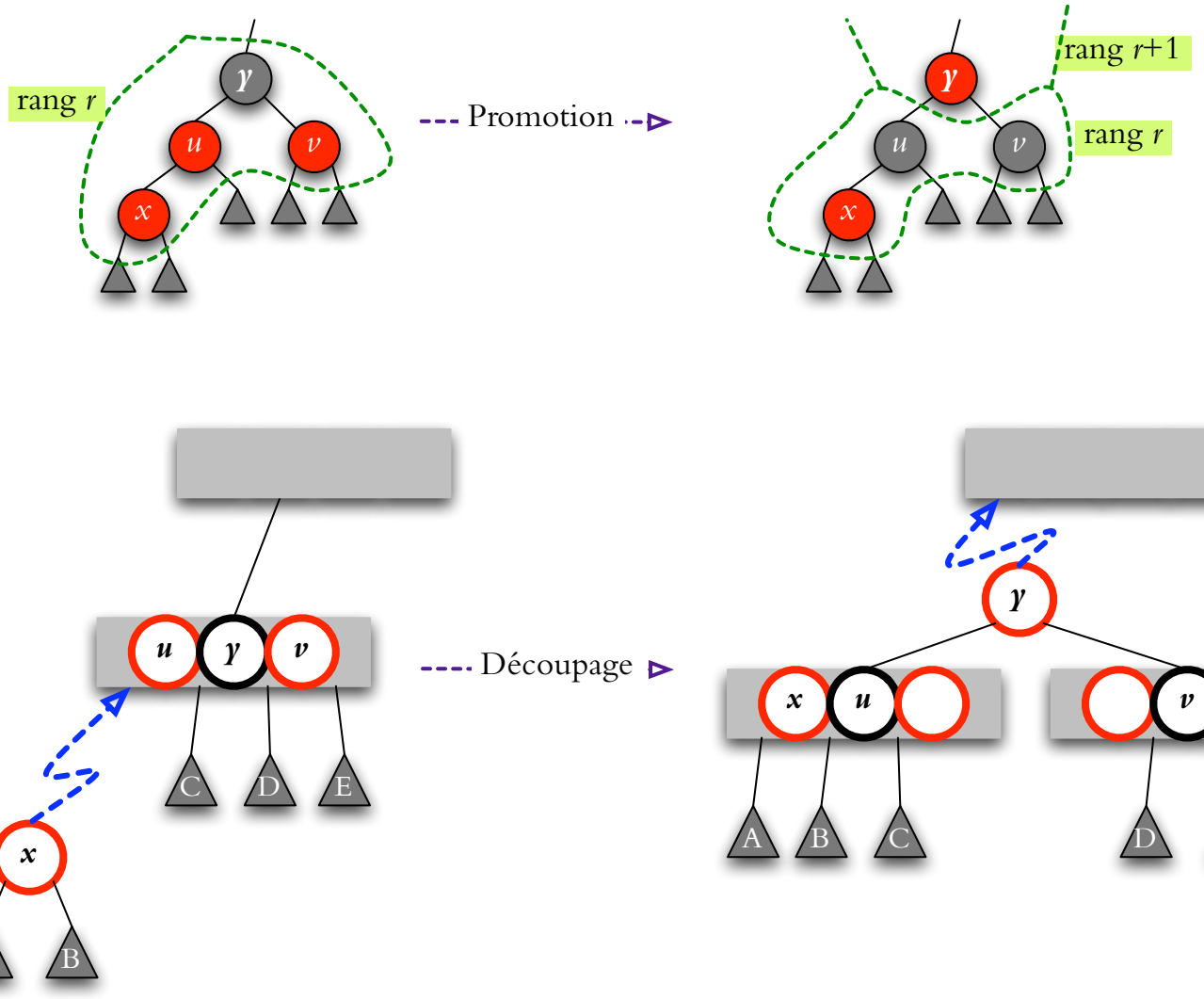
Arbre RN $\leftrightarrow$ arbre 2-3-4

Qu'est-ce qui se passe lors d'une insertion ?
On crée un nœud rouge : promotions+rotations en ascendant vers la racine

Rotation : nœud noir avec un enfant rouge et son grand-enfant rouge transformé en un nœud noir avec deux enfants rouges

