

# Modélisation algébrique de la structure musicale

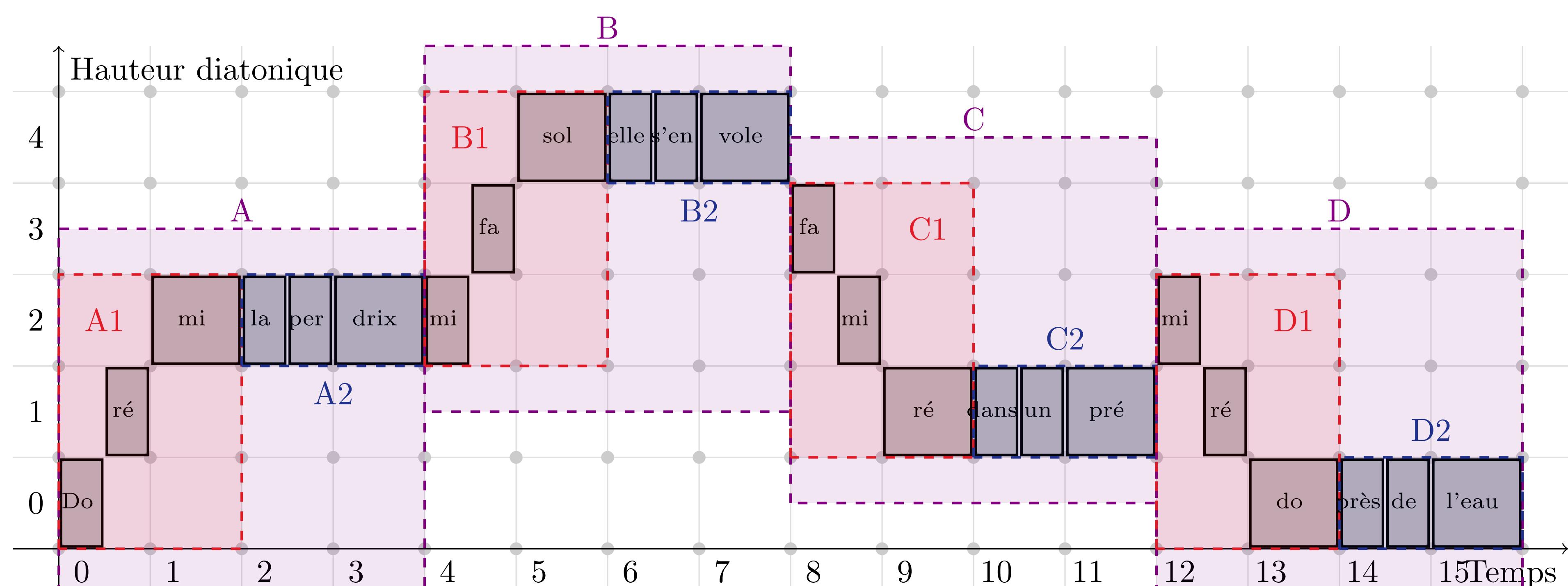
Alice Rixte - David Janin, Martin Laliberté

