

# DS3 (éléments de réponses)

## Exercice 1

**Question 1.** Les mots  $aabb$  et  $ababb$  de longueur 4 conviennent car ce ne sont pas des carrés.

**Question 2.** Si  $m \in L^c$  alors il existe  $u \in \Sigma^*$  tel que  $m = u^2$  et donc  $|m| = 2|u|$  est paire. Par contraposition on a donc que tout mot de longueur impaire est dans  $L$ .

Soit  $m$  de longueur paire. Si  $m \in L^c$  alors  $m = u^2$  et donc pour tout  $i \in \llbracket 1, n \rrbracket$  on a  $m_i = u_i = m_{i+n}$ . Réciproquement, si pour tout  $i \in \llbracket 1, n \rrbracket$  on a  $m_i = m_{i+n}$ ,  $m$  est un carré et l'équivalence de la deuxième partie de la question est démontrée.

**Question 3.** Montrons que :

$$\mathcal{L}(\mathcal{G}, A) = \{uav, |u| = |v|\} \quad \text{et} \quad \mathcal{L}(\mathcal{G}, B) = \{ubv, |u| = |v|\}$$

On procède par double inclusion :  $\mathcal{L}(\mathcal{G}, A) \subseteq \{uav, |u| = |v|\}$  se fait par récurrence sur la longueur des dérivations et  $\{uav, |u| = |v|\} \subseteq \mathcal{L}(\mathcal{G}, A)$  par récurrence sur la longueur des mots du premier langage.

Une même démonstration établit  $\mathcal{L}(\mathcal{G}, B)$ .

**Question 4.** Si  $m$  est un mot de  $L$  de longueur impaire, il a une lettre centrale valant soit  $a$  soit  $b$  donc appartient à  $\mathcal{L}(\mathcal{G}, A) \cup \mathcal{L}(\mathcal{G}, B)$ . Comme  $S \rightarrow A \mid B$ , on en déduit que  $m \in \mathcal{L}(\mathcal{G})$ .

**Question 5.** Si  $m = m_1 \dots m_{2n}$  est de longueur paire, d'après la question 2 il existe  $i \in \llbracket 1, n \rrbracket$  tel que  $m_i \neq m_{i+n}$ . Supposons que  $m_i = a$  et  $m_{i+n} = b$  et découpons  $m$  comme suit :

$$m = m_1 \dots m_i \dots m_{2i-1} m_{2i} \dots m_{i+n} \dots m_{2n}$$

Le facteur  $m_1 \dots m_{2i-1} \in L$  est un mot de longueur impaire dont la lettre centrale vaut  $a$  donc est dans  $\mathcal{L}(\mathcal{G}, A)$ . Le facteur  $m_{2i} \dots m_{2n}$  est un mot de longueur impaire dont la lettre centrale est  $b$  donc est dans  $\mathcal{L}(\mathcal{G}, B)$ . Or,  $S \rightarrow AB$  donc  $m \in \mathcal{L}(\mathcal{G})$ . Le raisonnement est similaire lorsque  $m_i = b$  et  $m_{i+n} = a$  : dans ce cas, la première règle utilisée pour dériver  $m$  est  $S \rightarrow BA$ .

**Question 6.** Les questions 4 et 5 montrent que  $L \subseteq \mathcal{L}(\mathcal{G})$ .

Considérons  $m \in \mathcal{L}(\mathcal{G})$ .

- ♦ Si  $|m|$  est impaire alors  $m \in L$  d'après la question 2.
- ♦ Si  $|m|$  est paire alors il existe une dérivation  $S \Rightarrow^* m$ . Parmi les quatre règles possibles au début de cette dérivation, on peut en écarter deux car  $A$  et  $B$  ne peuvent engendrer que des mots de longueur impaire (donc pas  $m$ ). On en déduit que notre dérivation de  $m$  est forcément de la forme  $S \Rightarrow AB \Rightarrow^* m$  ou  $S \Rightarrow BA \Rightarrow^* m$ .

