Recherche de représentant, point de vue symbolique sur les algorithmes réversibles

Ali Akhavi

GREYC, Université de Caen

Exposé au LIPN - Université de Sorbonne Paris Nord

4 mai 2021

Plan de l'exposé :

- Problème de la recherche d'un représentant
  - Des exemples
  - Enoncé général
- Algorithme réversible pour la recherche d'un représentant.
- Quelques applications :
  - Borne inférieure de complexité pour algorithmes, pour le problème
  - Caractérisation du langage des traces d'un algorithme

Exemples de problèmes de recherche d'un représentant

1. D un ensemble, muni d'un ordre total.
   Entrée : une séquence de n éléments de D.
   Sortie : Le représentant ordonné dans l'ordre croissant, parmi toutes les séquences qui contiennent les mêmes éléments que l'entrée.
2. Tri topologique.
   Entrée : un graphe orienté acyclique $G(V, E)$, avec $V = \{1, \ldots, n\}$ et une énumération des ses sommets.
   Sortie : une énumération des sommets de G qui respecte l'ordre partiel induit par le graphe G.
3. D le groupe libre Abélien $\mathbb{R}^p$, muni de la structure euclidenne.
   Entrée : une séquence de n éléments de D.
   Sortie : Un représentant parmi toutes les séquences de n éléments qui engendrent le même groupe Abélien libre que la séquence d'entrée et dont les vecteurs sont les plus courts possibles.
4. $D = \{0, 1\}^*$ sur lequel opère un groupe monogène, donnée par la description d'une machine M_f qui calcule l'action d'un générateur f sur un mot.
   Entrée : un élément $x \in D$ et M_f.
   Sortie : un représentant $x^\star$ de l'orbite de x sous l'action de f, dont le poids de Hamming est le plus grand.

Les exemples 1 et 3 précédents serviront de fil conducteur dans tout l'exposé.

La réduction des réseaux euclidiens

Un réseau euclidien de $\mathbb{R}^p$ est l'ensemble des combinaisons linéaires entières d'une famille libre $(b_1, b_2, \ldots, b_n)$ de $\mathbb{R}^p$ avec $n \leq p$:

$$\mathcal{L} := \{x \in \mathbb{R}^p; \quad x = \sum_{i=1}^{n} x_i b_i, \qquad x_i \in \mathbb{Z}\}$$

$(b_1, b_2, \ldots, b_n)$ est une base du réseau. Un réseau possède une infinité de bases (tous de même cardinal n, dimension de $\mathcal{L}$)

L'espace ambiant $\mathbb{R}^p$ étant muni de sa structure euclidienne canonique , certaines bases ont de bonnes propriétés euclidiennes (elles sont dites réduites) : les vecteurs sont assez courts et la base est assez orthogonale.

La réduction des réseaux euclidiens

Un réseau euclidien de $\mathbb{R}^p$ est l'ensemble des combinaisons linéaires entières d'une famille libre $(b_1, b_2, \ldots, b_n)$ de $\mathbb{R}^p$ avec $n \leq p$:

$$\mathcal{L} := \{x \in \mathbb{R}^p; \quad x = \sum_{i=1}^{n} x_i b_i, \qquad x_i \in \mathbb{Z}\}$$

$(b_1, b_2, \ldots, b_n)$ est une base du réseau. Un réseau possède une infinité de bases (tous de même cardinal n, dimension de $\mathcal{L}$)

L'espace ambiant $\mathbb{R}^p$ étant muni de sa structure euclidienne canonique , certaines bases ont de bonnes propriétés euclidiennes (elles sont dites réduites) : les vecteurs sont assez courts et la base est assez orthogonale.

Réduction d'un réseau $\mathcal{L}$:
A partir par d'une base quelconque de $\mathcal{L}$, construire une base réduite de $\mathcal{L}$.

La réduction des réseaux euclidiens

Un réseau euclidien de $\mathbb{R}^p$ est l'ensemble des combinaisons linéaires entières d'une famille libre $(b_1, b_2, \ldots, b_n)$ de $\mathbb{R}^p$ avec $n \leq p$:

$$\mathcal{L} := \{x \in \mathbb{R}^p; \quad x = \sum_{i=1}^{n} x_i b_i, \qquad x_i \in \mathbb{Z}\}$$

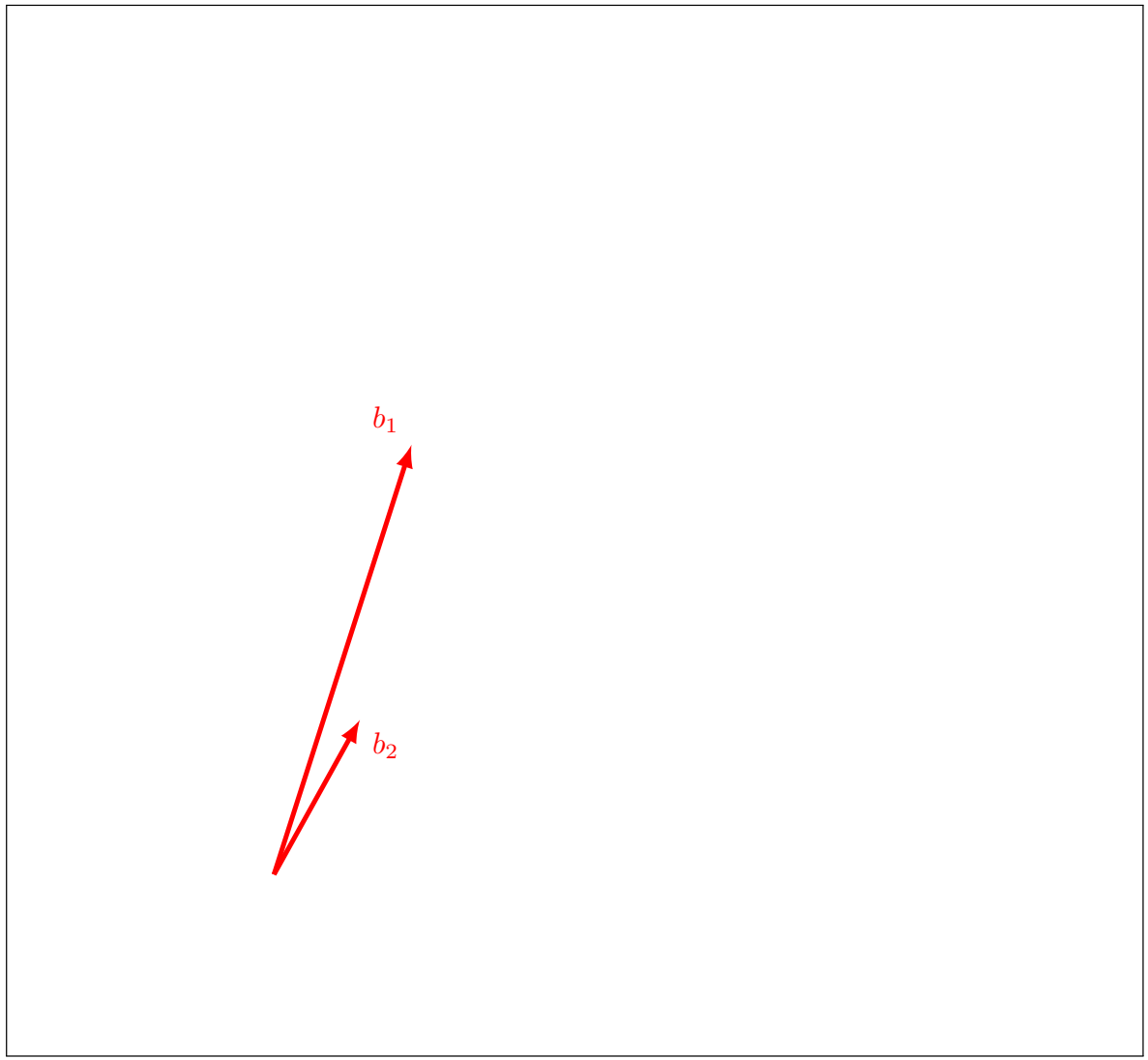
$(b_1, b_2, \ldots, b_n)$ est une base du réseau. Un réseau possède une infinité de bases (tous de même cardinal n, dimension de $\mathcal{L}$)

L'espace ambiant $\mathbb{R}^p$ étant muni de sa structure euclidienne canonique , certaines bases ont de bonnes propriétés euclidiennes (elles sont dites réduites) : les vecteurs sont assez courts et la base est assez orthogonale.

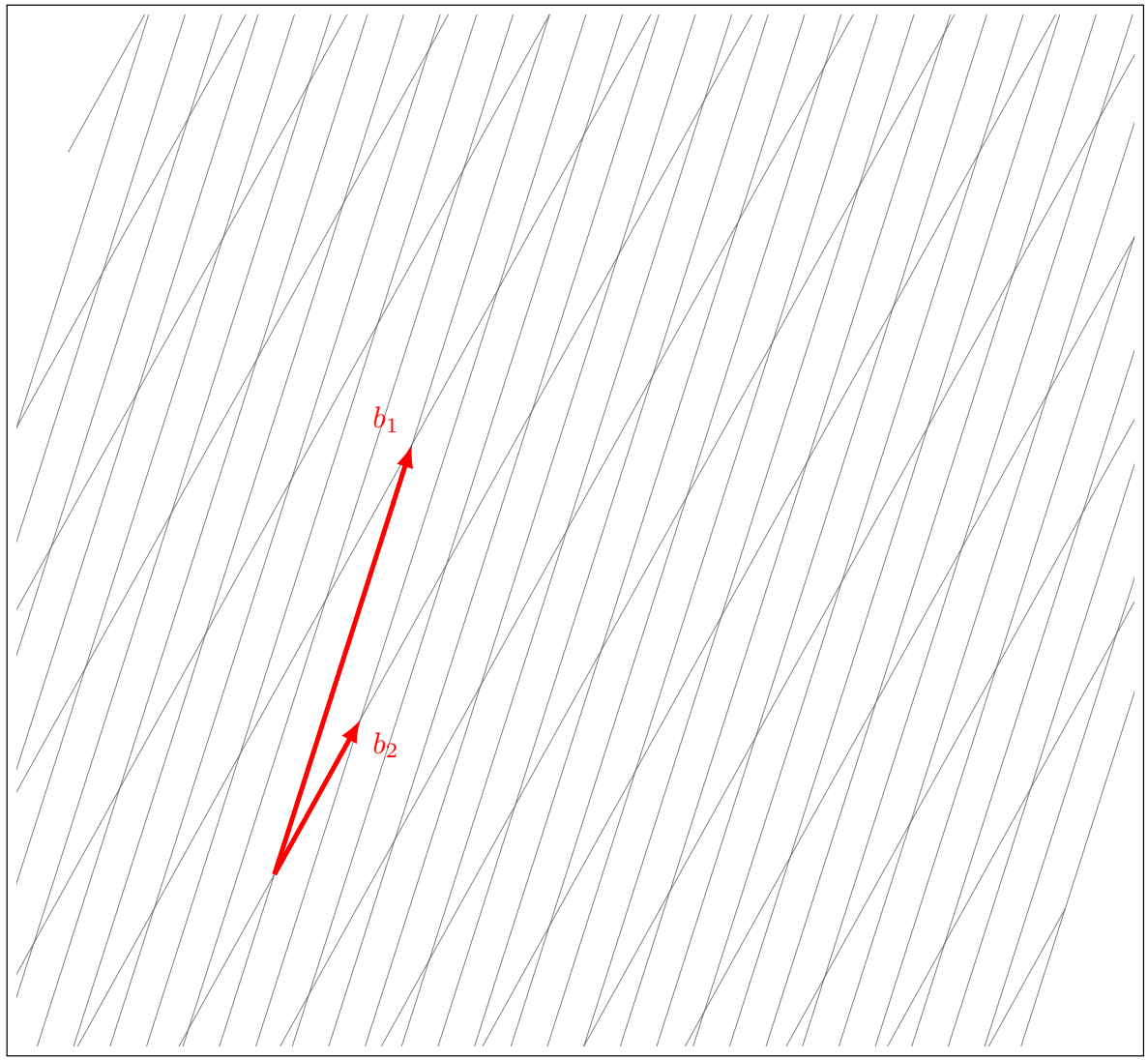
Réduction d'un réseau $\mathcal{L}$:
A partir par d'une base quelconque de $\mathcal{L}$, construire une base réduite de $\mathcal{L}$.

