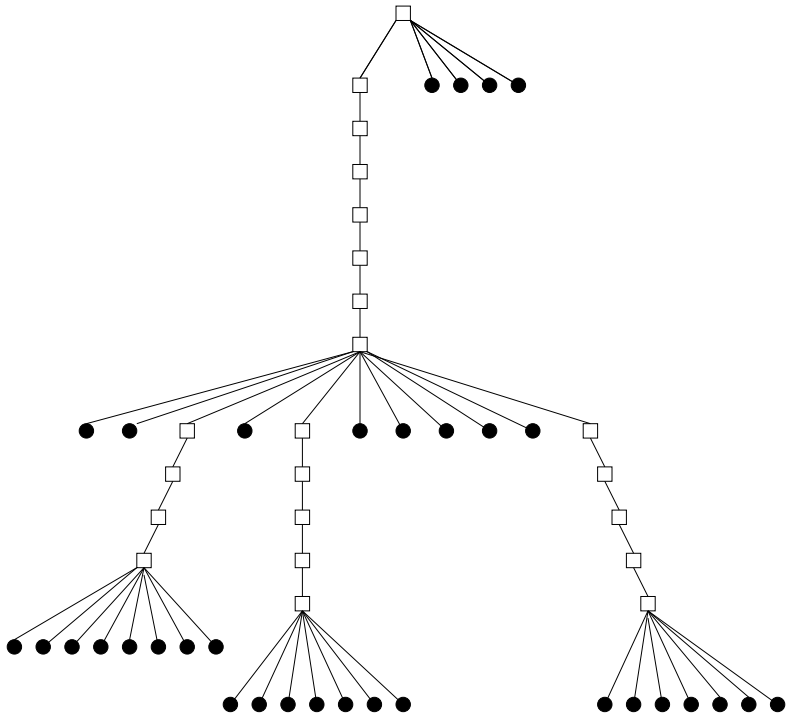
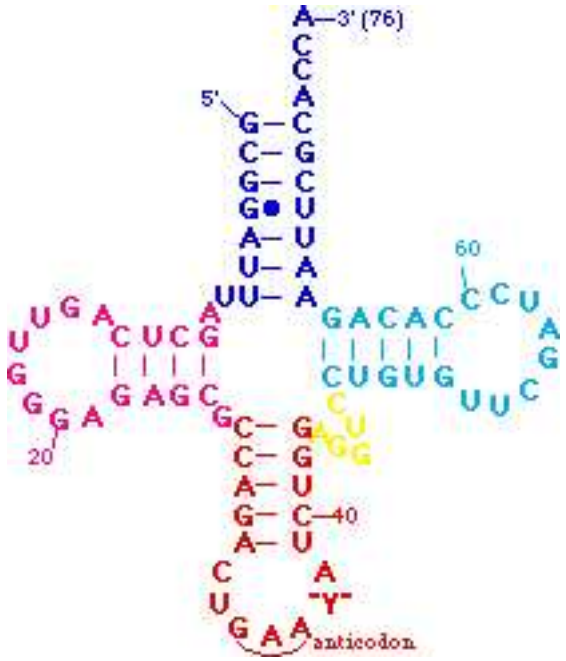
Algorithmes pour la comparaison d'ARN : distance d'édition entre arbres

Serge Dulucq (LABRI)

Hélène Touzet (LIFL)





□ : 2 bases appariées (tige)

● : base libre (boucle)

Représentation arborescente pour la structure secondaire de l'ARN de transfert

Opérations d'édition

- substitution d'un nœud
- insertion d'un nœud
- délétion d'un nœud

Distance d'édition sur les mots

A L G O R I T H M E

P A P I L L O N

▷ Décomposition en partant de la gauche

substitution		*insertion*		*délétion*	
A	LGORITHME	-	ALGORITHME	A	LGORITHME
P	APILLON	P	APILLON	-	PAPILLON

Appel sur tous les suffixes

▷ Décomposition en partant de la droite

- *substitution*

ALGORITHM	E
PAPILLO	N

- *insertion*

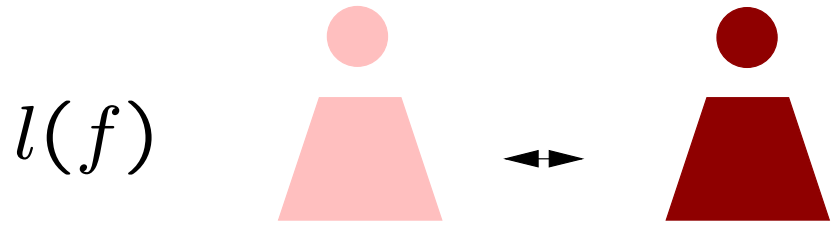
ALGORITHME	-
PAPILLO	N

- *délétion*

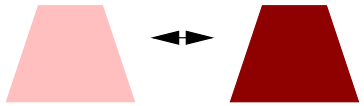
ALGORITHM	E
PAPILLON	-

Appel sur tous les préfixes

Distance d'édition sur les arbres

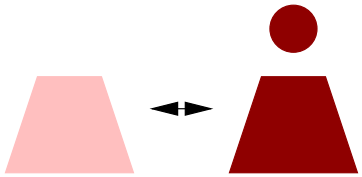


Substitution



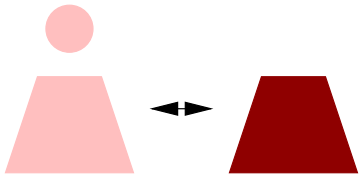
$\texttt{Distance}(f, f')$
+
$\texttt{sub}(l, l')$

Délétion



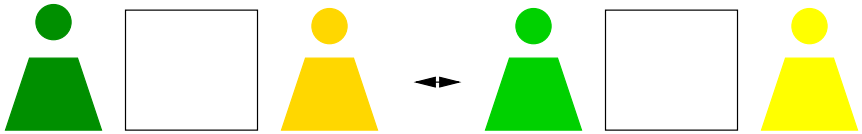
$\texttt{Distance}(f, l'(f'))$
+
$\texttt{del}(l)$

Insertion



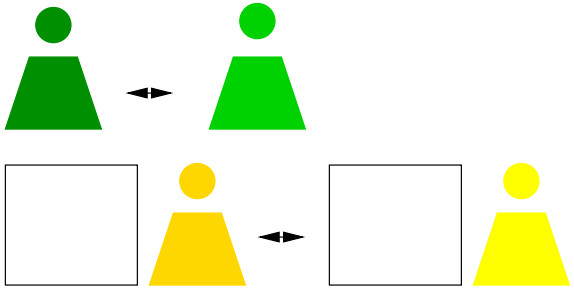
$\texttt{Distance}(l(f), f')$
+
$\texttt{ins}(l')$