Si on est dans le premier cas alors il existe  $u \in \mathcal{L}(\mathcal{G}, A)$  et  $v \in \mathcal{L}(\mathcal{G}, B)$  tels que  $m = uv = u'au''v'bv''$  avec  $|u'| = |u''|$  et  $|v'| = |v''|$ . Mais alors il existe  $i \in \llbracket 1, n \rrbracket$  tel que  $a = m_i = m_{i+n} = b$  et la question 2 assure que  $m \in L$ . Le second cas est similaire. Finalement  $\mathcal{L}(\mathcal{G}) \subseteq L$ , inclusion qui permet de conclure.

## Exercice 2

**Question 1.** Supposons qu'il existe une fonction  $f\_A : ('a \rightarrow 'b) \rightarrow 'a \rightarrow \text{bool}$  qui résout le problème A. Construisons alors la fonction suivante :

```
let rec paradoxe () =
  if f_A paradoxe () then paradoxe ();;
```

- ♦ Si `paradoxe ()` termine, alors `f_A paradoxe ()` renvoie `true` et chaque appel à `paradoxe ()` relancera un appel récursif similaire. Donc `paradoxe ()` ne termine pas.
- ♦ Si `paradoxe ()` ne termine pas, alors `f_a paradoxe ()` renvoie `false` et la fonction s'arrête au premier appel.

Dans les deux cas, on arrive à une contradiction. La fonction `paradoxe` ne peut pas exister et la fonction `f_A` non plus. On en déduit que le problème A est **indécidable**.

Par ailleurs, l'algorithme suivant résout les instances positives de A.

```
let f_A g x =
  ignore (g x);
  true;;
```

Donc A est **semi-décidable**.

**Question 2.** Montrons que F est indécidable et n'est pas semi-décidable par une réduction des instances positives du problème complémentaire de A, noté coA. Supposons qu'il existe une fonction  $f\_F : ('a \rightarrow 'b) \rightarrow \text{bool}$  qui résout les instances positives du problème F. Considérons la fonction suivante.

```
let co_f_A g x =
  let rec h y = g x in
  f_F g;;
```

- ♦ Si  $(g, x)$  est une instance positive de coA alors  $g(x)$  ne termine pas. On en déduit qu'il n'existe aucun  $y$  tel que  $h\ y$  termine. Donc  $h$  est une instance positive de F.
- ♦ Si  $(g, x)$  est une instance négative de coA alors  $g(x)$  termine. On en déduit qu'il existe une infinité de  $y$  tels que  $h\ y$  termine. Donc  $h$  est une instance négative de F.

Finalement, la fonction **co\_f\_A** résout les instances positives de coA. Sachant que coA n'est pas semi-décidable, on en déduit que F n'est **pas semi-décidable**, et donc **indécidable**.

**Question 3.** Montrons que D est **indécidable** par une réduction de A. Supposons qu'il existe une fonction **f\_D** qui résout D. Considérons la fonction suivante.

```
let f_A g x =
  let h y = g x in
  f_D g;;
```

Alors **f\_A** résout bien le problème de l'arrêt : il y a soit une infinité, soit aucun argument  $y$  tel que  $h\ y$  termine, selon que  $(g, x)$  soit une instance positive ou négative de A. On en déduit que D est indécidable.