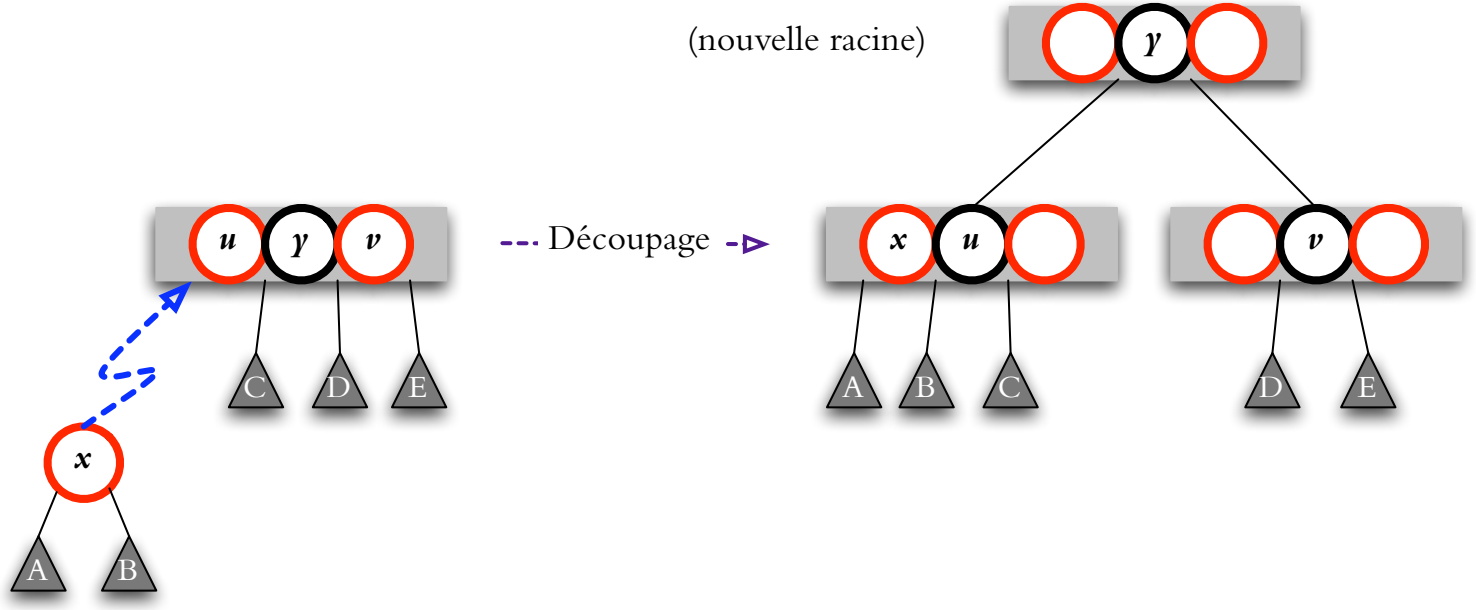


Arbre RN $\leftrightarrow$ arbre 2-3-4 (promotions)



Arbre RN $\leftrightarrow$ arbre 2-3-4 (cont)

Cas spécial : promotion de la racine



$\Rightarrow$ la hauteur de l'arbre croît par le découpage de la racine

(arbre binaire de recherche : la hauteur croît par l'ajout de feuilles)

Recherche dans mémoire externe

Supposons qu'on veut stocker un grand nombre de données : **enregistrements** avec **clés**

On veut minimiser l'accès au disque dur : 10^5 fois plus de temps que l'accès à la mémoire principale

Stocker un arbre rouge et noir ??

Arbre 2-3-4 est plus efficace : on modifie «quelques» nœuds seulement

Comment améliorer ?
on généralise à M sous-arbres au lieu de 4.

Une autre bonne idée : on va mettre les données aux feuilles, et ne stocker que des clés à des nœuds internes («B^+-arbre»)

B-arbre

Données stockées aux feuilles : L enregistrements à une feuille

Clés stockés aux nœuds internes : $(M - 1)$ clés à un nœud interne, M enfants

Clé i : valeur minimale dans le sous-arbre $(i + 1)$

Racine : $2..M$ enfants
Nœuds internes : $\lceil M/2 \rceil..M$ enfants (taille : $M \cdot |\text{clé}| + (M - 1) \cdot |\text{pointeur}|$)
Feuilles : $\lceil L/2 \rceil..L$ enregistrements (taille : $L \cdot |\text{enregistrement}|$)

Feuilles ont la même profondeur

Choix de M et L : on utilise un bloc (taille typique : 4k, 8k, . . .) par nœud

B-arbre

Thm. La hauteur h de B arbre est bornée par

$$h \leq 1 + \log_{\lceil M/2 \rceil} \frac{N}{2} \leq 1 + \frac{\lg \frac{n}{L}}{\lg M - 1}$$

où N est le nombre de feuilles et n est le nombre d'enregistrements.

Exemple (du livre) : blocs de 8k, clés de 32 octets, enregistrements de 256 octets, $M = 228$, $L = 32$
$h = 4$ suffit jusqu'à $N = 2.9 \cdot 10^6$ ou $n = 47 \cdot 10^6$

$\Rightarrow$ nombre d'accès au disque est determiné par h : très peu (en plus, on peut garder la racine et peut-être même le premier niveau en mémoire principale)

B-arbre (cont)

Insertion d'un enregistrement : s'il y a de la place dans la feuille, aucun problème

s'il n'y a pas de place : **débordement** de la feuille

solution : découpage de la feuille $\rightarrow$ éléments distribués en deux feuilles de tailles $\lfloor \frac{L}{2} \rfloor + 1$ et $\lceil \frac{L}{2} \rceil$.

peut causer un débordement au parent : découpage si nécessaire en ascendant vers la racine

$\Rightarrow$ la hauteur croît en découpant la racine

B-arbre (cont)

Suppression d'un élément : si la feuille est toujours assez complète, aucun problème

et si le nombre d'éléments tombe en-dessous de $\lceil L/2 \rceil$?

1. prendre des éléments des sœurs immédiates

2. si elles sont au minimum, alors fusionner les feuilles $\longrightarrow$ le parent perd un unfant

3. continuer avec le parent de la même manière si nombre d'enfants $< \lceil M/2 \rceil$

$\Rightarrow$ la hauteur décroît en enlevant la racine (quand elle a un enfant seulement)