Distance d'édition sur les forêts I

$l(f) \circ t$  $l'(f') \circ t'$

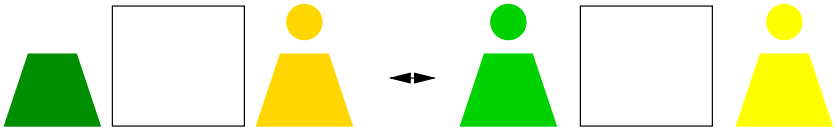
Décomposition gauche

Substitution



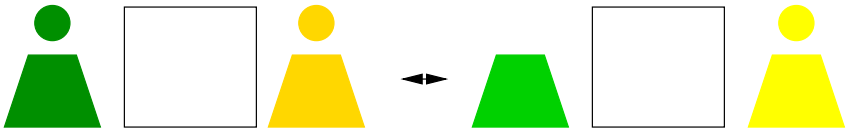
$\texttt{Distance}(l(f), l'(f'))$
$+$
$\texttt{Distance}(t, t')$

Délétion



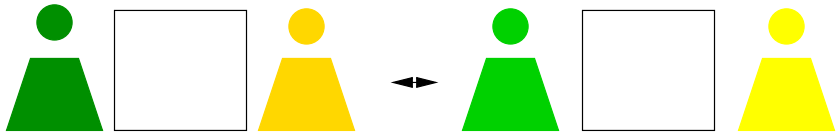
$\texttt{Distance}(f \circ t, l'(f') \circ t') + \texttt{del}(l)$

Insertion



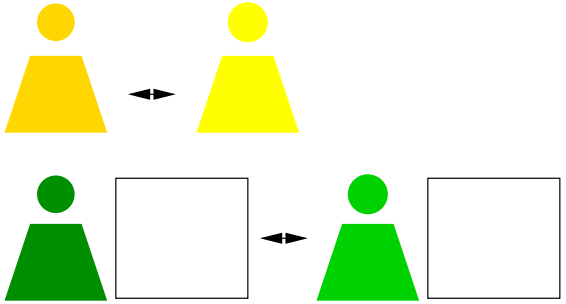
$\texttt{Distance}(l(f) \circ t, f' \circ t') + \texttt{ins}(l')$

Distance d'édition sur les forêts II

$t \circ l(f)$  $t' \circ l'(f')$

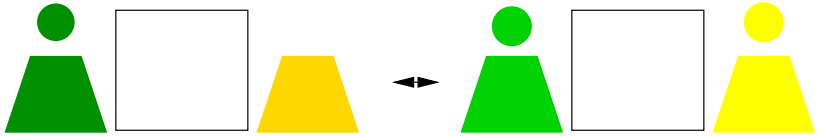
Décomposition droite

Substitution



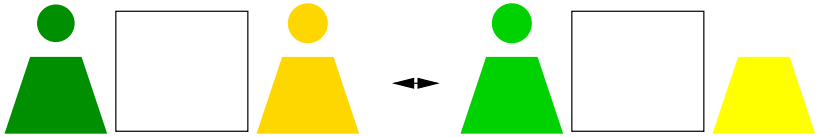
$$\texttt{Distance}(l(f), l'(f')) + \texttt{Distance}(t, t')$$

Délétion



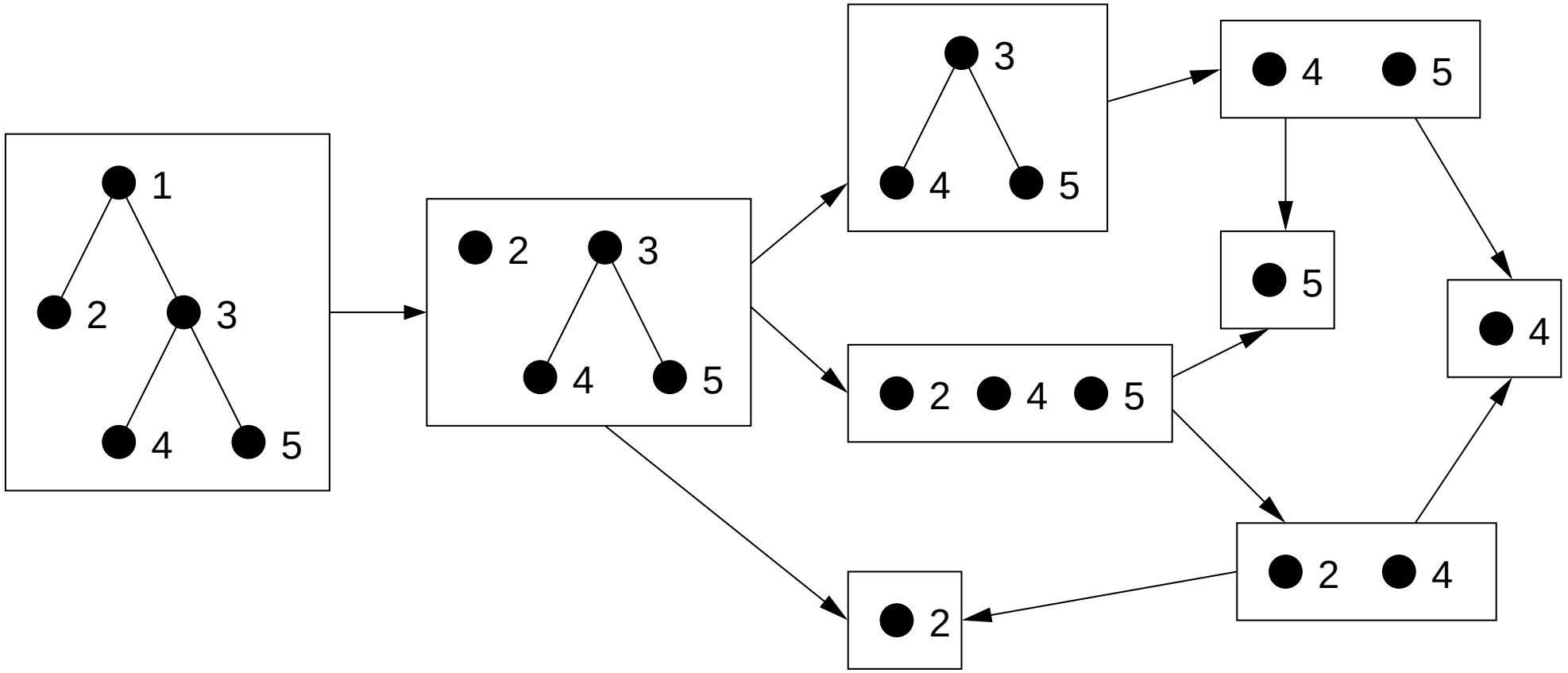
$$\texttt{Distance}(t \circ f, l'(f') \circ l'(f')) + \texttt{del}(l)$$