Par ailleurs, ce problème est **semi-décidable** en considérant l'algorithme suivant. En prenant les entrées possibles de  $g$  triées (par ordre lexicographique, par exemple) :

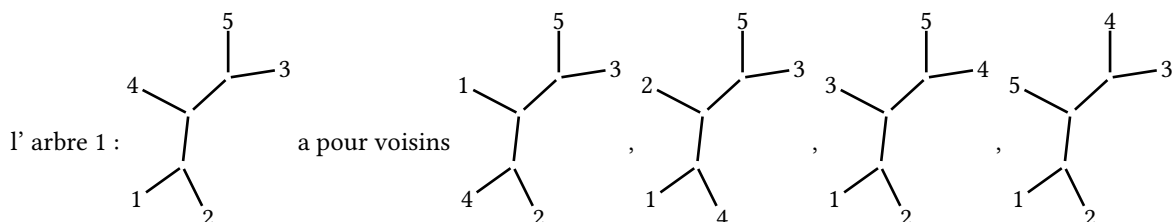
- ♦  $n = 1$
- ♦ Tant que le calcul ne s'est pas terminé pour au moins 10 arguments :
  - ♦ simuler une étape de calcul supplémentaire de  $f$  pour chacun des  $n$  premiers arguments
  - ♦  $n = n + 1$

Alors cet algorithme résout bien les instances positives de D qui est donc semi-décidable.

## Exercice 3

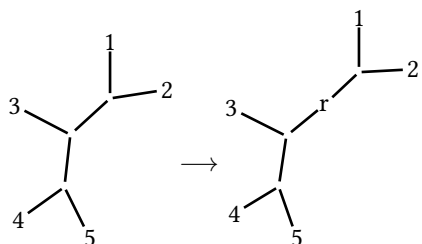
**Question 1.** Passer de l'arbre 1 à l'arbre 2 peut se faire en échangeant les sous arbre contenant 2 et 4 par un mouvement NNI sur la branche entre e et f.

Il y a 2 NNI possibles autour des 2 branches internes, soit 4 voisins en tout (à déformation près). La feuille 4 de l'arbre 1 peut être permutée avec n'importe laquelle des 4 autres :



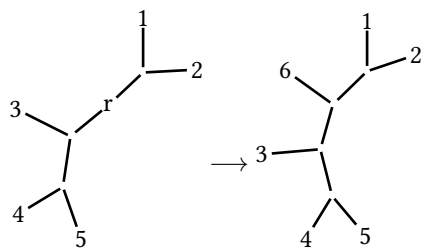
**Question 2.** Soit  $N$  et  $B$  le nombre de nœuds et de branches. Comme ce sont des arbres,  $B = N - 1$ . les feuilles étant de degré 1 et les nœuds internes de degré 3, le lemme de la poignée de main impose  $n + 3(N - n) = 2(N - 1)$ . D'où  $2n - 2$  nœuds et  $2n - 3$  branches par arbre.

**Question 3.** Pour créer un arbre raciné à  $n$  feuilles, il suffit de partir d'un arbre non raciné, de couper un branche en deux pour y insérer une racine(r) comme ci-dessous :



Comme il y a  $2n - 3$  branches par arbre non raciné, on a  $|RB(n)| = (2n - 3)|B(n)|$ .

De même on passe d'un arbre raciné à un arbre non raciné en ajoutant une feuille à la racine :



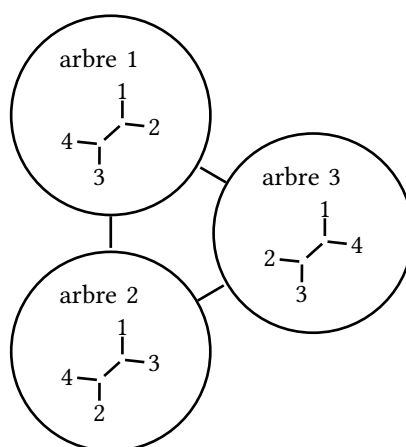
D'où :

$$\forall n \geq 3 \quad |RB(n)| = |B(n-1)|$$

En regroupant ces deux équations, il vient  $|B(n)| = (2n-5)|B(n-1)|$ . Soit  $|B(n)| = (2n-5) \times (2n-7) \times \dots \times 1$  ou encore :

$$\forall n \geq 3 \quad |B(n)| = \prod_{k=0}^{n-3} (2k+1)$$

**Question 4.**  $G_{\text{NNI}}(4)$  est un graphe à  $3 \times 1 = 3$  sommets :



**Question 5.** Il y a  $2n-3$  branches dans un arbre de  $B(n)$ , dont  $n$  sont reliées à des feuilles, soit  $n-3$  branches internes. Pour chacune de celle-ci, il y a 2 mouvements NNI possibles. Ainsi, un arbre possède  $2(n-3)$  voisins dans  $G_{\text{NNI}}(n)$ , si  $n \geq 3$ .