LX - Univ. Bordeaux, LaBRI, France

**LaBRI**

## Comment encoder la structure musicale de la comptine *Do ré mi, la perdrix* ?

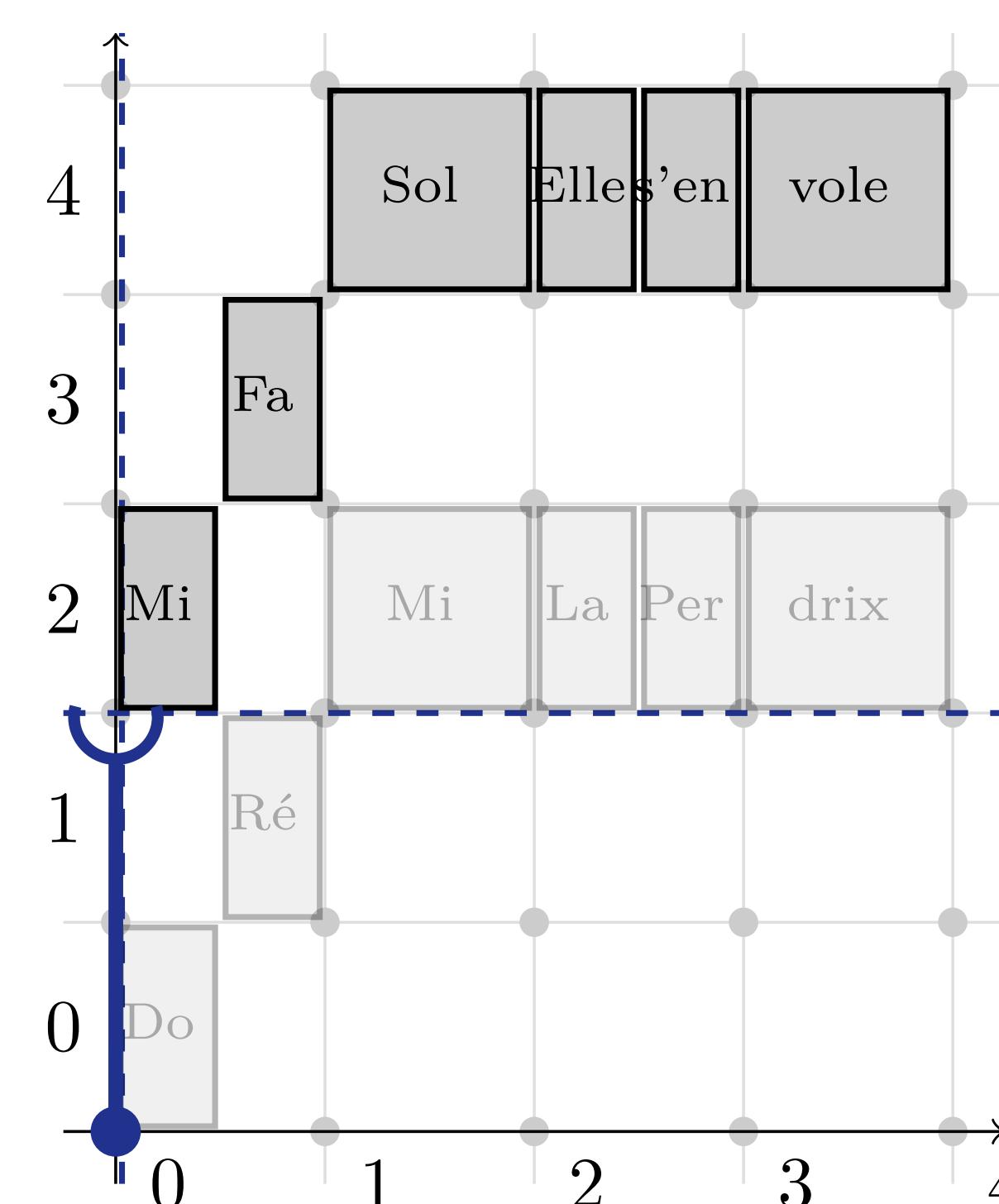
- ▶ La comptine *Do ré mi, la perdrix* peut être divisée en quatre motifs similaires *A, B, C et D*.
- ▶ Chacun de ces motifs est lui-même composé de deux sous-motifs partageant le même rythme.