Insertion



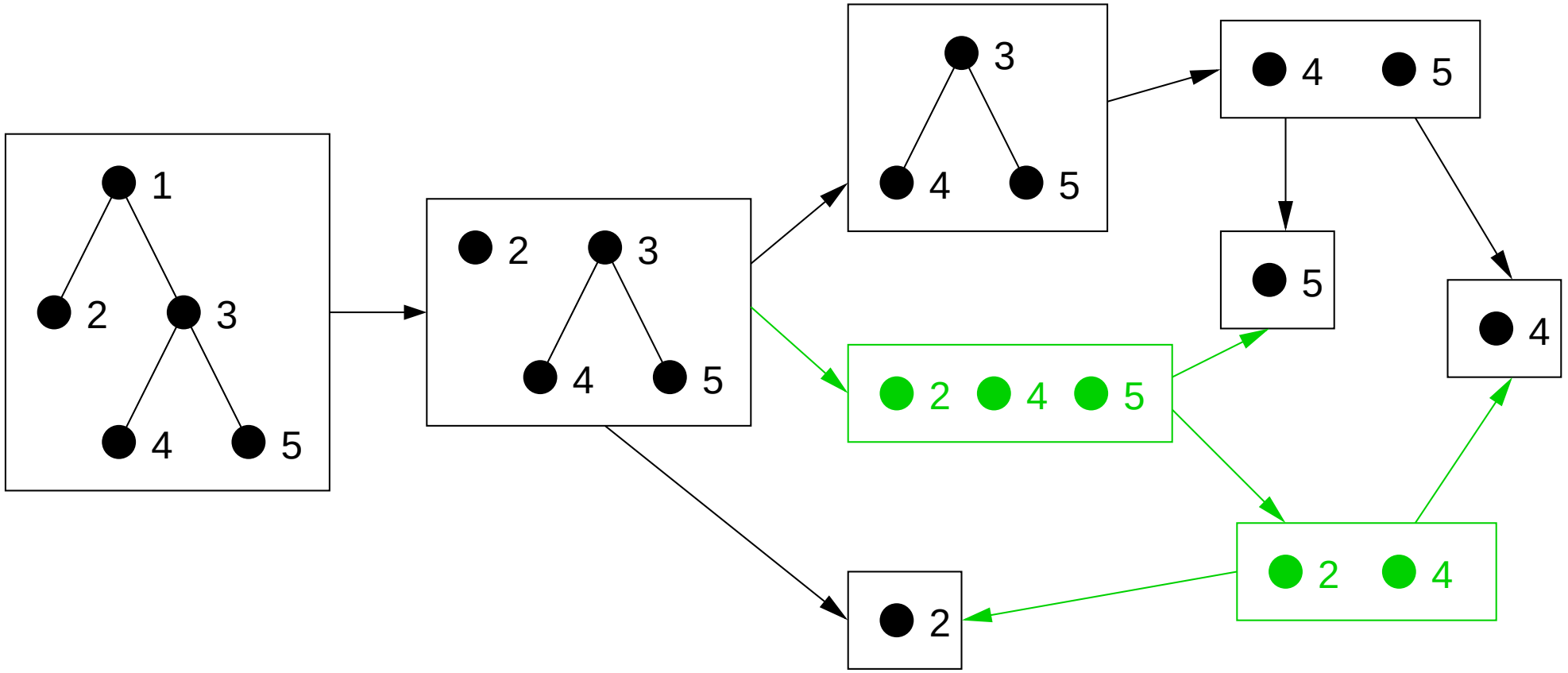
$$\texttt{Distance}(t \circ l(f), t' \circ t') + \texttt{ins}(l')$$

Droite/gauche pour les arbres ?



décomposition droite

Droite/gauche pour les arbres ?



décomposition gauche

Peigne à n branches ($2n+1$ nœuds)

▷ Décomposition gauche :

$$3n + 1 \text{ sous-forêts}$$

▷ Décomposition droite :

$$2n^2 + 1 \text{ sous-forêts}$$

Stratégie de décomposition

- succession de choix **droite** ou **gauche**
- $\mathcal{S}$: *forêt* $\times$ *forêt* $\rightarrow$ {droite, gauche}
- **Zhang & Shasha** (1989) :