**Question 6.** Une chenille est un arbre pour lequel aucun nœud n'est relié à trois branches internes.

- ♦ Appelons **tête** un nœud interne dont deux branches portent des feuilles, **articulation** un nœud relié à trois branches internes. Appelons **patte** un sous-arbre connecté à cette articulation. Soit  $e$  une branche liée à une articulation. Un mouvement NNI par rapport à  $e$  rapproche nécessairement une patte d'une tête. Tant que l'articulation existe, on peut déplacer celle-ci vers la tête. Lorsque la patte atteint la tête, elle disparaît. On peut ainsi faire disparaître toutes les articulations pour obtenir une chenille.
- ♦ Reste à remarquer que l'on peut passer d'une chenille à l'autre par NNI, car :
  - ◊ à chaque chenille est associée une permutation de  $\llbracket 1 \dots n \rrbracket$  correspondant au chemin qui la définit ;
  - ◊ un mouvement NNI qui échange 2 feuilles sur une chenille correspond à une transposition ;
  - ◊ toute permutation se décompose en transpositions.

Donc  $G_{\text{NNI}}(n)$  est connexe.

**Question 7.**

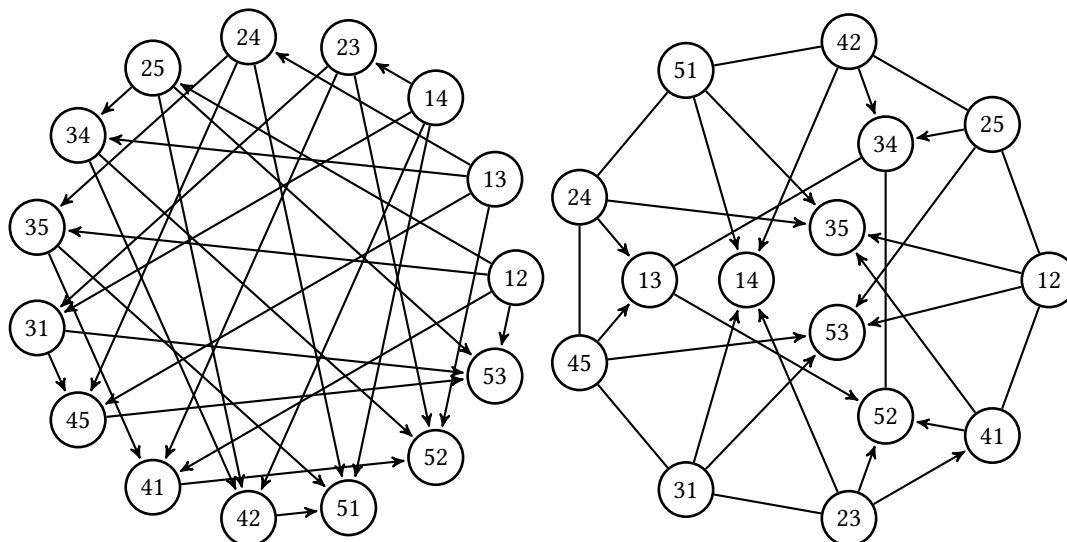
- ♦ Soit  $e$  et  $f$  les deux branches d'une transformation SPR. Tant que  $e$  et  $f$  ne sont pas adjacentes, il existe des branches internes les reliant, et donc un mouvement NNI permettant rapprocher  $e$  de  $f$ . La transformation SPR correspond à la suite des mouvements NNI qui font « glisser »  $e$  jusqu'à  $f$ .
- ♦ Un mouvement NNI est un mouvement SPR où  $e$  et  $f$  sont séparés par exactement une branche, comme en Q1.
- ♦ Il en résulte que  $G_{\text{NNI}}(n)$  est un sous-graphe de  $G_{\text{SPR}}(n)$ , ayant les mêmes sommets. Comme  $G_{\text{NNI}}(n)$  est connexe,  $G_{\text{SPR}}(n)$  est connexe.