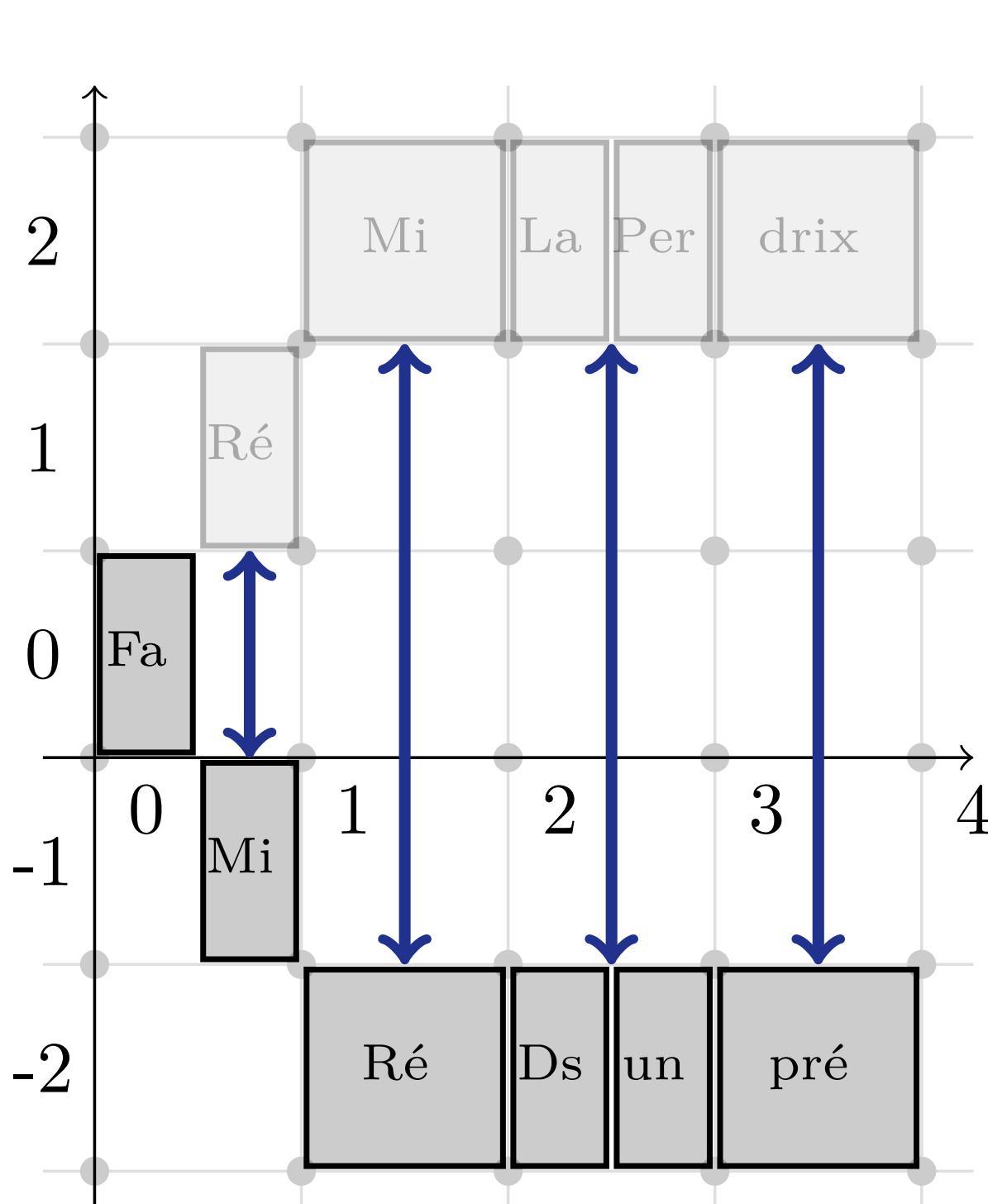
La similarité entre motifs s'exprime en transformant l'espace :

- ▶ Le motif *B* est une transposition diatonique du motif *A*.
- ▶ Le motif *C* est une inversion (symétrie verticale) du motif *A*.
- ▶ Le sous-motif *A2* est une projection du motif *A1*.