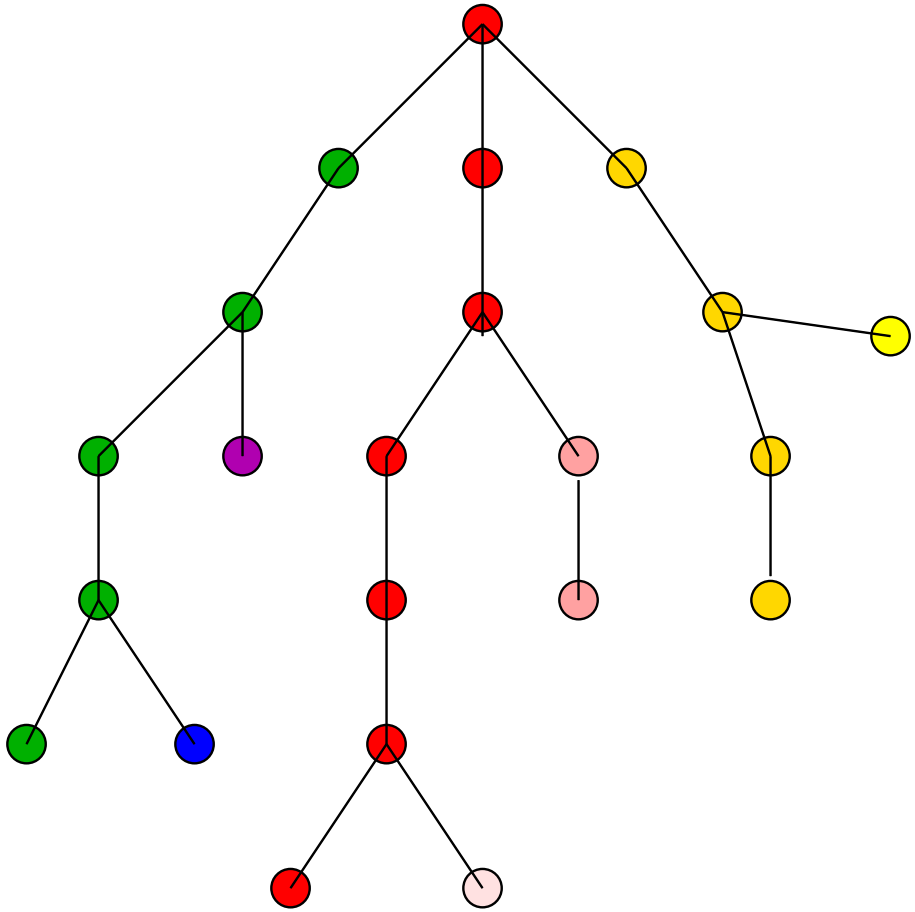
$$(f, g) \rightarrow \text{gauche}$$

- **Klein** (1998) :