**Question 8.** Les sommets de  $G_{NNI}(5)$  sont les arbres de  $B(5)$ , il y a donc  $5 \times 3 \times 1 = 15$  sommets d'après Q2. Et d'après Q5, chaque arbre à  $2(5 - 3) = 4$  voisins.

**Question 9.** Comme tout les arbres de  $B(5)$  se déduisent les uns des autres par permutation de feuilles, il suffit de raisonner sur un arbre particulier, l'arbre 5 dans la suite. Pour simplifier l'exposé, soit  $a$  l'étiquette centrale, associée au seul nœud n'ayant qu'une feuille (le 3 dans l'arbre 5). Tout mouvement NNI échange  $a$  et une autre feuille  $b$ , noté  $a \leftrightarrow b$ .

- ♦ Les cycles de longueur 3 correspondent aux suites de mouvements NNI qui échangent  $a$  avec les deux feuilles initialement du même côté de  $a$ . ( $3 \leftrightarrow 1 \leftrightarrow 2 \leftrightarrow 3$  et  $3 \leftrightarrow 4 \leftrightarrow 5 \leftrightarrow 3$  pour l'arbre 5). Il y a deux cycles de longueur 3 pour chaque arbre, les deux sens de parcours représentant le même cycle.
- ♦ Les cycles de longueur 5 correspondent aux suites de mouvements NNI qui donnent à  $a$  les 5 valeurs possibles. Il faut pour cela changer de côté à chaque fois. ( $3 \leftrightarrow 1 \leftrightarrow 4 \leftrightarrow 2 \leftrightarrow 5 \leftrightarrow 3$ ,  $3 \leftrightarrow 1 \leftrightarrow 5 \leftrightarrow 2 \leftrightarrow 4 \leftrightarrow 3$ ,  $3 \leftrightarrow 2 \leftrightarrow 4 \leftrightarrow 1 \leftrightarrow 5 \leftrightarrow 3$ , et  $3 \leftrightarrow 2 \leftrightarrow 5 \leftrightarrow 1 \leftrightarrow 4 \leftrightarrow 3$  pour l'arbre 5). Il y a quatre cycles de longueur 5 pour chaque arbre.
- ♦ Pour 2 échanges successifs, il y a 4 choix pour le premier, puis 3 pour le second pour ne pas retomber sur l'arbre initial, soit 12 arbres accessibles en 2 mouvements. Comme il y a 15 arbres dans  $G_{NNI}(5)$ , deux ( $15 - 12 - 1$ ) arbres sont inaccessibles en 2 mouvements, mais le sont en 3 mouvements.

**Question 10.** Pour simplifier le graphe, un arbre est dénoté par un nombre à deux chiffres  $ac$  :  $a$  représente l'étiquette centrale,  $c$  le voisin de  $(a - 2) \bmod 5 + 1$  (les arbres 1,2 et 5 sont respectivement notés 45,24 et 31). Il suffit de tracer soigneusement les 6 arêtes reliant 12 à 25 et 35, 13 à 24 et 34, 14 à 23 et 31 : les 24 autres se tracent par rotation de  $\frac{2k\pi}{5}$  sur la figure de gauche. La figure de droite représente le même graphe, les cycles y étant plus facile à observer.