Nombreuses applications :
Calcul formel , géométrie discrète, et cryptologie.

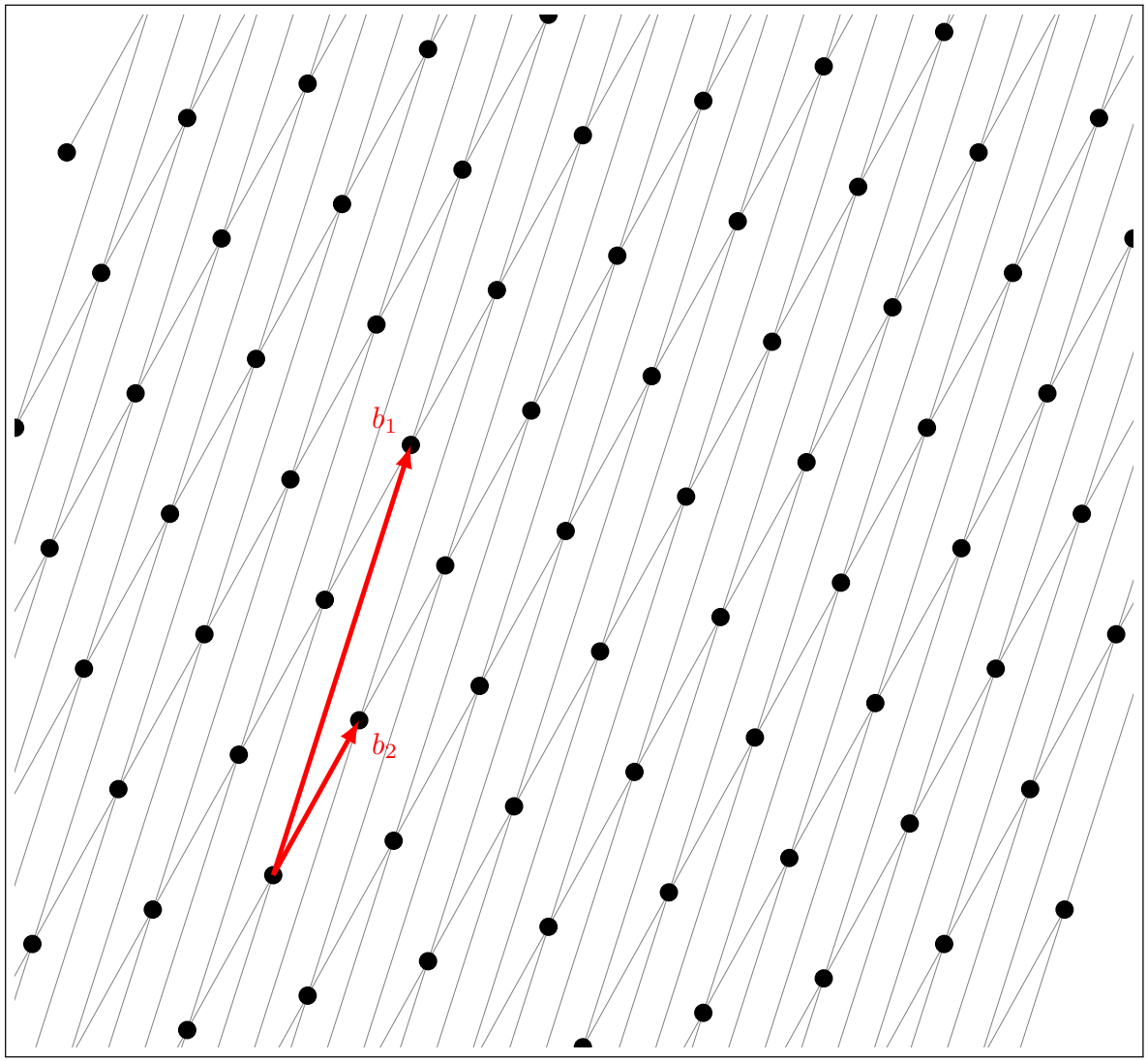
Un réseau euclidien



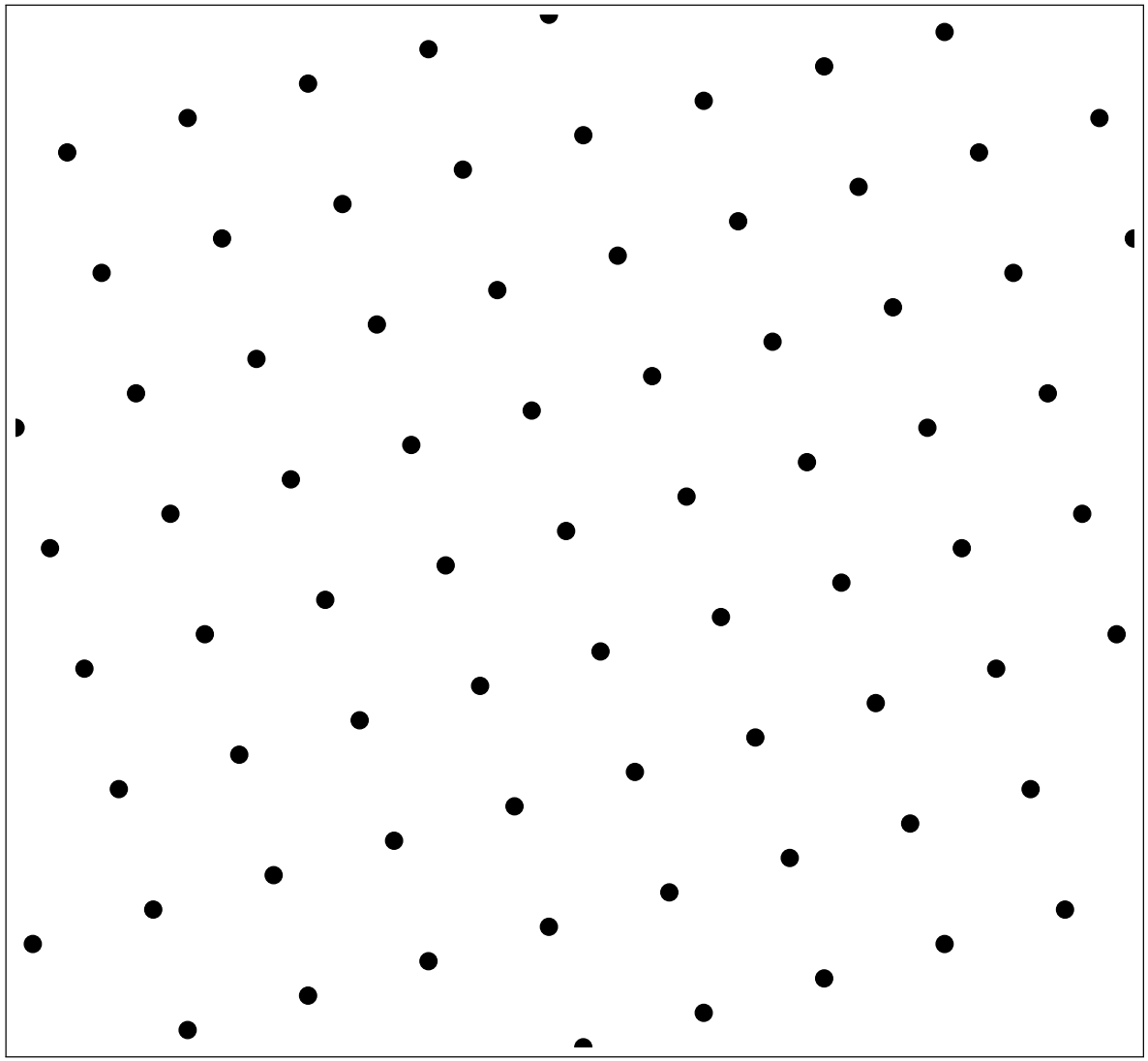
Un réseau euclidien



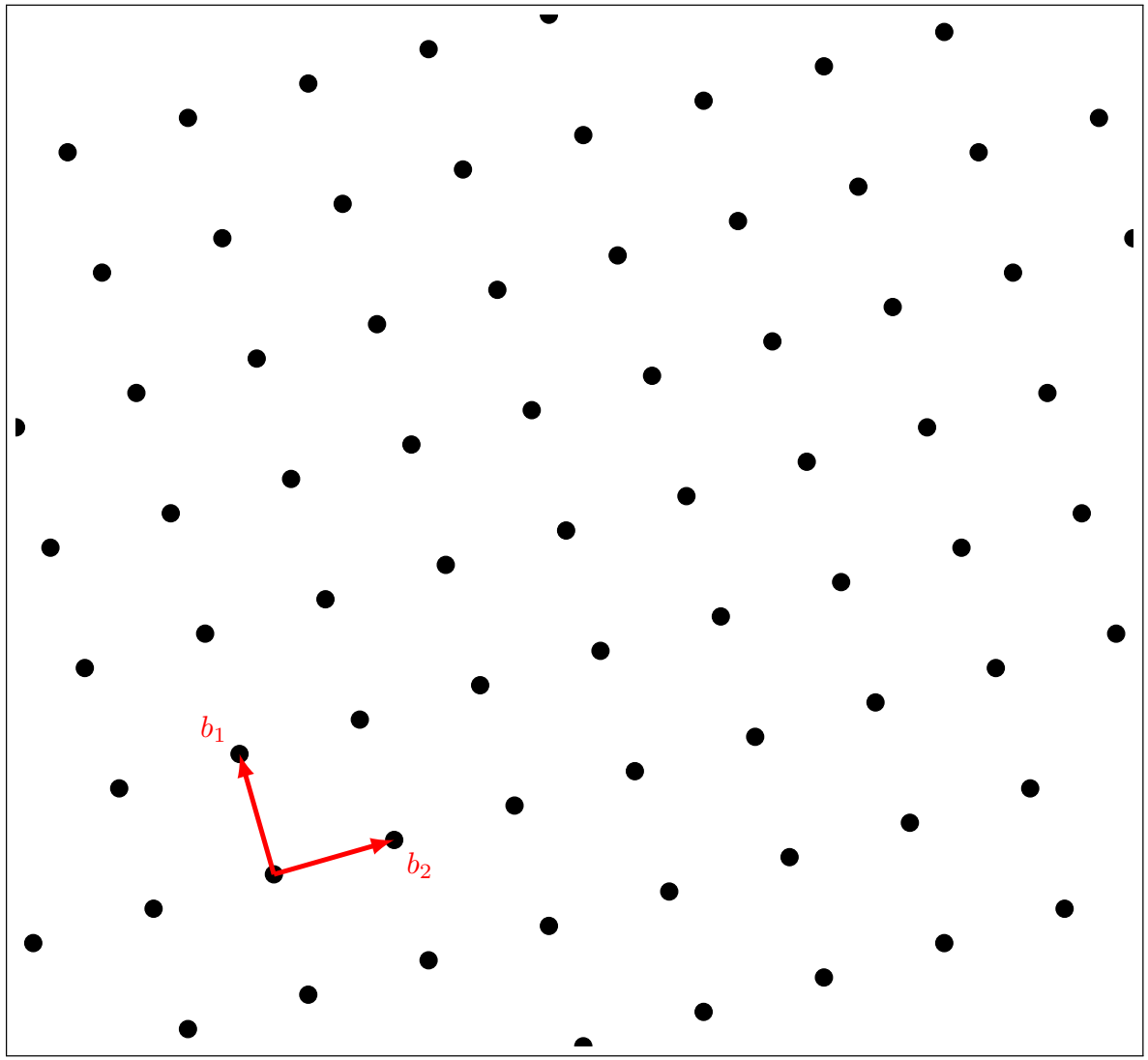
Un réseau euclidien



Un réseau euclidien



Un réseau euclidien



Exemples de problèmes de recherche d'un représentant

1. D un ensemble, muni d'un ordre total.
   Entrée : une séquence de n éléments de D.
   Sortie : Le représentant ordonné dans l'ordre croissant, parmi toutes les séquences qui contiennent les mêmes éléments que l'entrée.
2. Tri topologique.
   Entrée : un graphe orienté acyclique $G(V, E)$, avec $V = \{1, \ldots, n\}$ et une énumération des ses sommets.
   Sortie : une énumération des sommets de G qui respecte l'ordre partiel induit par le graphe G.
3. D le groupe libre Abélien $\mathbb{R}^p$, muni de la structure euclidenne.
   Entrée : une séquence de n éléments de D.
   Sortie : Un représentant parmi toutes les séquences de n éléments qui engendrent le même groupe Abélien libre que la séquence d'entrée et dont les vecteurs sont les plus courts possibles.
4. $D = \{0, 1\}^*$ sur lequel opère un groupe monogène, donnée par la description d'une machine M_f qui calcule l'action d'un générateur f sur un mot.
   Entrée : un élément $x \in D$ et M_f.
   Sortie : un représentant $x^\star$ de l'orbite de x sous l'action de f, dont le poids de Hamming est le plus grand.