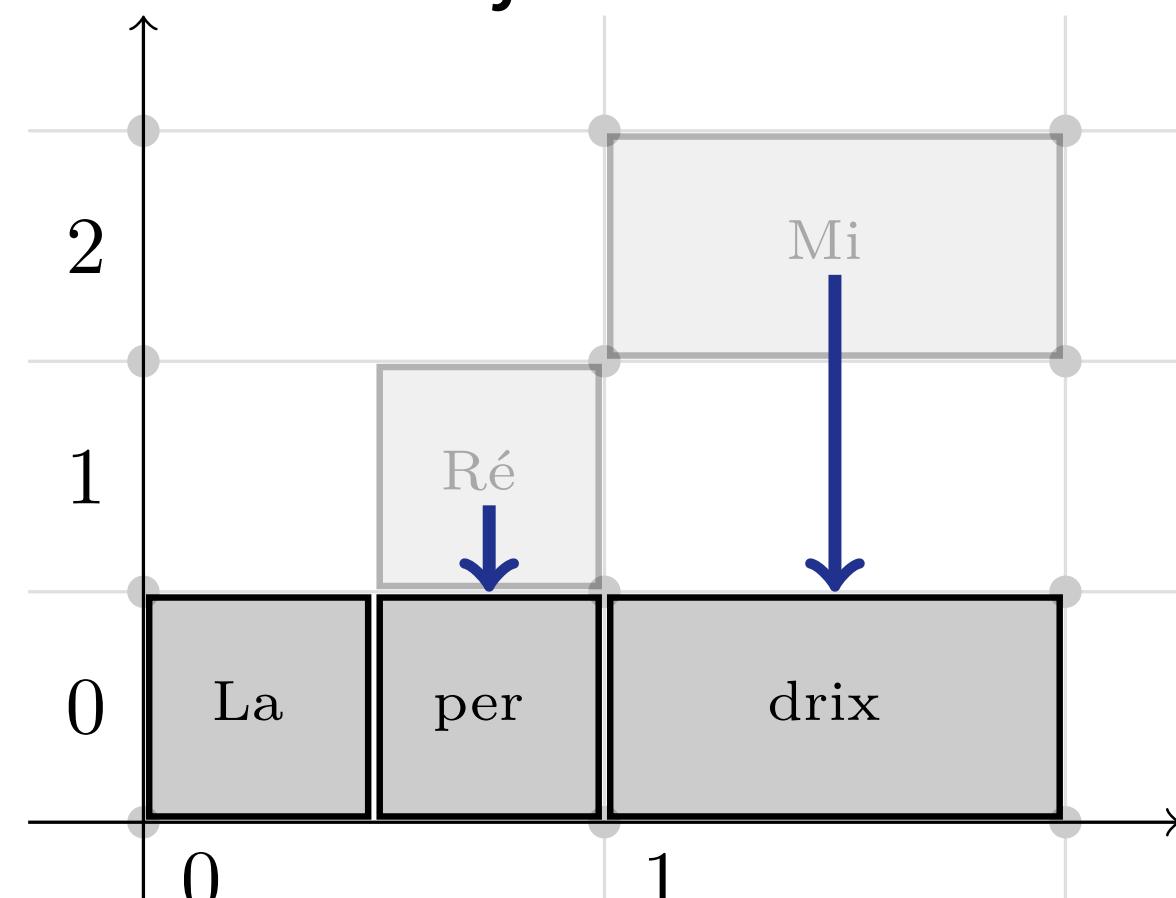
#### Transposition diatonique :