On observe bien (à droite) que chaque nœud est inclus dans quatre pentagones, deux triangles, et seuls deux nœuds sont inaccessibles en deux « pas » : ( $42 \rightsquigarrow 41$  et  $42 \rightsquigarrow 45$  par exemple).

**Question 11.**

```
let rec feuilles = function
  Feuille x -> [x]
  | Noeud (a, b) -> feuilles a @ feuilles b
```

**Question 12.**

```
let rec degres (g:graphe) = match g with
  [] -> []
  | (_, l) :: q -> List.length l :: degres q
```

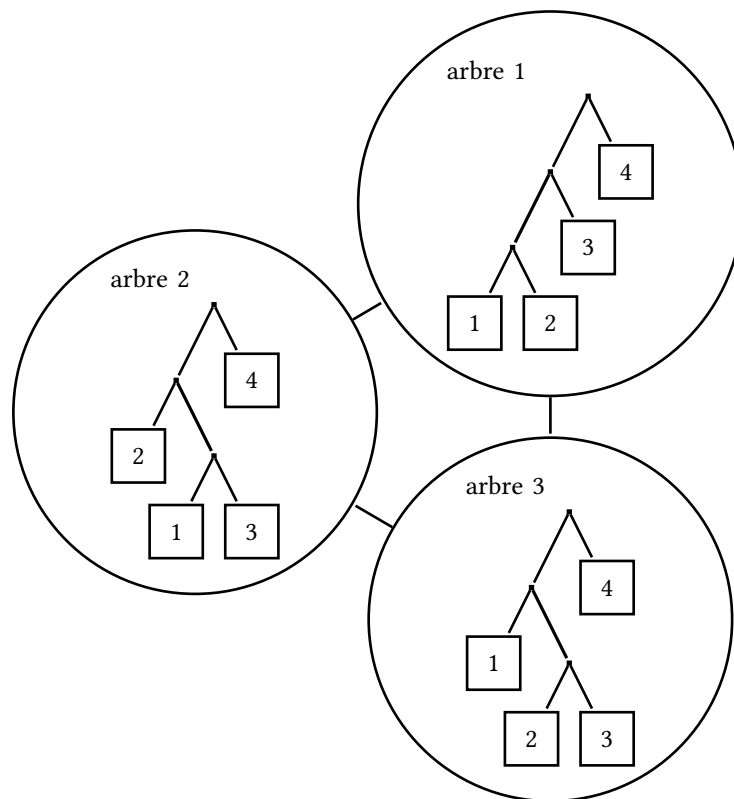
**Question 13.** `egaux` est correcte par unicité de la représentation des arbres.

```
let rec egaux a b = match (a, b) with
  Feuille x, Feuille y -> x = y
  | Noeud (ag, ad), Noeud(bg, bd) -> egaux ag bg && egaux ad bd
  | _ -> false
```

**Question 14.**

```
let rec appartient a = function
  [] -> false
  | b::q -> egaux a b || appartient a q
```

**Question 15.** Le graphe de la Q4 est repris ci-dessous en version arborescente normalisée. Chaque étiquette représente ici un sous-arbre et la branche pour le mouvement NNI est en gras.



La partie de l'arbre contenant le sous-arbre 4 est toujours considérée fixe.

Il faut distinguer les transformations NNI, selon que l'échange se fasse selon une branche père-fils gauche (NNIG) ou père-fils droit (NNID).

Pour passer de arbre 2 ou arbre 3 vers arbre 1 sur la figure précédente (NNID), il faut savoir ordonner 1 et 2 dans arbre 1 pour garantir que la plus grande étiquette soit à droite. Les transformations inverses (NNIG) sont sans ambiguïté.