Les exemples 1 et 3 précédents serviront de fil conducteur dans tout l'exposé.

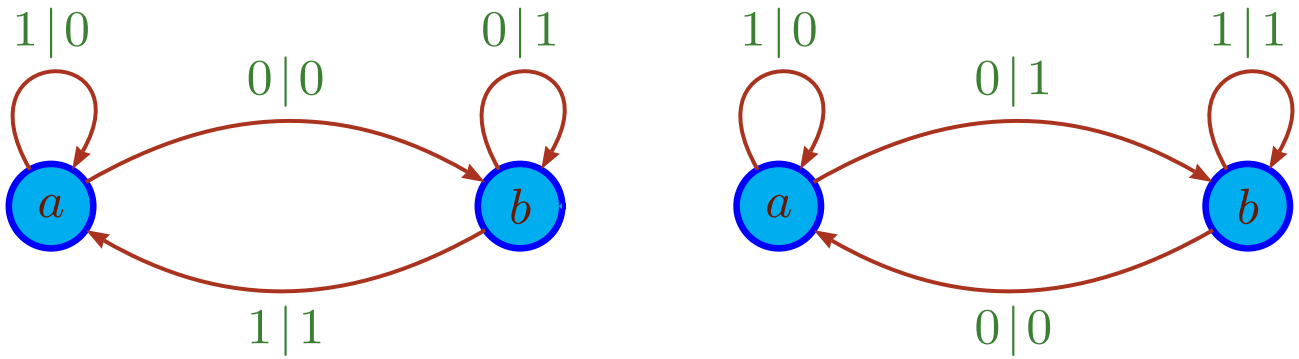
Transducteurs

déterministes, complets, lettre à lettre : Automates de Mealy

$$\big(A, \Sigma, \delta = (\delta_i : A \to A)_{i\in\Sigma}, \rho = (\rho_x : \Sigma \to \Sigma)_{x\in A}\big)$$

- A ensemble fini d'états, ici $A = \{a, b\}$.
- Σ ensemble fini de lettres, ici $\Sigma = \{0, 1\}$.
- δ_i applications
- ρ_x applications

$x \in \{0,1\}^*$, f donné par un état du transducteur.
On cherche $x^\star$ dans $O(x) := (x, f(x), \ldots, f^k(x))$



On définit $\rho_x : \Sigma^n \to \Sigma^n$, pour tout $n \geq 0$, en posant que l'image du mot vide est le mot vide et en utilisant par induction

$$\forall u \in \Sigma,\ \forall v \in \Sigma^*, \qquad \rho_x(uv) = \rho_x(u)\rho_{\delta_u(x)}(v)\ .$$

L'application ρ_x définit ainsi une application de Σ^* dans Σ^*, qui préserve la longueur et les préfixes.

Transducteurs

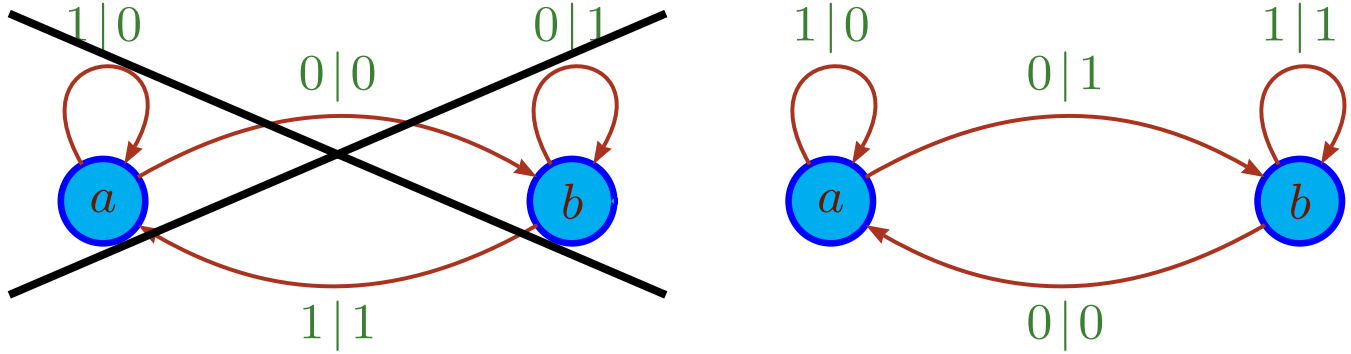
déterministes, complets, lettre à lettre : Automates de Mealy

inversibles

$$\big(A, \Sigma, \delta = (\delta_i : A \to A)_{i\in\Sigma}, \rho = (\rho_x : \Sigma \to \Sigma)_{x\in A}\big)$$

- A ensemble fini d'états, ici $A = \{a, b\}$.
- Σ ensemble fini de lettres, ici $\Sigma = \{0, 1\}$.
- δ_i applications
- ρ_x applications $\rightsquigarrow$ permutations

$x \in \{0,1\}^*$, f donné par un état du transducteur.
On cherche $x^\star$ dans $O(x) := (x, f(x), \ldots, f^k(x))$



On définit $\rho_x : \Sigma^n \to \Sigma^n$, pour tout $n \geq 0$, en posant que l'image du mot vide est le mot vide et en utilisant par induction

$$\forall u \in \Sigma,\ \forall v \in \Sigma^*, \qquad \rho_x(uv) = \rho_x(u)\rho_{\delta_u(x)}(v)\ .$$

L'application ρ_x définit ainsi une application de Σ^* dans Σ^*, qui préserve la longueur et les préfixes.

Le problème Recherche Représentant cas général

Un ensemble X et un groupe G qui opère sur X.
Entrée : un élément $x \in X$.
Sortie : un représentant $x^\star$ de l'orbite de x sous l'action de G qui vérifie une propriété π fixée.

X et G seront liés : On a :

- X : ensemble des bases des algèbres libres de rang n d'une variété d'algèbres.
- G : groupe d'automorphisme de l'algèbre libre de rang n.

On va précise maintenant le rapport entre X et G et donner quelques exemples.

Le problème de recherche d'un représentant : le cas général

On considère maintenant un ensemble D
(avec une structure supplémentaire, D est considéré comme une algèbre libre d'une variété d'algèbres).
Ici une structure algébrique « classique » : ensemble, monoïde, groupe,...
On désigne par D_n une algèbre libre engendrée par n éléments de D.
Les n éléments $x = (x_1, \ldots, x_n)$ de D forment alors une séquence génératrice ou encore une base de D_n. On note : $D_n = \langle x \rangle$.

- Automorphisme de D_n : bijection de D_n dans D_n, compatible avec les opérations de D_n. Il peut être défini avec une paire de bases.
- $\mathtt{Aut}(D_n)$: groupe des automorphismes de D_n.

Le groupe $\mathtt{Aut}(D_n)$ opère naturellement sur D^n.
Une orbite est aussi l'ensemble de toutes les bases d'une même algèbre libre de rang n : $O(x) := \{\sigma \star x : \sigma \in \mathtt{Aut}(D_n)\} = \{x' \in D : \langle x' \rangle = \langle x \rangle\}$.
Le problème s'énonce alors :

Problème Recherche Représentant (D, n, π) :

- **Entrée :** un élément $x \in D^n$.
- **Sortie :** $x^\star \in O(x)$ vérifiant une propriété π définie via la structure de D.

On illustre maintenant ce qui est annoncé sur quelques exemples.

Exemples d'algèbres libres de rang n d'une variété.

- Les ensembles.
  Aucune opération.
  Les deux seuls termes construits sur deux variables x_1 et x_2 sont : x_1, x_2.

- Les monoïdes.
  Il y a une opération binaire notée $\cdot$ et une constante notée 1.
  les identités (ou axiomes) $\quad x.(y.z) = (x.y).z, \quad x.1 = 1.x = x.$
  Les termes construits sur les deux variables x_1 et x_2 sont :
  $$x_1, x_2, x_1 \cdot x_2, x_1 \cdot x_1, x_2 \cdot x_2 \ldots$$

- Les groupes.
  Il y a une opération binaire notée $\cdot$, une opération unaire notée $^{-1}$ et une constante notée 1.
  Les identités(ou axiomes) :
  $$x.(y.z) = (x.y).z, \quad x.1 = 1.x = x, \quad x.x^{-1} = x^{-1}.x = 1$$
  Les termes construits sur les deux variables x_1 et x_2 sont donc
  $$x_1, x_2, x_1^{-1}, x_2^{-1}, x_1 \cdot x_2^{-1}, x_1 \cdot x_1, \ldots$$

- Les anneaux.
  Il y a deux opérations binaires $\cdot$ et $+$, une opération unaire $-$ et une fonction nullaire notée 1.
  L'ensemble des termes construits sur la variable x_1 est l'ensemble $\mathbb{Z}[x_1]$ des polynômes à une indéterminée à coefficients sur $\mathbb{Z}$.

Groupe d'automorphismes. Exemples.

Un (auto)morphisme f de D_n peut être définie par la donnée des images des éléments d'une base, exprimée dans la base. Par exemple, si $(x_1, \ldots, x_n)$ est une base de D_n, f peut être donnée par n *termes* $(t_1, \ldots, t_n)$ construits sur $(x_1, \ldots, x_n)$:

$$f \begin{cases} x_1 & = & t_1(x_1, \ldots, x_n) \\ \vdots & \vdots & \vdots \\ x_n & = & t_n(x_1, \ldots, x_n) \end{cases}.$$

Pour une algèbre libre D_n de rang n d'une variété, il n'est pas toujours facile de déterminer $\mathtt{Aut}(D_n)$. Voici deux exemples, où $\mathtt{Aut}(D_n)$ est bien connu.

- Pour un ensemble D_n de cardinal n ,
  le groupe $\mathtt{Aut}(D_n)$ est $\mathfrak{S}_n$= l'ensemble des permutations sur n éléments.
  Exemples d'éléments de $\mathtt{Aut}(D_3)$

$$f \begin{cases} x_1 & = & x_2 \\ x_2 & = & x_1 \\ x_3 & = & x_3 \end{cases} \qquad g \begin{cases} x_1 & = & x_1 \\ x_2 & = & x_3 \\ x_3 & = & x_2 \end{cases}$$

- Pour D_n groupe libre de rang n, le groupe $\mathtt{Aut}(D_n)$ a été déterminé dans les années 1930 par Nielsen. Exemples d'éléments f, g, h de $\mathtt{Aut}(D_3)$

$$f \begin{cases} x_1 & = & x_1^{-1} \\ x_2 & = & x_2 \\ x_3 & = & x_3 \end{cases} \qquad g \begin{cases} x_1 & = & x_1 \\ x_2 & = & x_1 x_2 \\ x_3 & = & x_3 \end{cases} \qquad h \begin{cases} x_1 & = & x_2 \\ x_2 & = & x_1 \\ x_3 & = & x_3 \end{cases}$$

Le problème Recherche Représentant : récapitulatif

- un ensemble X et un groupe G qui opère sur X.
  Entrée : un élément $x \in X$.
  Sortie : un représentant $x^\star$ de l'orbite de x sous l'action de G qui vérifie une propriété fixée π.