$(f, g) \rightarrow$ **gauche** si le premier nœud de f est sur le chemin lourd

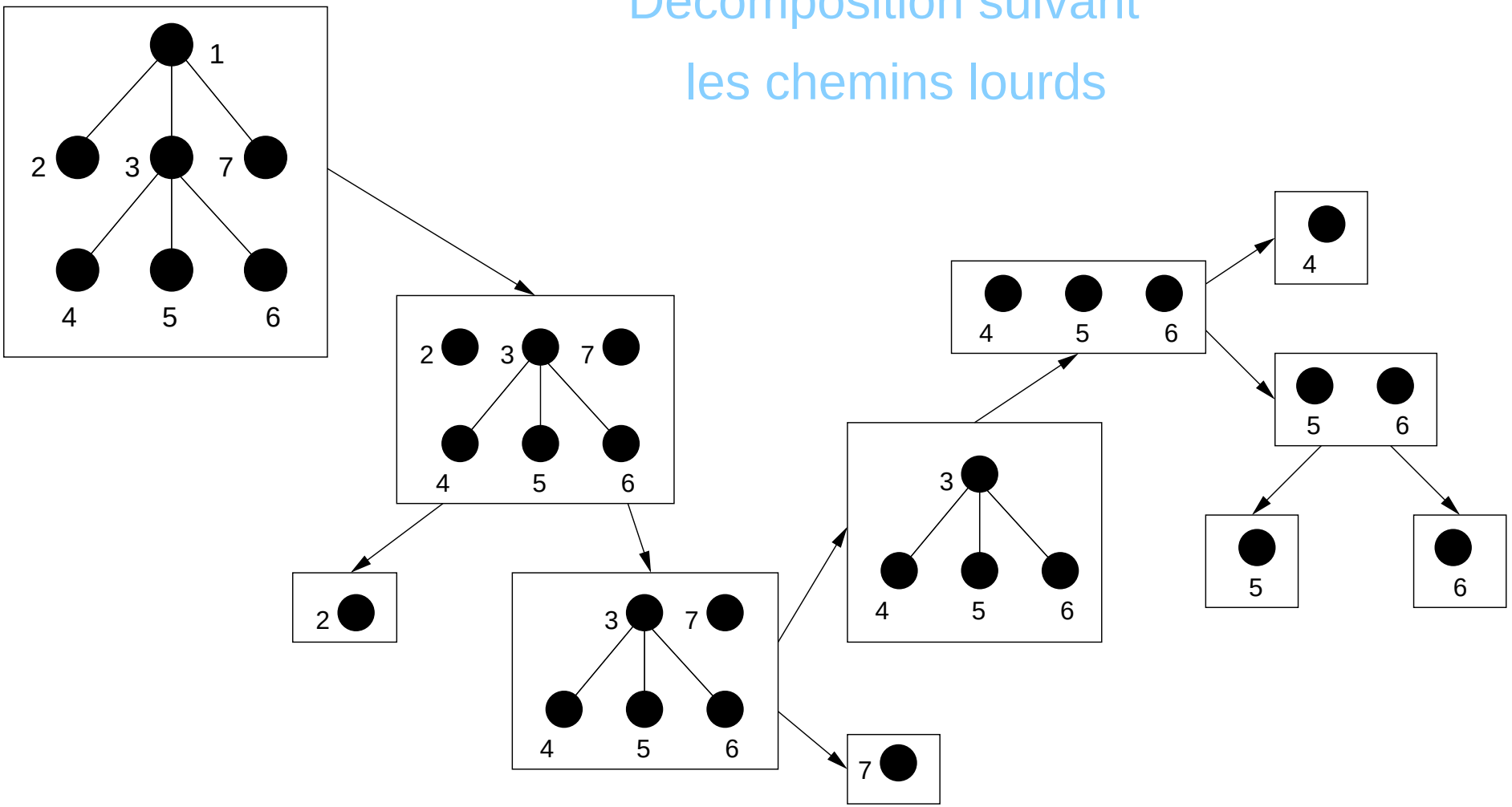
droite, sinon

Chemins lourds

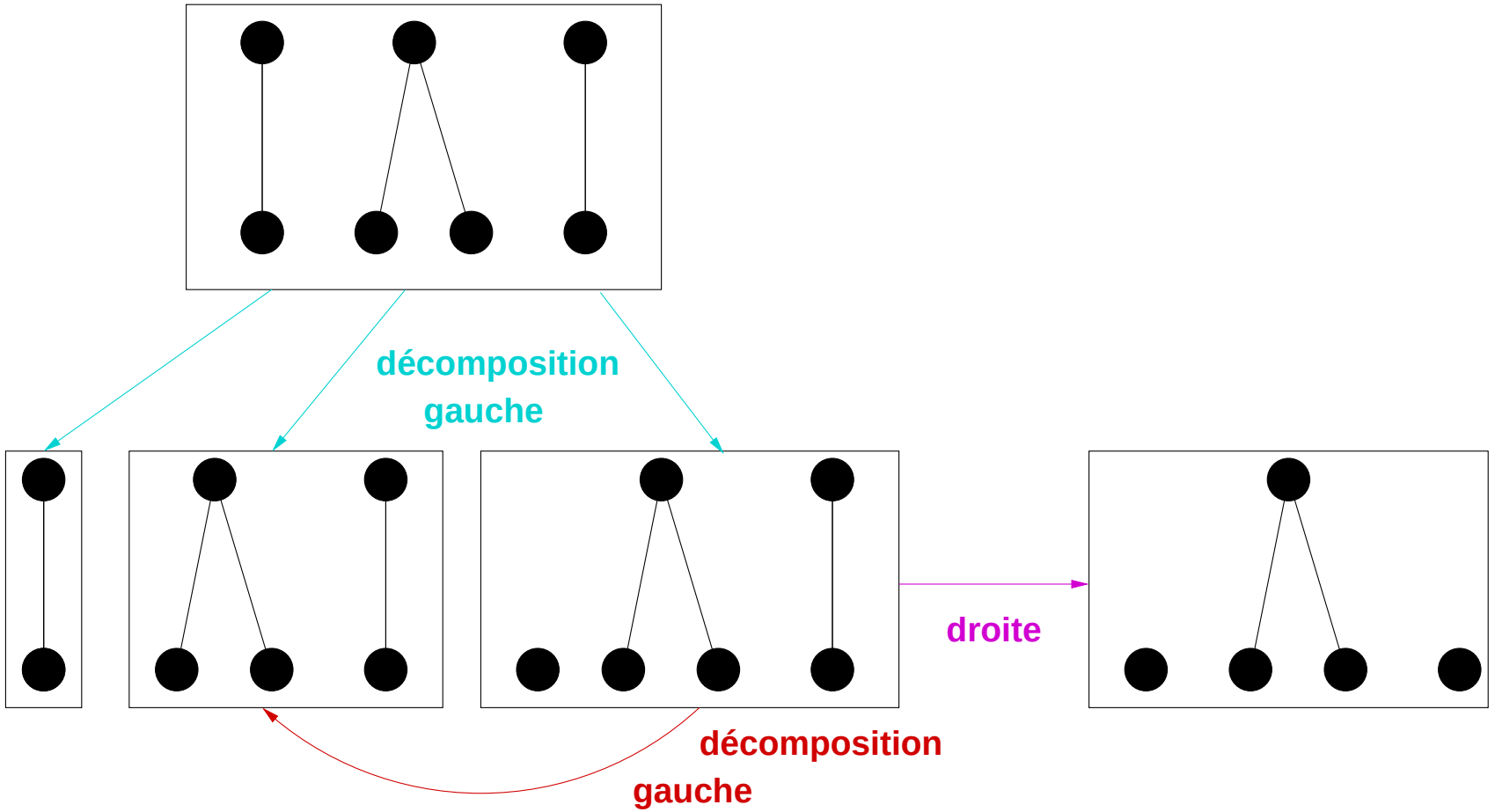


Stratégie de Klein

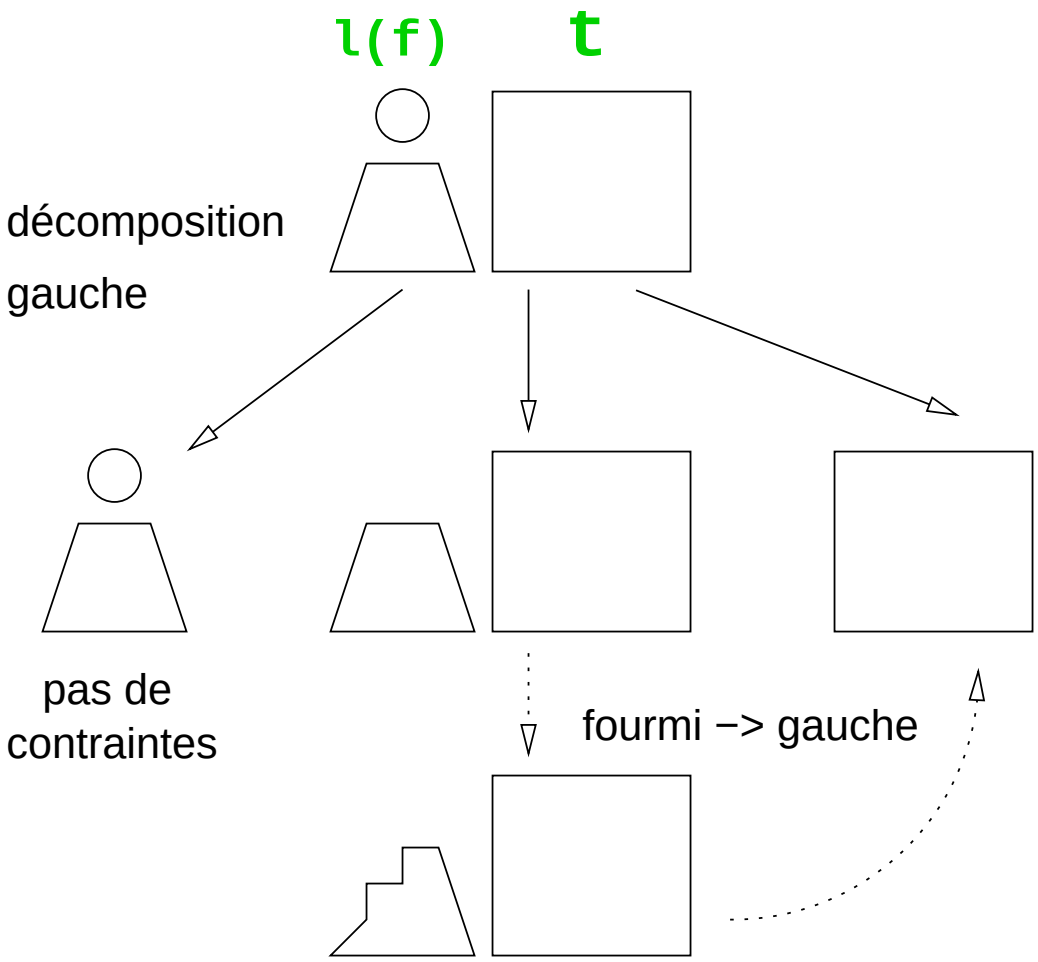
Décomposition suivant les chemins lourds