#### Inversion :

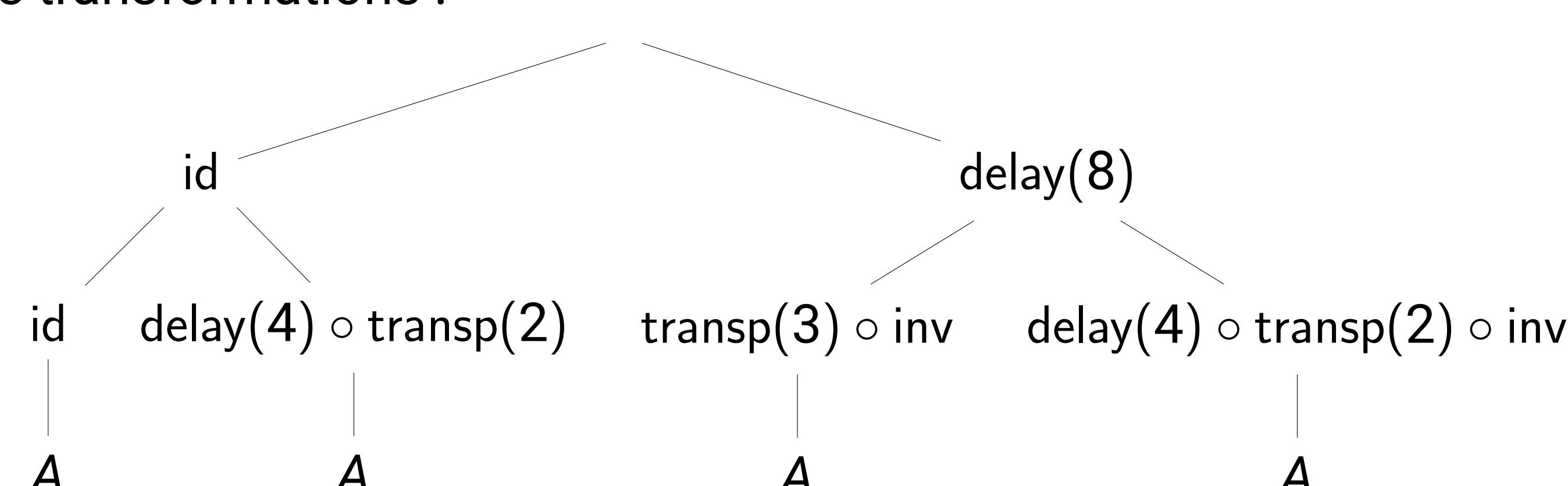


#### Projection :

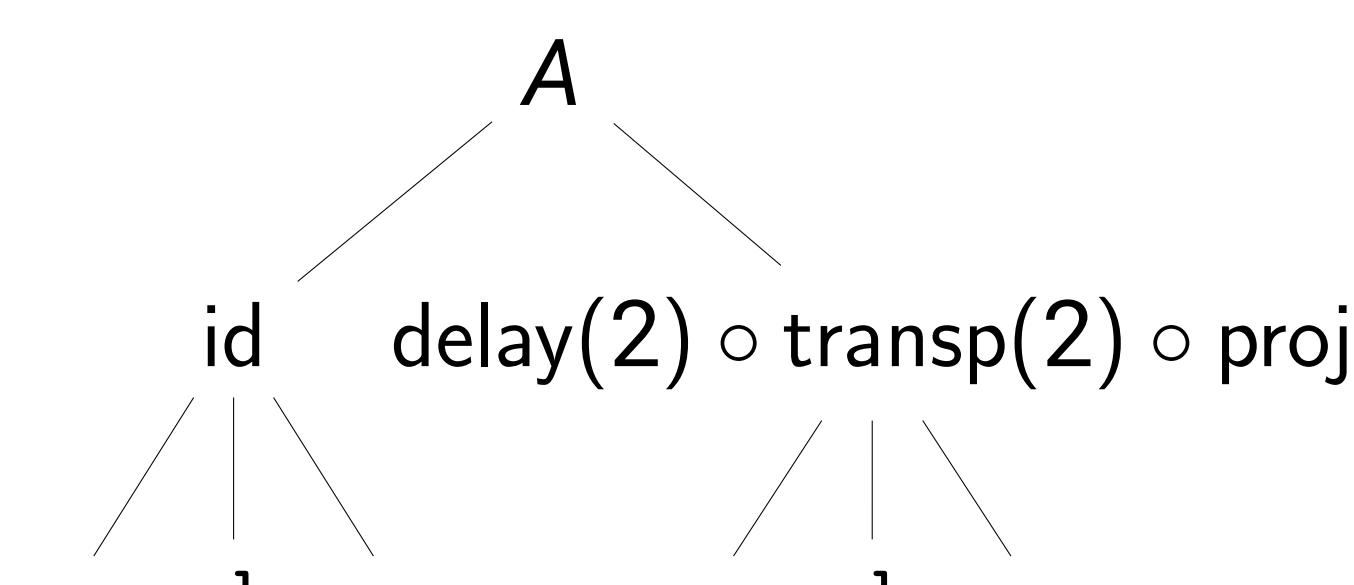


### Hierarchie de transformations

La comptine *Do ré mi, la perdrix* peut être encodée par une hiérarchie de transformations :



où *A* est le sous-arbre :



### Un langage de programmation

Les hiérarchies de transformations peuvent être exprimées à l'aide d'un langage de programmation déclaratif.

- ▶ On déclare une note avec l'instruction `atom (t, h)` où *t* la position temporelle de la note, *h* la hauteur de la note et *d* sa durée.
- ▶ L'instruction `change c` change le repère courant en appliquant la fonction *c*.