- D : une algèbre libre d'une variété (ou encore un modèle d'une théorie).
  Il existe également des relations sur D (exemples : ordre, préodre, ...).
- D_n : une algèbre libre de rang n de la variété.
- $G = \mathtt{Aut}(D_n)$: groupe des automorphismes de D_n.
- $X = D^n$: l'ensemble des séquences de n éléments de D.
- Le groupe $\mathtt{Aut}(D_n)$ opère naturellement sur D^n.
  $\sigma \in \mathtt{Aut}(D_n)$ et $(x_1, \ldots, x_n) \in D^n$, $\sigma \star x := (\sigma(x_1), \ldots, \sigma(x_n))$.
- Une orbite est aussi l'ensemble de toutes les bases d'une même algèbre libre de rang n
  $O(x) := \{\sigma \star x : \sigma \in \mathtt{Aut}(D_n)\} = \{x' \in D : \langle x' \rangle = \langle x \rangle\}$.

Problème Recherche Représentant(D, n, π) :
Entrée : $x \in X$.
Sortie : $x^\star \in O(x) \iff \langle x \rangle = \langle x^\star \rangle$
$\pi(x^\star)$ est vrai.

Recherche d'un représentant : nos deux exemples « fil conducteur »

Premier exemple. Ensembles totalement ordonnés et problème du tri.

Un ensemble D muni d'un ordre total, $D_n \subset D$, $|D_n| = n$.
Le groupe $\mathfrak{S}_n$ des permutations de D_n opère sur D^n.
Pour $x \in D^n$ et $\sigma \in \mathfrak{S}_n$, on note $\sigma \star x = (\sigma(x_1), \ldots, \sigma(x_n))$.
L'orbite de x, désignée par $O(x)$ est $\{\sigma \star x \mid \sigma \in \mathfrak{S}_n\}$.

Problème du tri.

- **Entrée :** $x \in D^n$
- **Sortie :** $x^\star \in O(x)$ ordonné dans l'ordre croissant.

Deuxième exemple. Réseaux euclidiens et problème de la réduction.

D un groupe libre abélien, muni d'un produit scalaire (ex : $\mathbb{R}^p$ ou $\mathbb{Z}^p$)
D_n le sous groupe libre engendré par n éléments de D.
Le groupe $GL(n, \mathbb{Z})$ des automorphismes de D_n opère sur D^n.
Pour $x \in D^n$ et $\sigma \in GL(n, \mathbb{Z})$, on note $\sigma \star x = (\sigma(x_1), \ldots, \sigma(x_n))$.
L'orbite de x, désignée par $O(x)$ est $O(x) := \{\sigma \star x \mid \sigma \in GL(n, \mathbb{Z})\}$

Problème de la réduction.

- **Entrée :** $x \in D^n$
- **Sortie :** $x^\star \in O(x)$ dont les vecteurs sont les « plus courts possibles ».

(II) Algorithmes pour le problème de la recherche du représentant.

Algorithme pour le problème de la recherche d'un représentant

- Une algèbre D considérée comme une algèbre libre d'une variété d'algèbres
- Algorithme $A : D^n \to D^n$, qui termine quand la propriété $\pi(x)$ est vérifiée
- un ensemble $\mathcal{F}$ de transformations élémentaires ,
$$\mathcal{F} \subset \mathtt{Aut}(D_n), \quad \mathcal{F} \quad \text{engendre} \quad \mathtt{Aut}(D_n).$$
- une stratégie $\mathcal{S}$ déterministe

  qui détermine la transformation élémentaire à appliquer à chaque étape.

  définie via la structure sur D et prolongée sur D^n.

Algorithme $A(D, n, \pi)(\mathcal{F}, \mathcal{S})$

Entrée : $x \in D^n$.
Tant que non $\pi(x)$ **faire**
$f := \mathcal{S}(x)$
$x := f(x)$
Retourner x

Remarque : Un tel algorithme n'utilise que des transformations bijectives.
S'il termine, il résout le problème $\mathcal{P}(D, n, \pi)$.

Rappel de la modélisation des algorithmes par Gurevich : les trois postulats

I Un algorithme déterministe séquentiel est un système à transition d'états.
Un état $\rightsquigarrow$ Une photo instantanée de la mémoire pendant une exécution.
Une exécution de l'algorithme est donc une séquence d'états.

II Les états d'un algorithme sont des algèbres (des structures) de même vocabulaire et, le long d'un exécution, de même univers.
Tout est invariant par isomorphisme :

- L'ensemble des états est clos par isomorphisme.
- La fonction de transition commute avec tout isomorphisme.

III Les transitions d'un état à l'état suivant sont gouvernées par une description fixe et finie :
Les transitions sont déterminées par un ensemble fini de termes "critiques".

Plus précisément, il existe un ensemble fini de termes du vocabulaire des états, tel que si deux états coïncident sur les valeurs des termes critiques, alors ils seront modifiés exactement de même façon.

Une machine à états abstraits (ASM) de Gurevich

Un ASM est la donnée de

- un ensemble S d'états, tous des algèbres de même vocabulaire. S est clos par isomorphisme,
- un programme P , qui consiste en un nombre fini d'instructions, toutes de la forme : p alors pour $1 \leq i \leq k$, $t_i := u_i$,
  avec t_i et u_i des termes du vocabulaire et p un terme relationnel.
  $\{t_i := u_i : 1 \leq i \leq k\}$ est appelé une règle de mise à jour.

Théorème (Gurevich 00)

Tout processus de calcul satisfaisant les trois postulats (i.e. tout algorithme séquentiel déterministe raisonnable) est simulé pas à pas par un ASM.

$+$ un quatrième postulat $\rightarrow$ Preuve de la thèse de Church
(Dershowitz et Gurevich 08).

Théorème : Tout ASM *nullaire*, *réversible* qui finit toujours, résout un problème de recherche de représentant.

Retour sur un algorithme pour le problème de la recherche d'un représentant

- Une algèbre D considérée comme une algèbre libre d'une variété d'algèbres
- Algorithme $A : D^n \to D^n$, qui termine quand la propriété $\pi(x)$ est vérifiée
- un ensemble $\mathcal{F}$ de transformations élémentaires ,
$$\mathcal{F} \subset \mathtt{Aut}(D_n), \quad \mathcal{F} \quad \text{engendre} \quad \mathtt{Aut}(D_n).$$
- une stratégie $\mathcal{S}$ déterministe

  qui détermine la transformation élémentaire à appliquer à chaque étape.

  définie via la structure sur D et prolongée sur D^n.

Algorithme $A(D, n, \pi)(\mathcal{F}, \mathcal{S})$

Entrée : $x \in D^n$.
Tant que non $\pi(x)$ **faire**
$f := \mathcal{S}(x)$
$x := f(x)$
Retourner x

Remarque : Un tel algorithme n'utilise que des transformations bijectives.
S'il termine, il résout le problème $\mathcal{P}(D, n, \pi)$.

Identification entre les entrées et les automorphismes

On considère le graphe (orienté et coloré) des exécutions potentielles :

- L'ensemble des sommets est l'ensemble D^n des entrées de l'algorithme.
- Il y un arc de couleur f entre deux sommets x et y ssi $y = f(x)$.
  Chaque transformation élémentaire donne lieu à une couleur
  L'ensemble des arcs est $\{(x, f(x)) : x \in D^n \text{ et } f \in \mathcal{F}\}$.
- Si l'algorithme A termine toujours, alors l'ensemble des traces d'exécutions de A définit une forêt couvrante du graphe des exécutions potentielles.

On considère aussi le graphe de Cayley du groupe $\mathtt{Aut}(D_n)$ avec le système générateur $\mathcal{F}$.

Théorème On considère

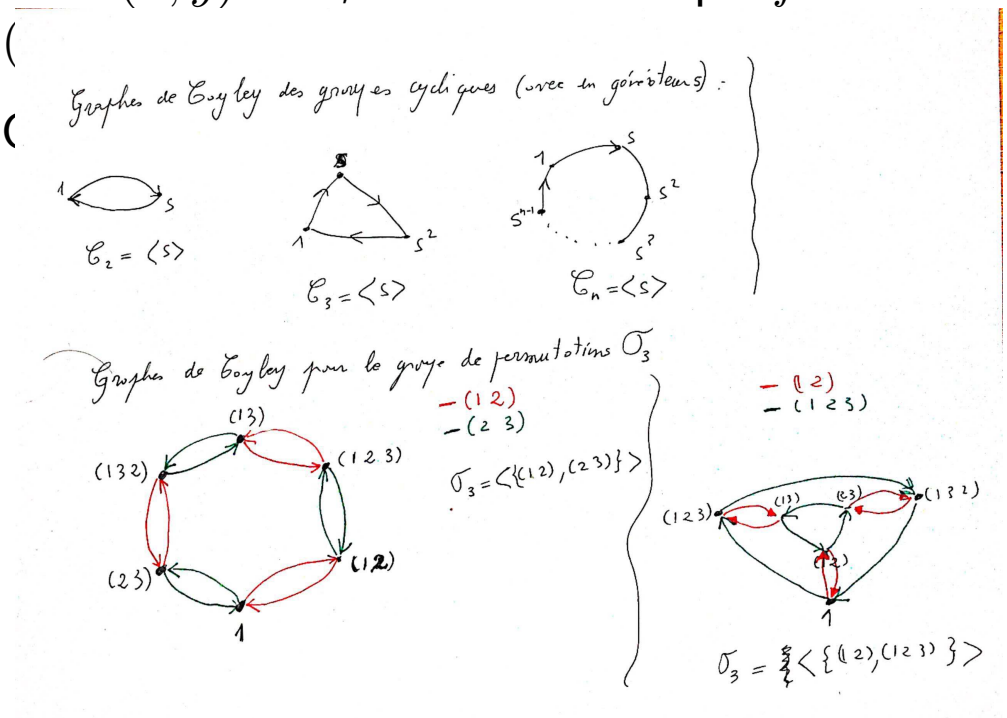
- un problème de représentant $\mathcal{P}(D, n, \pi)$,
- un algorithme $A : D^n \to D^n$ qui résout ce problème,
- le graphe de ses exécutions potentielles,
- le graphe de Cayley du groupe $\mathtt{Aut}(D_n)$ avec le système générateur $\mathcal{F}$.

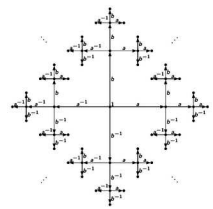
Alors, les composantes connexes du graphe des exécutions potentielles de l'algorithme A sont toutes isomorphes entre elles et isomorphes au graphe de Cayley du groupe $\mathtt{Aut}(D_n)$ avec le système générateur $\mathcal{F}$.

Rappel : graphe de Cayley

Soit G un groupe et $\mathcal{F}$ une partie génératrice de G.