Exemple de stratégie dispendieuse



Les stratégies fourmis

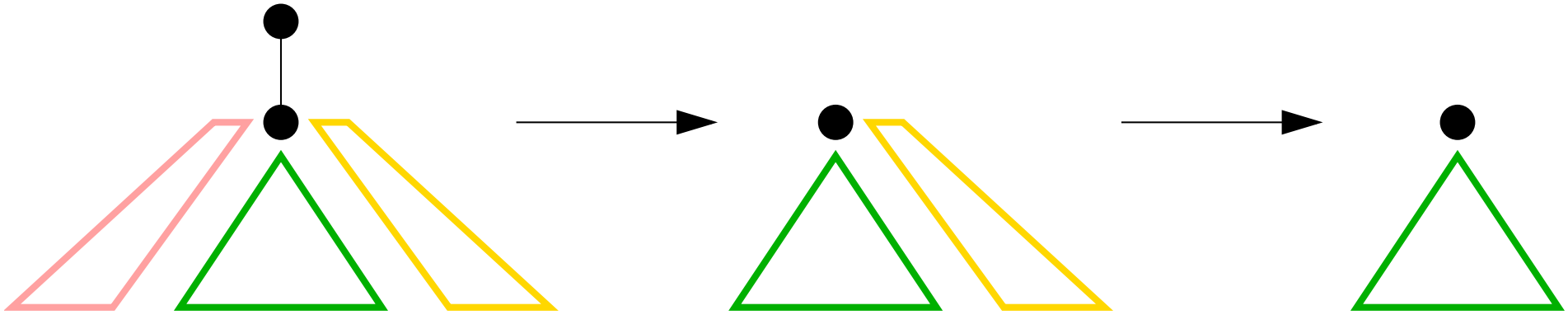


#sous-forets(l(f)t)=

#sous-forets(l(f))+|l(f)|+#sous-forets(t)

Définir une stratégie fourmi à partir d'un recouvrement

- $\phi(i) \in \{droite, gauche\}$ si i est un noeud d'arité 0 ou 1
- $\phi(i)$ est un fils de i sinon : le fils **favori**



décompositions gauches décompositions droites

Recouvrement pour Zhang et Klein

- **Zhang :**
  - le fils favoris est le fils droit
  - la direction est gauche

  Zhang est fourmi pour les deux arbres
- **Klein :**
  - le fils favori est le fils de taille maximale
  - la direction est gauche

  Klein est fourmi pour le premier arbre

Nombre de sous-forêts sur un arbre

$A = l(A_1 \circ \ldots \circ A_n)$

▷ **pour une stratégie quelconque**

meilleur des cas

$$|A| - |A_j| + \texttt{\#sous-forêts}(A_1) + \cdots + \texttt{\#sous-forêts}(A_n) \qquad O(n \log(n))$$

A_j est le fils de taille maximale

pire des cas

$$\frac{n(n+3)}{2} - \sum_{i \in A} |A(i)|$$

$$\frac{1}{2}\, n^2 + \frac{\sqrt{\pi}}{2}\, n^{\frac{3}{2}} + O(n) \text{ en moyenne}$$

▷ **pour un arbre muni d'un recouvrement**

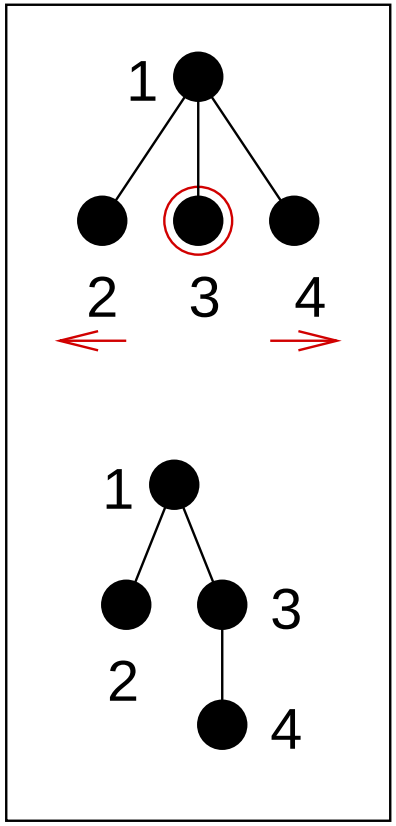
$$\texttt{\#sous-forêts}(A) = |A| - |A_j| + \texttt{\#sous-forêts}(A_1) + \cdots + \texttt{\#sous-forêts}(A_n)$$