La fonction auxiliaire **nmax** détermine la plus grande étiquette d'un arbre (au fond à droite). Puis **nnig**, **nnid**, **nni** réalisent respectivement les transformations associées, et les deux quand c'est possible. Enfin **voisinsNNI** explore récursivement toutes les branches internes de l'arbre :

```
let rec nmax = function Feuille x -> x | Noeud(_, a) -> nmax a
let orderNoeud (a, b) = if nmax a < nmax b then Noeud(a, b) else Noeud(b, a)
let nnig (a, b) c = [Noeud(b, Noeud(a, c)); Noeud(a, Noeud(b, c))]
let nnid a (b, c) = [Noeud(b, Noeud(a, c)); Noeud(orderNoeud(a, b), c)]

let nni abr =
  (match abr with Noeud (Noeud(a, b), c) -> nnig (a, b) c | _ -> [])
  @ (match abr with Noeud (a, Noeud(b, c)) -> nnid a (b, c) | _ -> [])

let rec voisinsNNI = function
  Noeud (g, d) ->
    List.map (fun g -> Noeud (g, d)) (nni g @ voisinsNNI g)
    @ List.map (fun d -> Noeud (g, d)) (nni d @ voisinsNNI d)
  | _ -> []
```

**Question 16.**

```
let rec chenille n =
  if n = 1 then Feuille 1 else
    Noeud (chenille (n-1), Feuille n)
```

**Question 17.**

```
let rec insere arbre (graphe:graphe) =
  match graphe with
  | [] -> [arbre, []]
  | (s,l)::q when egaux s arbre -> graphe
  | (s,l)::q -> (s,l) :: insere arbre q
```

**Question 18.**

```
let rec relie (graphe:graphe) arbre1 arbre2 =
  match graphe with
  | [] -> graphe
  | (s,l)::q when egaux s arbre1 && not (appartient arbre2 l)
    -> (arbre1, arbre2::l) :: relie q arbre1 arbre2
  | (s,l)::q when egaux s arbre2 && not (appartient arbre1 l)
    -> (arbre2, arbre1::l) :: relie q arbre1 arbre2
  | sl::q -> sl :: relie q arbre1 arbre2
```

**Question 19.** Les fonctions précédentes sont inutiles, car `voisinsNNI a` est la liste d'adjacence de `a`.

```
let rec construction fait afaire = match afaire with
| [] -> []
| arbre :: q when not (appartient arbre fait)->
  let vois = voisinsNNI arbre in
  (arbre, vois) :: construction (arbre :: fait) (q @ vois)
| _ :: q -> construction fait q

let grapheNNI n = construction [] [chenille n]
```

`construction` termine car :

- ♦ le second cas de filtrage (`when not (appartient arbre fait)`) est appelé au plus  $|B(n)|$  fois;
- ♦ le troisième cas de filtrage fait diminuer strictement  $|afaire|$ .

Et `construction` est correct car  $G_{NNI}(n)$  est connexe, donc tous les voisins sont visités et la fonction renvoie le graphe complet.

**Question 20.** Supprimer la feuille  $n - 1$  avec sa branche adjacente d'un arbre quelconque de  $S_i$ , puis la rajouter sur une branche quelconque de cet arbre ( $n$  choix possibles), c'est exactement la définition d'un mouvement SPR entre deux arbres quelconques de  $S_i$ . Deux arbres quelconques sont donc reliés. Donc  $G_{SPR}(n+1)_{|S_i}$  est complet.

**Question 21.**  $n \geq 4$  car il n'y a pas d'arêtes dans  $G_{SPR}(3)$ . Soit  $t_j$  l'arbre obtenu par SPR en utilisant deux branches nommées  $e$  et  $f$ , à partir de  $t_i$ . Soit  $t'_i$  et  $t'_j$  les arbres obtenus à partir de  $t_i$  et  $t_j$  par addition de l'étiquette  $(n+1)$  sur une branche  $g$ . Alors l'opération SPR sur les branches  $e$  et  $f$  appliquée à  $t'_i$  donne  $t'_j$ . Il y a donc au moins  $2n - 3$  arêtes (au moins 5) entre  $S_i$  et  $S_j$ .  $S_i$  est relié à  $S_j$  dans  $G_{SPR}(n+1)$  par des arêtes.