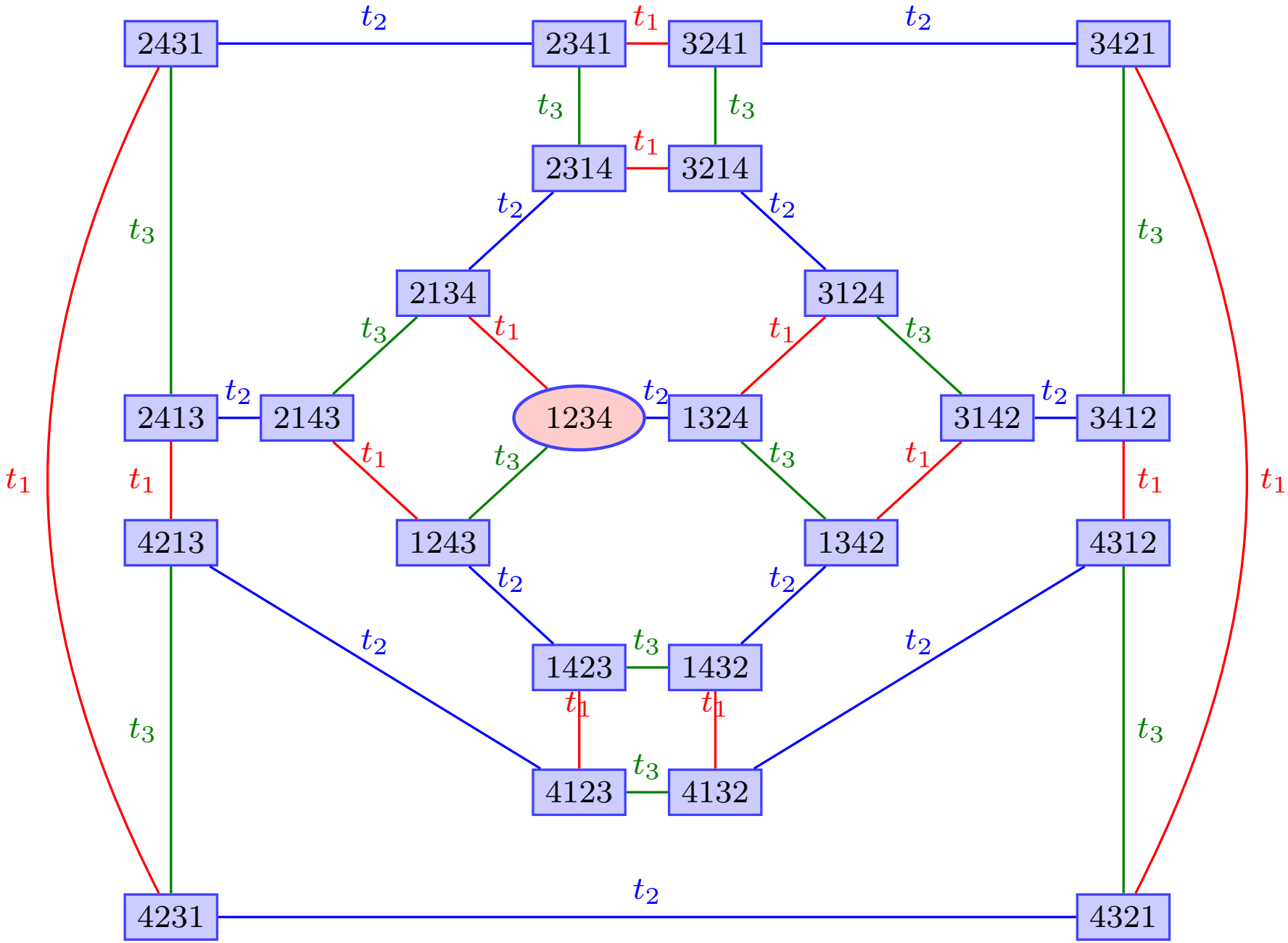
Le graphe de Cayley du groupe G avec le système générateur $\mathcal{F}$ est le graphe $Cay(G, \mathcal{F}) = (V, E)$ où $V := G$ et $E := \{(x, xf), f \in \mathcal{F}\}$.
Pour $(x, y) \in E$, il existe un unique $f \in \mathcal{F}$ tel que $y = xf$. La couleur de l'arc (