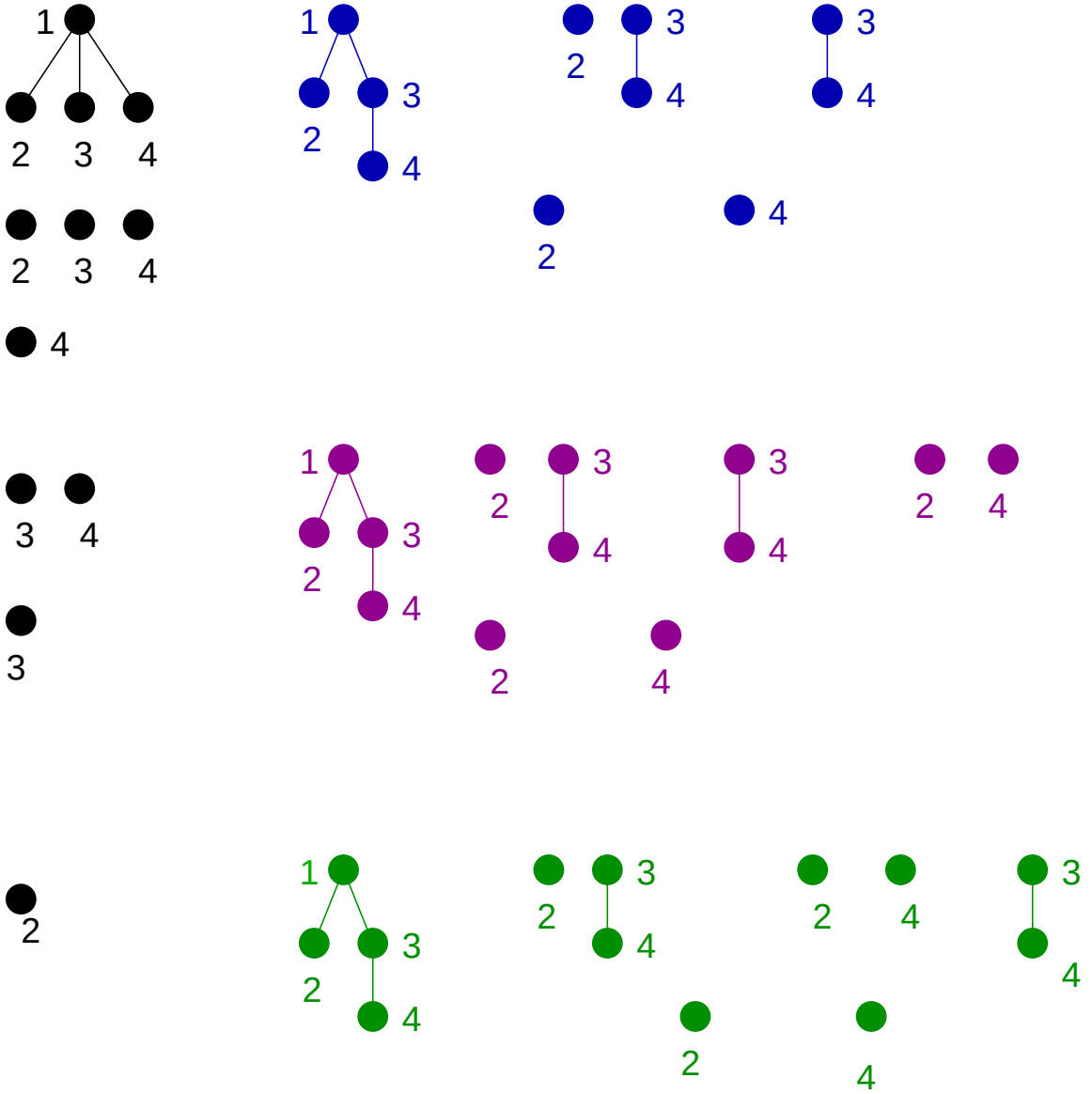
A_j est le fils favori

Conséquence 1: nombre exact de sous-forêts pour Zhang (forêts droites)

Conséquence 2: Klein est optimal pour l'arbre muni du recouvrement

Que se passe-t-il pour l'autre arbre ?

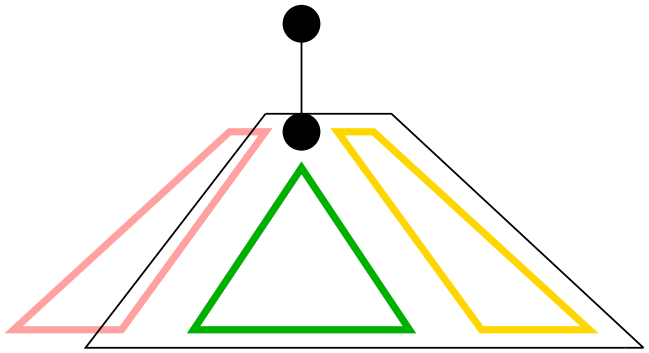




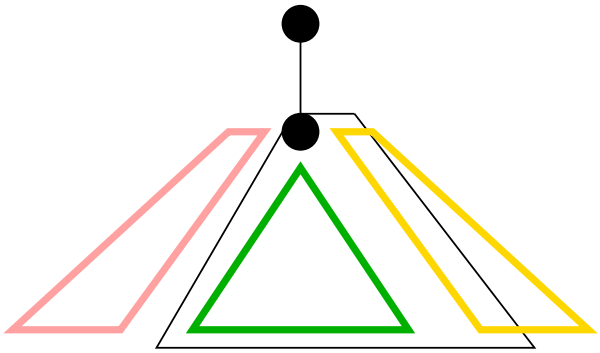
Que se passe-t-il pour l'autre arbre ?

Il a exactement trois types de forêts pour l'arbre directeur

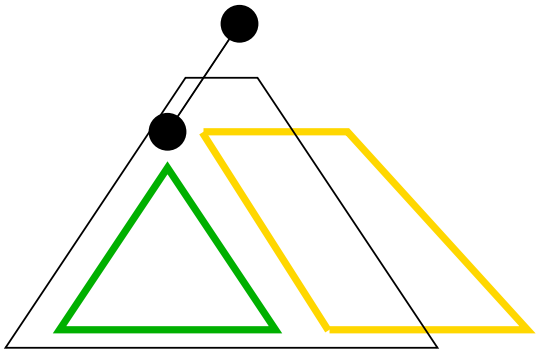
- ▷ celles qui sont comparées avec toutes les forêts gauches de B
- ▷ celles qui sont comparées avec toutes les forêts droites de B
- ▷ celles qui sont comparées avec toutes les forêts de B



toutes les forets droites



toutes les forets



toutes les forets gauches

Les fils favoris et les fils uniques héritent des forêts de leur parent.

- **Free:** les nœuds qui n'héritent de rien
- **Left :** les nœuds qui héritent des forêts gauches
- **Right :** les nœuds qui héritent des forêts droites
- **All:** les nœuds qui héritent de toutes les forêts

1. A est un seul nœud de direction droite

$$\begin{array}{rcrcl} \mathsf{Free}(A) & = & \mathsf{Left}(A) & = & \texttt{\#left}(B) \\ \mathsf{All}(A) & = & \mathsf{Right}(A) & = & \texttt{\#special}(B) \end{array}$$

2. A est un seul nœud de direction gauche

$$\begin{array}{rcrcl} \mathsf{Free}(A) & = & \mathsf{Right}(A) & = & \texttt{\#right}(B) \\ \mathsf{All}(A) & = & \mathsf{Left}(A) & = & \texttt{\#special}(B) \end{array}$$

3. $A = l(A')$ et la direction de l est droite

$$\begin{array}{rcrcl} \mathsf{Free}(A) & = & \mathsf{Left}(A) & = & \texttt{\#left}(B) + \mathsf{Right}(A') \\ \mathsf{All}(A) & = & \mathsf{Right}(A) & = & \texttt{\#special}(B) + \mathsf{All}(A') \end{array}$$

4. $A = l(A')$ et la direction de l est gauche

$$\begin{array}{rcrcl} \mathsf{Free}(A) & = & \mathsf{Right}(A) & = & \texttt{\#right}(B) + \mathsf{Left}(A') \\ \mathsf{All}(A) & = & \mathsf{Left}(A) & = & \texttt{\#special}(B) + \mathsf{All}(A') \end{array}$$

5. $A = l(A_1 \circ \cdots \circ A_n)$ et le fils favori est A_1

$$\begin{array}{rcl} \mathsf{Free}(A) & = & \mathsf{Left}(A) = \sum_{i>1} \mathsf{Free}(A_i) + \mathsf{Left}(A_1) + \texttt{\#left}(B)(|A| - |A_1|) \\ \mathsf{All}(A) & = & \mathsf{Right}(A) = \sum_{i>1} \mathsf{Free}(A_i) + \mathsf{All}(A_1) + \texttt{\#special}(B)(|A| - |A_1| \end{array}$$