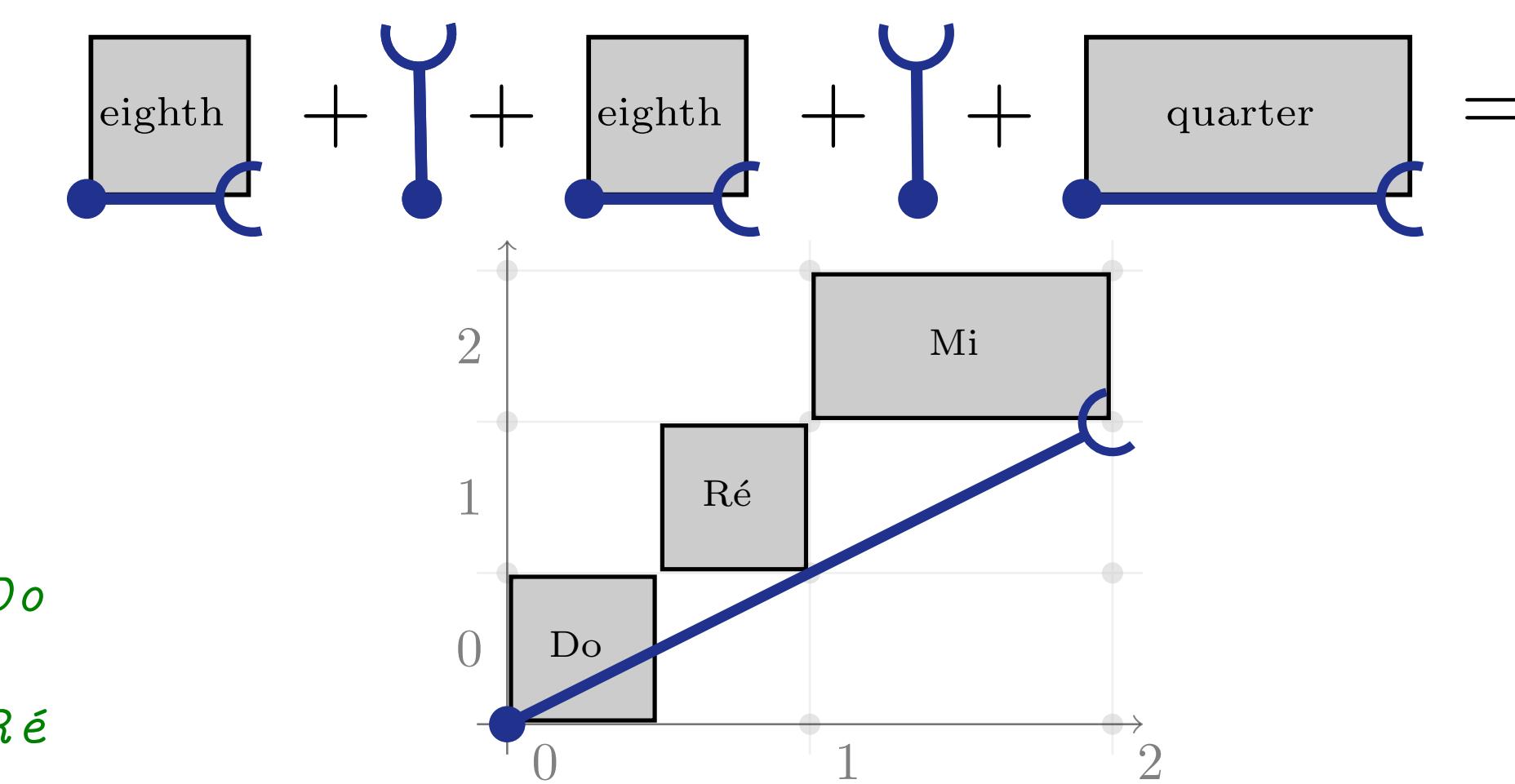
```

eighth = do
  atom (0,0) (1/2);
  change (delay (1/2))

quarter = do
  atom (0,0) 1;
  change (delay 1)

tileA1 = do
  eighth;
  change (transp 1); --Do
  eighth;
  change (transp 1); --Ré
  quarter;
  change (transp 1); --Mi

```



### Encoder la hiérarchie

- ▶ Pour encoder la hiérarchie de transformations, on utilise l'instruction `t |-> p` qui applique la transformation *t* au programme *p*.
- ▶ Une fois tous les changements de repère appliqués, le programme suivant exprime la structure de *Do ré mi, la perdrix* :

```

tileA = do
  idle |-> tileA1; --A1
  proj |-> tileA1; --A2

drmFull = do
  idle |-> tileA; -- A
  idle |-> tileA; -- B
  change (transp (-1));
  inv |-> tileA; -- C
  change (transp 1);
  inv |-> tileA; -- D

```

### Perpectives

**Musique électronique live** Avoir accès à la structure du morceau permet d'appliquer en live des transformations harmoniques, rythmiques ou encore mélodiques à la partition.

**Analyse musicale** Le modèle présenté ici donne une sémantique formalisant ce qu'on entend par structure musicale, ouvrant ainsi des perspectives d'analyse en musicologie computationnelle.