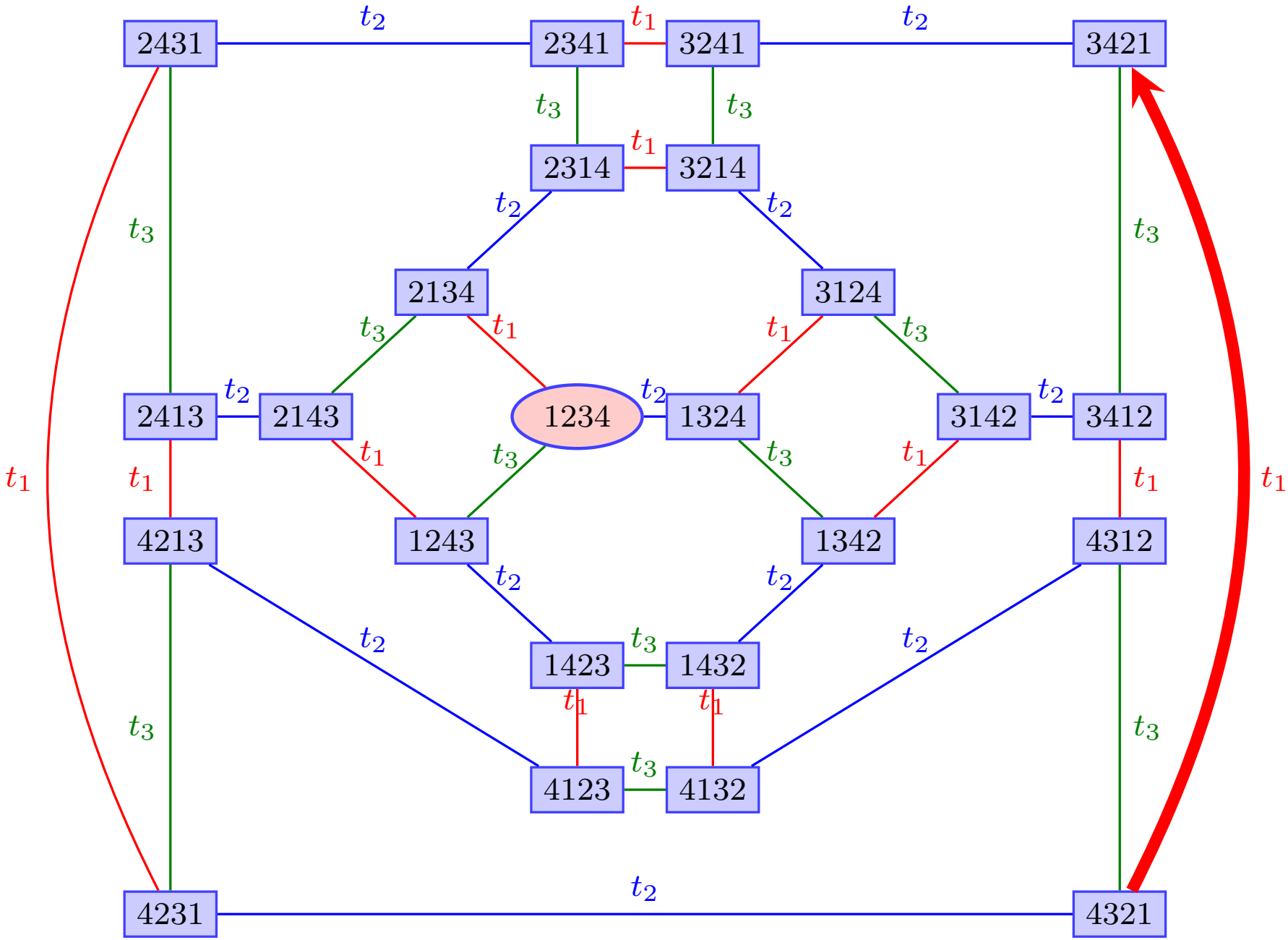


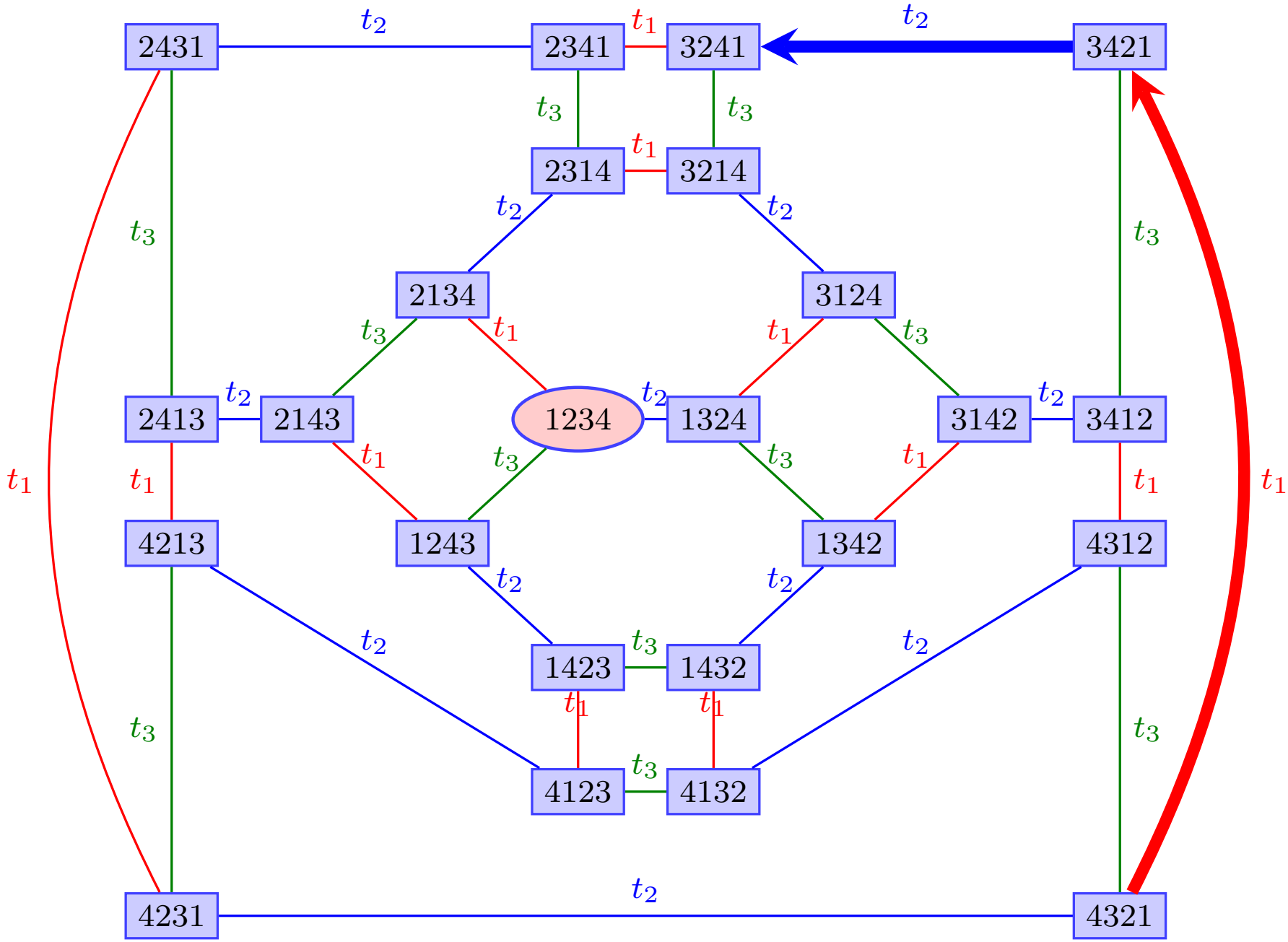


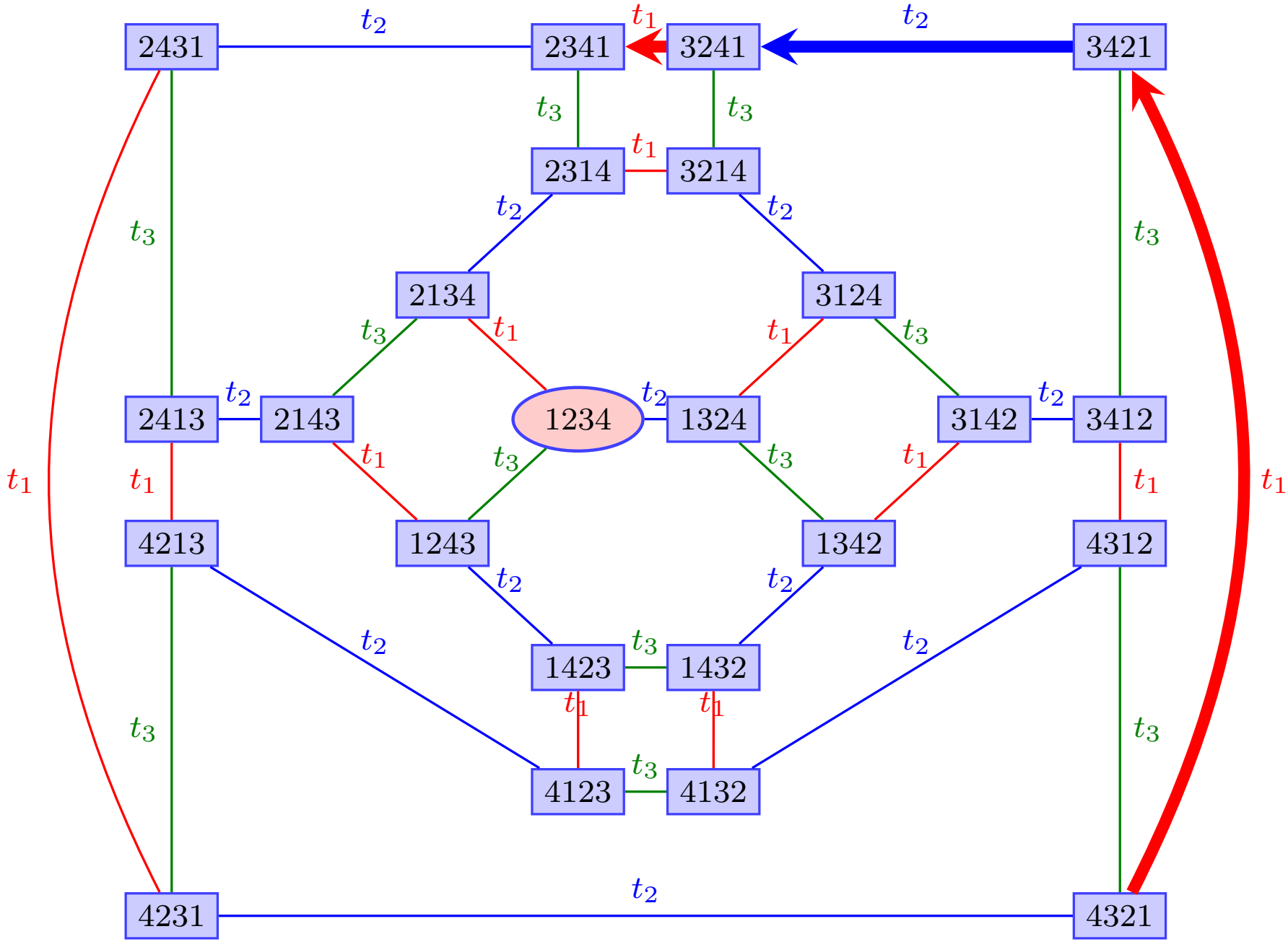
Exemple du tri insertion sur l'ensemble $\{1, 2, 3, 4\}$.

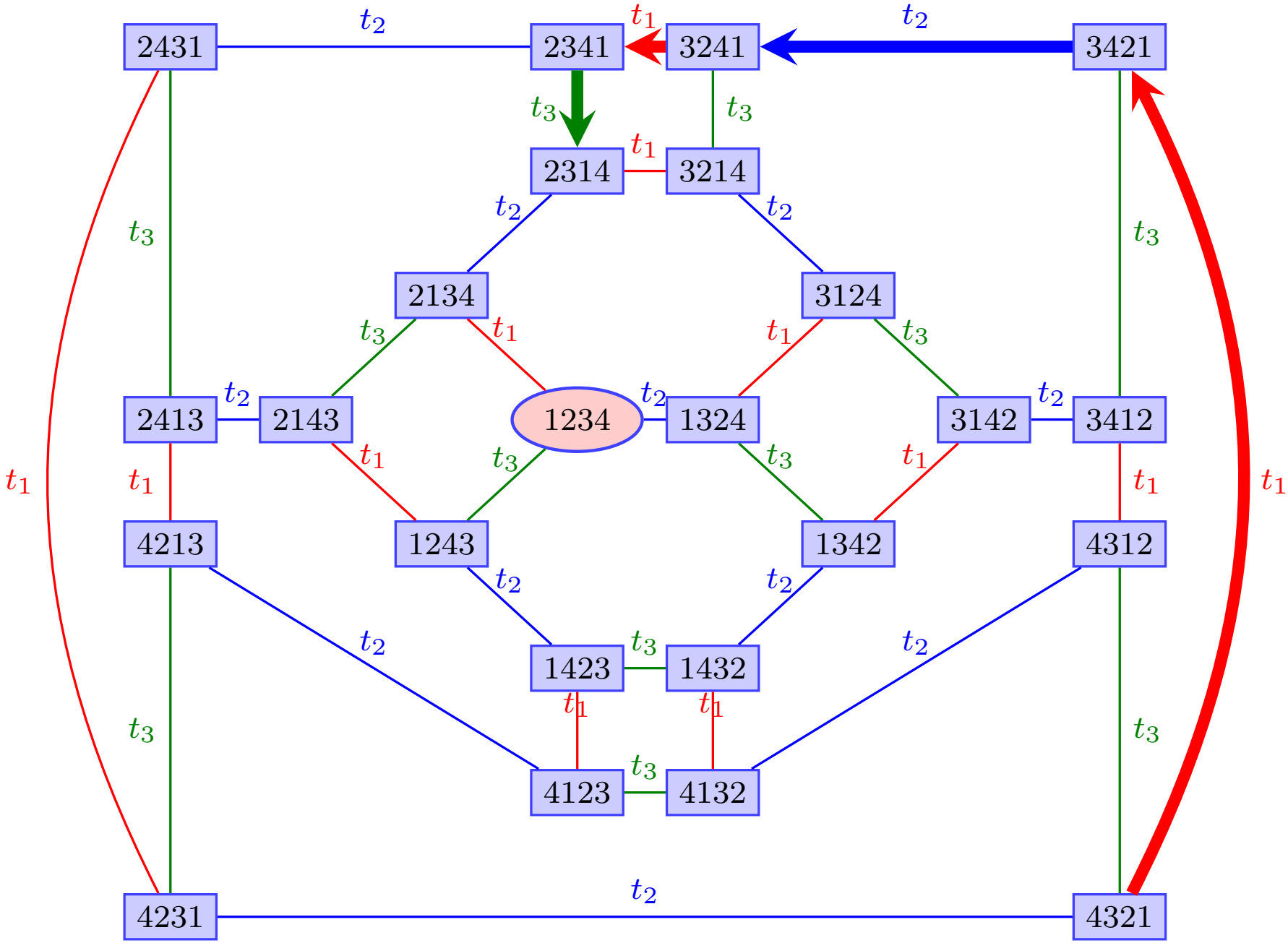
- Il y a trois transformations élémentaires :
  pour $1 \leq i \leq 3$, t_i échange le $i$ème élément et le $(i+1)$ème élément.
- pour chaque entrée, il existe une sortie unique.
- Les traces d'exécution forment une forêt de recouvrement du *graphe des exécutions potentielles* de l'algorithme.