6. $A = l(A_1 \circ \cdots \circ A_n)$ et le fils favori est A_1

$$\begin{array}{rcl} \mathsf{Free}(A) & = & \mathsf{Right}(A) = \sum_{i<n} \mathsf{Free}(A_i) + \mathsf{Right}(A_n) + \texttt{\#right}(B)(|A| - |A_n \\ \mathsf{All}(A) & = & \mathsf{Left}(A) = \sum_{i<n} \mathsf{Free}(A_i) + \mathsf{All}(A_n) + \texttt{\#special}(B)(|A| - |A_n| \end{array}$$

7. sinon, le fils favori est A_j avec $1 < j < n$

$$\begin{array}{rl} \mathsf{Free}(A) = & \sum_{i \neq j} \mathsf{Free}(A_i) + \mathsf{All}(A_j) + \texttt{\#right}(B)(1 + |A_1 \circ \cdots \circ A_{j-1}|) \\ & + \texttt{\#special}(B)|A_j \circ \cdots \circ A_n| \\ \mathsf{Right}(A) = & \mathsf{Free}(A) \\ \mathsf{All}(A) = & \mathsf{Left}(A) = \sum_{i \neq j} \mathsf{Free}(A_i) + \mathsf{All}(A_j) + \texttt{\#special}(B)(|A| - |A_j|) \end{array}$$

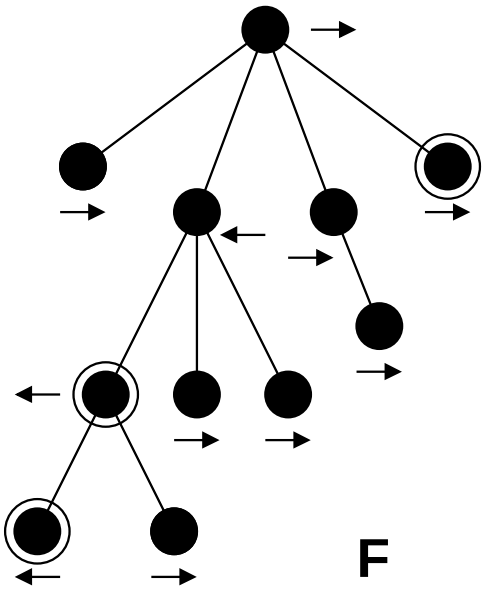
Comment en déduire un recouvrement optimal

- programmation dynamique
- 4 tables : *Free*, *All*, *Left*, *Right*

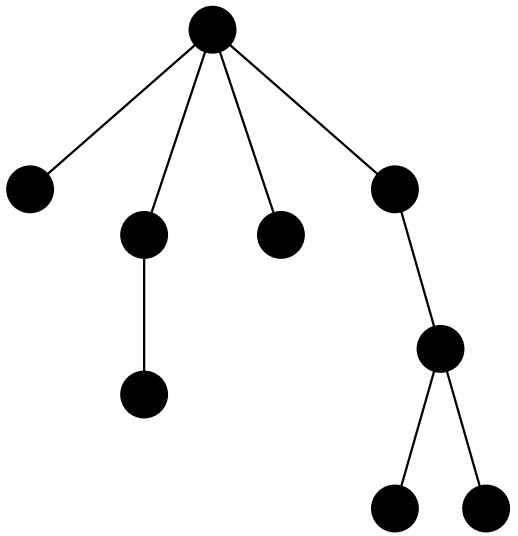
$$\begin{aligned} \mathit{Free}(A) \;=\; & \textstyle\sum_{i\geq 1} \mathit{Free}(A_i) \\ & + \min \begin{cases} \mathit{Left}(A_1) - \mathit{Free}(A_1) + \texttt{\#left}(B) * (|A| - |A_1|) \\ \mathit{All}(A_j) - \mathit{Free}(A_j) + \texttt{\#special}(B)|A_j \circ \cdots \circ A_n| \\ + \texttt{\#right}(B)(1 + |A_1 \circ \cdots \circ A_{j-1}|), \quad 1 < j < n \\ \mathit{Right}(A_n) - \mathit{Free}(A_n) + \texttt{\#right}(B) * (|A| - |A_n|) \end{cases} \end{aligned}$$

- Coût du prétraitement : $O(\sum_i \text{arité(A(i))}) + O(|B|) \;=\; O(|A|) + {} + O(|B|)$

optimal	:	340
droite	:	405
gauche	:	350
Klein	:	391

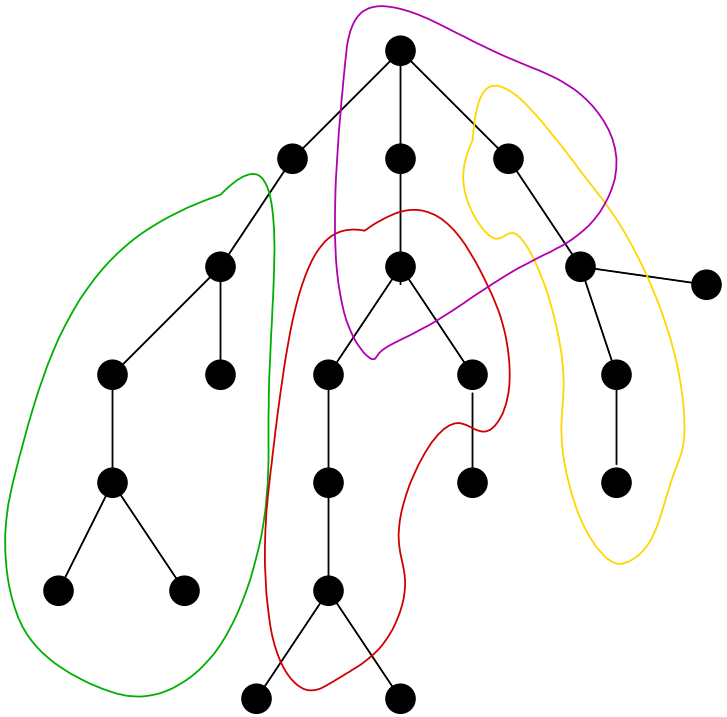


F



G

Distance d'édition avec gaps



Distance: NP-dur

Indel sur les sous-structures terminales :
$O(n^2)$