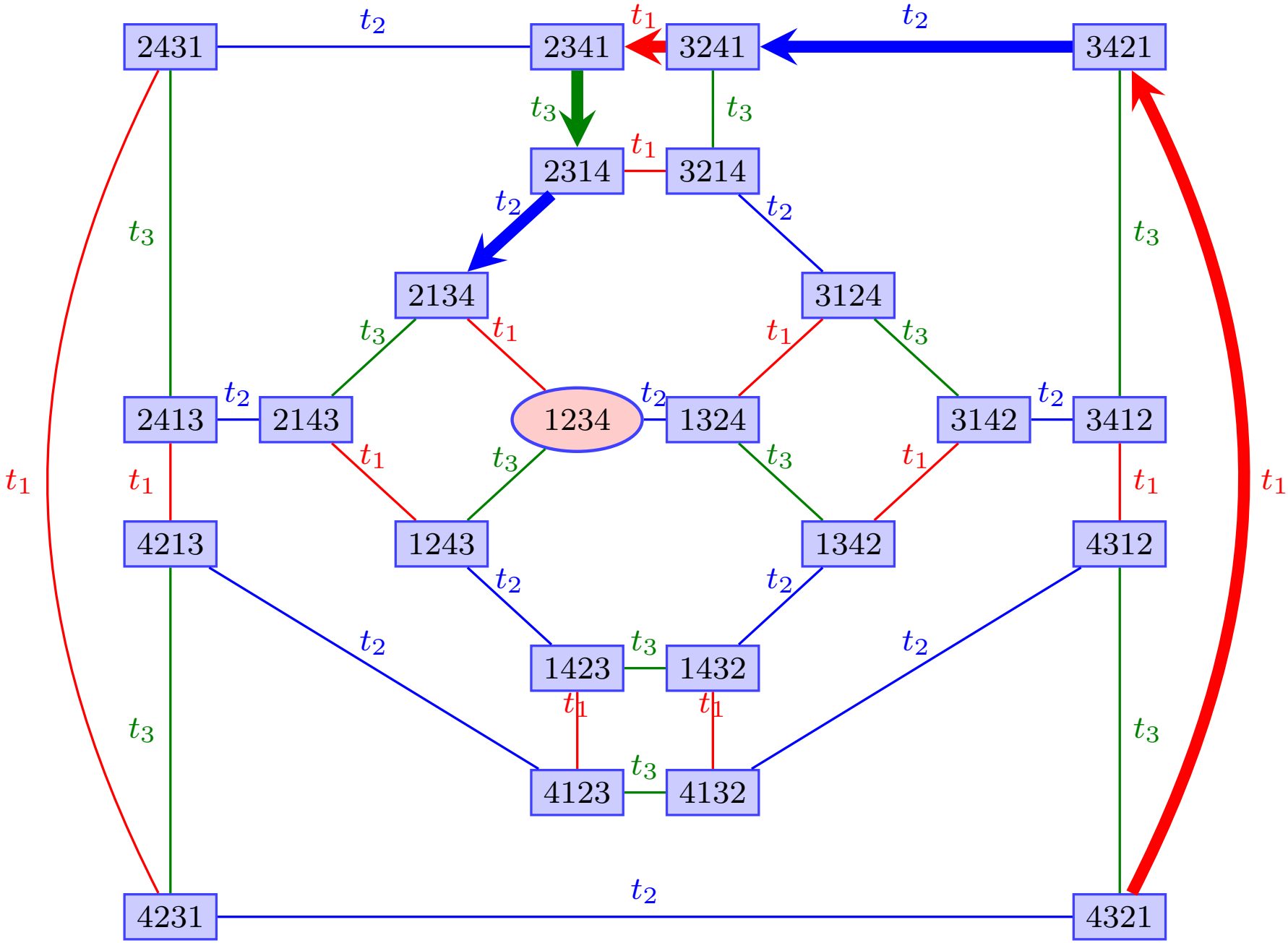

2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3

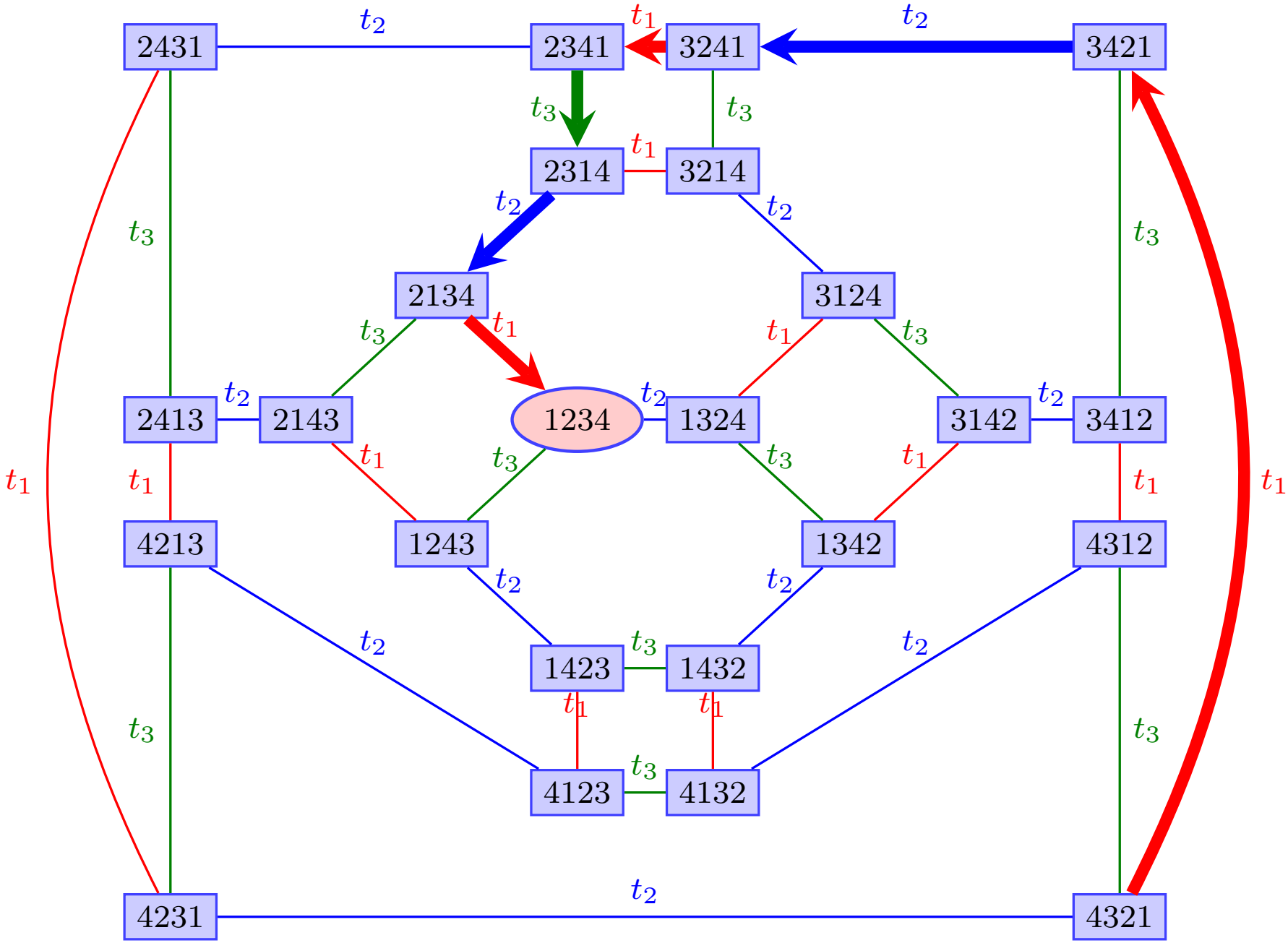

2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3

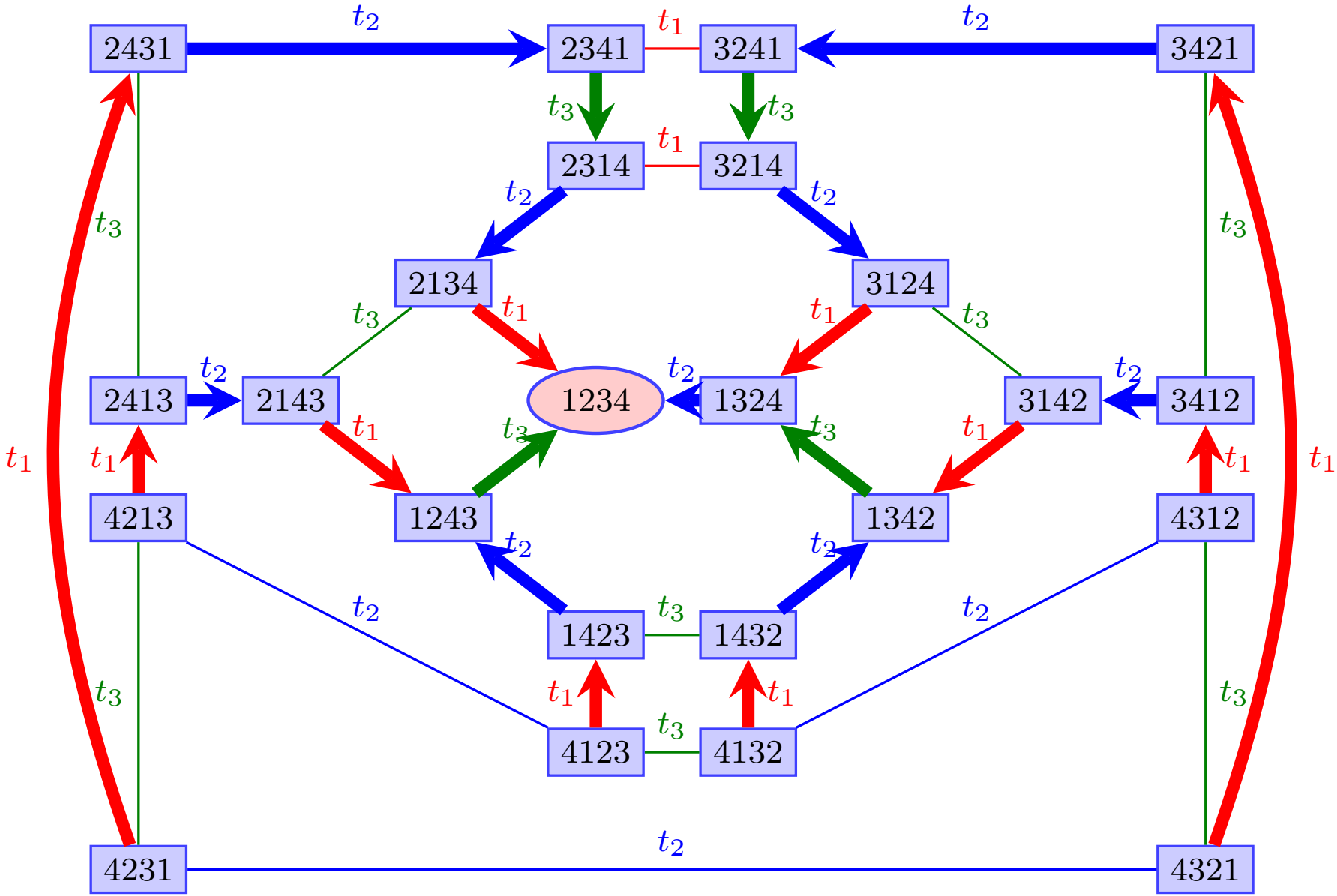

2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3


2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3


2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3


2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3


2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3


2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3

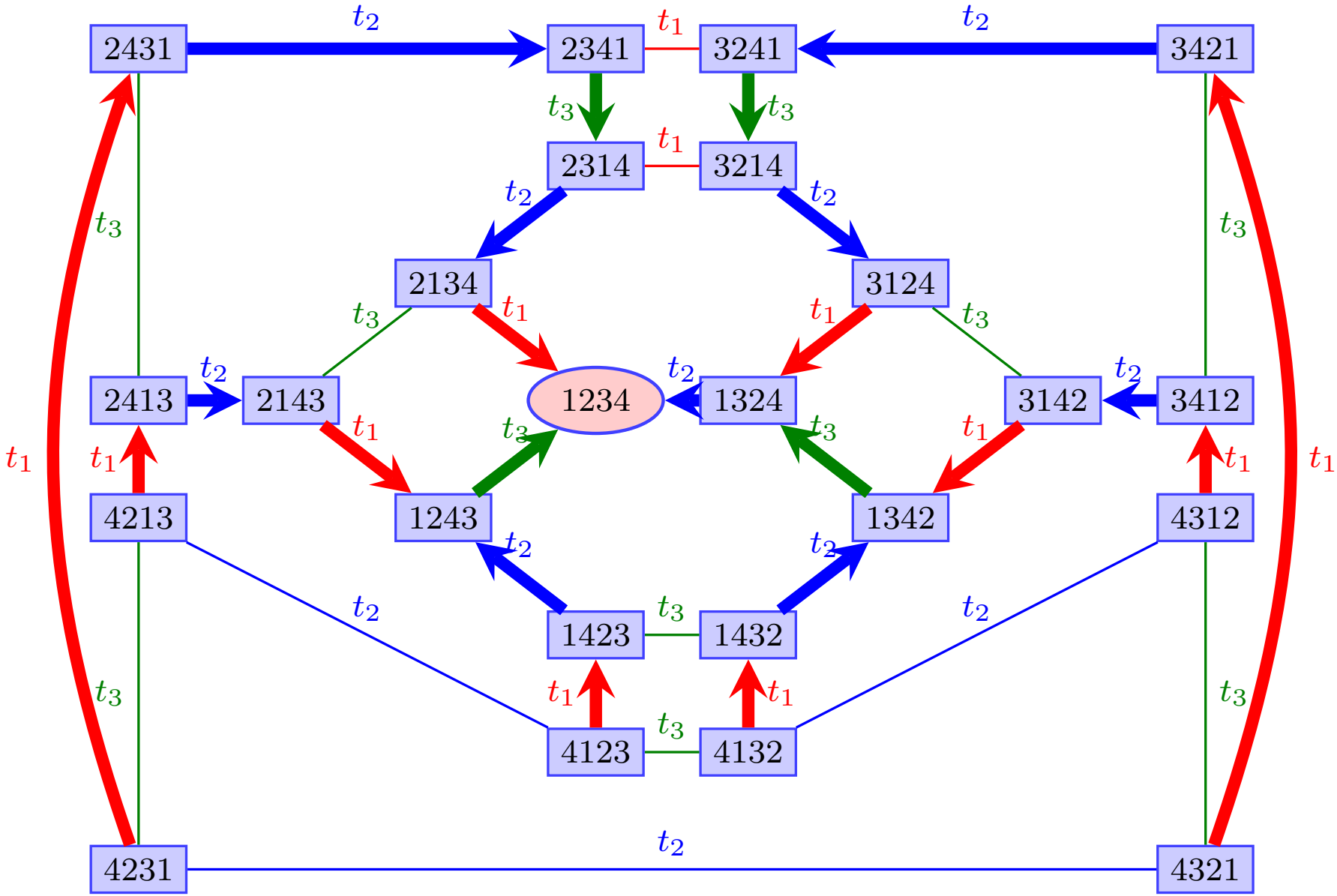
(III) Quelques applications

1. Recherche de borne inférieure et complexité
2. Caractérisation du langage des traces d'un algorithme.

Bornes inférieures, quand $\mathtt{Aut}(D_n)$ est fini.

Théorème Considérons le problème $\mathcal{P}(D, n, \pi)$ avec les hypothèses supplémentaires suivantes
(i) le groupe $\mathtt{Aut}(D_n)$ est fini
(ii) pour $x \in D^n$, le représentant $x^\star$ est unique.
Considérons un algorithme A qui résout le problème $\mathcal{P}(D, n, \pi)$ avec un système $\mathcal{F}$ de transformations élémentaires.
Alors, le diamètre du graphe de Cayley du groupe $\mathtt{Aut}(D_n)$ avec le système générateur $\mathcal{F}$ est une borne inférieure pour la complexité de l'algorithme A.

Corollaire Pour tout algorithme de tri de n éléments, dont les transformations élémentaires sont les échanges de deux éléments d'indices consécutifs, il existe une entrée qui nécessite une exécution en $n(n-1)/2$ étapes.
(i. e. l'algorithme a une borne inférieure de complexité en $n(n-1)/2$.)


2431
2341
3241
3421
2314
3214
2134
3124
2413
2143
1234
1324
3142
3412
4213
1243
1342
4312
1423
1432
4123
4132
4231
4321
t_1
t_2
t_3

Bornes inférieures, quand $\mathtt{Aut}(D_n)$ est infini (travail en cours).

Théorème (travail en cours) Considérons le problème $\mathcal{P}(D, n, \pi)$ avec les hypothèses supplémentaires suivantes
(i) le groupe $\mathtt{Aut}(D_n)$ est infini
(ii) pour $x \in D^n$, le représentant $x^\star$ est unique.
Considérons un algorithme A qui résout le problème $\mathcal{P}(D, n, \pi)$ avec
(iii) l'ensemble $\mathcal{F}$ des transformations élémentaires de A est fini
(iv) l'action d'une transformation élémentaire *n'augmente pas significativement* la taille d'une donnée :

$$\forall k \in \mathbb{N},\ \forall x \in X, \forall f_1, \ldots, f_k \in \mathcal{F}, \quad |f_1 \ldots f_k(x)| = \tilde{O}_x(\log(k)).$$

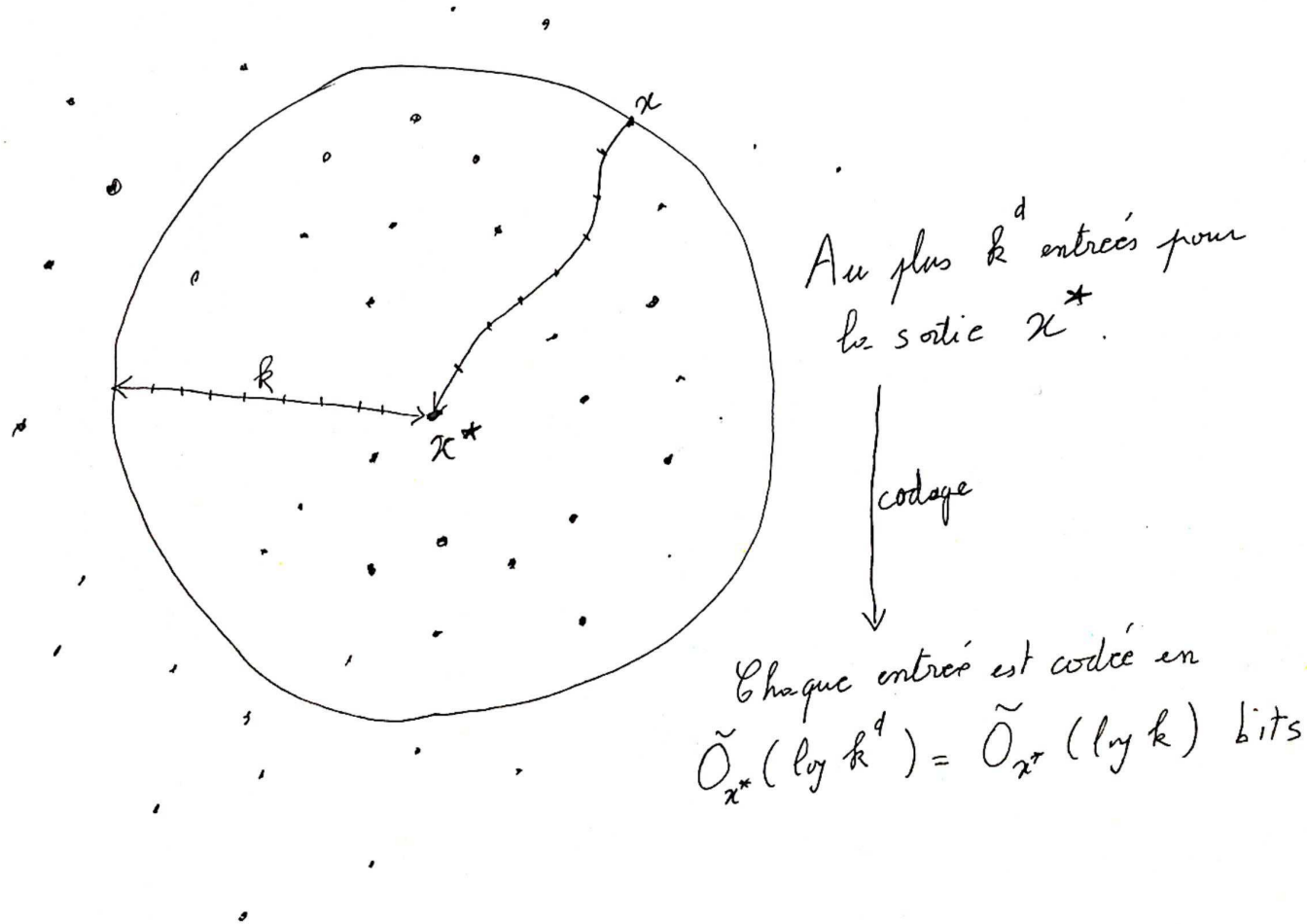
Alors l'algorithme A a une complexité algébrique exponentielle.

Remarque : La condition (iv) nécessite que $\mathtt{Aut}(D_n)$ soit à croissance polynomiale.

Corollaire (travail en cours) Sous les hypothèses du théorème précédent, si de plus (v) le calcul des itérés des transformations élémentaires sur une donnée ne peut pas être *très significativement accéléré* :

$\forall k \in \mathbb{N},\ \forall x \in X, \forall f_1, \ldots, f_k \in \mathcal{F}$, le calcul de $f_1 \ldots f_k(x)$ se fait en temps $\Omega(k)$

Alors l'algorithme A a une complexité en bit exponentielle.



$\rightsquigarrow$ Il existe donc des entrées de taille $\tilde{O}_x(\log(k))$ qui nécessitent k transformations élémentaires.

La croissance d'un groupe finiement engendré ne dépend pas du système de générateurs choisi.

$\rightsquigarrow$ Le résultat s'applique à *tout algorithme*.

(III) Quelques applications

1. Recherche de borne inférieure et complexité.
2. Caractérisation du langage des traces d'un algorithme.

Algorithme $A(D, n, \pi)(\mathcal{F}, \mathcal{S})$

Entrée : $x \in D^n$.
Tant que non $\pi(x)$ **faire**
$f := \mathcal{S}(x)$
$x := f(x)$
Retourner x

- Entrée : Une séquence $(f_1, f_2, \ldots, f_k) \in \mathcal{F}^*$
- Question : La séquence $(f_1, f_2, \ldots, f_k) \in F^*$ est-elle une trace d'exécution de l'algorithme ?
  Plus précisément, existe-t-il $x \in D^n$, tel que sur l'entrée x l'algorithme s'arrête après avoir effectué exactement la séquence de transformation élémentaire $(f_1, f_2, \ldots, f_k)$?

Caractérisation du langage des traces d'exécution : un exemple

Théorème (AA, C. Moreira)

Soient $\mathcal{F} := \{t_1, \ldots, t_{n-1}\}$, i *et* j *deux entiers compris entre* 1 *et* $n-1$ *et* ω *un mot de* Σ^*. *On considère les règles de réécriture suivantes sur* Σ^*.

$$\textit{alors } (t_i, \omega) > 1, \textit{ alors } t_i\, \omega\, t_i \longrightarrow \omega; \tag{1}$$

$$\textit{Si } (t_{i+1}, \omega) > 1, \textit{ alors } t_{i+1}\, t_i\, \omega\, t_{i+1} \longrightarrow t_i\, t_{i+1}\, t_i\, \omega; \tag{2}$$

$$\textit{Si } (t_{i+1}, \omega) > 1, \textit{ alors } t_{i+1}\, \omega\, t_i \longrightarrow \omega\, t_{i+1}\, t_i; \tag{3}$$

$$\textit{Si } j - i > 1, \textit{ alors } t_{j+1}\, t_j\, t_i \longrightarrow t_{j+1}\, t_i\, t_j. \tag{4}$$

$$\textit{Si } (t_i, \omega) > 1, \textit{ alors } t_{i+1}\, \omega\, t_i \longrightarrow t_{i+1}\, t_i\, \omega; \tag{5}$$

$$\textit{Si } i - j > 1, \textit{ alors } t_i\, t_j\, t_{j-1} \longrightarrow t_j\, t_i\, t_{j-1}. \tag{6}$$

Soient σ *une permutation de* $\{1, \ldots, n\}$ *et* ω_1 *une décomposition de* σ *en terme de transpositions* $(t_i)_{0<i<n}$. *La décomposition de* σ *effectuée par l'algorithme du tri insertion s'obtient en appliquant un nombre fini de fois les règles* (1), (2), (3), (4). *La décomposition de* σ *effectuée par l'algorithme du tri séléction s'obtient en appliquant un nombre fini de fois les règles* (1), (2), (5), (6).

Indécidabilité du problème général

Théorème (AA, J. Cassaigne, J. Clément)

Le problème suivant appelé Trace Admissible *est indécidable.*

- Entrée : *Un système dynamique symbolique donné par*
  - *un ensemble décidable X,*
  - *un ensemble fini $\mathcal{F}$ de permutations calculables sur X (avec $Id_x \in \mathcal{F} \subset \mathfrak{S}_{\mathcal{X}}$)*
  - *Une famille $(P_f)_{f\in\mathcal{F}}$ de sous-ensembles décidables de X, indexée par les permutation de $\mathcal{F}$.*

  et un mot $\omega = (f_0 f_1 \dots f_k) \in \mathcal{F}^*$. On suppose de plus que tous les trajectoires du système dynamique sont finis.
- Question : *Le mot ω correspond-il à une trajectoire valide du système dynamique ? Plus précisément :*
  Existe-t-il $x_0 \in X$ and $k \in$ tel que pour tout $i \in \{0, \dots, k\}$, $x_i \in P_{f_i}$, $x_{i+1} = f_i(x_i)$ et $x_{k+1} \in P_{Id_X}$?

Remarque : Les sorties sont les point fixes , i.e. P_{Id_X}.

Caractérisation du langage des traces d'exécution : état des lieux

- tri insertion, tri selection, tri bulle $\rightsquigarrow$ systèmes de réécriture ✓
- Algorithme d'Euclide, algorithme de Gauss $\rightsquigarrow$ systèmes de réécriture ✓
- Le problème général est indécidable
- Le problème est semi-décidable.
  $L(A) = \{w \in \mathcal{F}^* : w$ une trace d'exécution de l'algorithme$\}$.

$$L(A) = \bigcup_{\pi(x^\star)=1} L_{x^\star}(A) \quad \text{où}$$

$$L_{x^\star}(A) = \{w \in \mathcal{F}^* : w \text{ une trace d'exécution qui termine sur } x^\star\}.$$

- On définit une relation d'équivalence sur les sorties :

$$x_1^\star \equiv x_2^\star \iff L_{x_1^\star}(A) = L_{x_2^\star}(A)$$

Travail en cours : Trouver une condition suffisante *naturelle* pour que le nombre de classes de $\equiv$ soit fini.

Conclusion et perspectives.

- Problème de la recherche d'un représentant
- Un algorithme réversible résout un problème de la recherche d'un représentant.
- Quelques applications
  - Borne inférieure de complexité pour algorithmes, pour le problème
  - Caractérisation du langage des traces d'un algorithme
  - Modélisation du calcul uniforme
  - Reconstruction d'un algorithme à partir des traces d'exécution (apprentissage)
  - Retrouver le problème à partir d'un algorithme (reverse engineering)