**Question 22.**

- ♦ **Initialisation** :  $G_{SPR}(4) = G_{NNI}(4)$  est clairement hamiltonien (voir Q 34).
- ♦ **Hérédité** : Soit  $n \geq 5$  et  $G_{SPR}(n-1)$  hamiltonien. Par construction,  $B(n) = \bigcup_{i=1}^n S_i$ , et les  $S_i$  sont disjoints. On commence par connecter les  $S_i$  entre eux à l'aide des arêtes exhibées à la question précédente, en formant un cycle : c'est possible car  $G_{SPR}(n-1)$  hamiltonien. puis les  $G_{SPR}(n)_{|S_i}$  étant complets, il est simple de fermer le cycle dans chaque  $S_i$ .

Donc  $G_{SPR}(n)$  est hamiltonien.

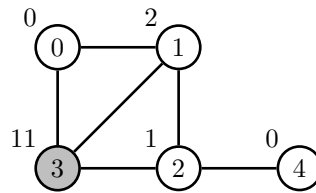
## Exercice 4

### Préliminaires

**Question 1.** On remarque que  $|\Delta_s| = 0$  et que  $|c + c'| = |c| + |c'|$ . Ainsi, si  $c \rightarrow_s c'$ , alors  $c' = c + \Delta_s$  et  $|c'| = |c|$ .

**Question 2.** Avec les notations de l'énoncé,  $c_2 = c + \Delta_s + \Delta_t = c + \Delta_t + \Delta_s = c'_2$ .

**Question 3.** On obtient :



**Question 4.** On raisonne par l'absurde. Supposons qu'il existe une suite infinie d'éboulements depuis  $c$ . Comme  $S$  est fini, l'ensemble  $I = \{s \in S \mid s \text{ s'éboule un nombre infini de fois}\}$  est non vide. Par ailleurs,  $q \notin I$  (car le puits  $q$  ne s'éboule jamais), donc  $S \setminus I$  est non vide. Par connexité de  $G$ , il existe deux sommets voisins  $s \in I$  et  $t \in S \setminus I$ . Mais alors, le sommet  $t$  accumule un nombre infini de grains au cours des éboulements, ce qui est absurde car il n'y a qu'un nombre fini de grains initialement.

**Question 5.** On montre le résultat par récurrence sur la taille maximale des avalanches.

- ♦ Si la taille maximale est 0, alors  $c$  est stable, donc  $k = \ell = 0$  et les deux multi-ensembles sont vides.
- ♦ Supposons le résultat établi pour une taille maximale d'avalanche inférieure ou égale à  $n \in \mathbb{N}$  fixé. Supposons que  $c$  est une configuration dont la taille maximale d'avalanche est  $n + 1$ . Alors, en utilisant les notations de l'énoncé,  $c_1$  et  $c'_1$  ont une taille maximale d'avalanche  $\leq n$ . Si  $s_1 = t_1$ , on a  $c_1 = c'_1$  et par hypothèse de récurrence on obtient bien le résultat voulu. Sinon,  $t_1$  est un sommet instable dans  $c_1$  et  $s_1$  est un sommet instable dans  $c'_1$ . Notons alors  $c''_2$  la configuration telle que  $c_1 \rightarrow_{t_1} c''_2$  et  $c'_1 \rightarrow_{s_1} c''_2$  (la notation est pertinente car les éboulements sont commutatifs). En considérant une avalanche :

$$c''_2 \rightarrow_{u_3} c''_3 \rightarrow_{u_4} \dots \rightarrow_{u_m} c''_m$$

on en déduit, par hypothèse de récurrence, que  $m = k$  et  $m = \ell$ , puis que  $\{t_1, u_3, \dots, u_m\} = \{s_2, \dots, s_k\}$  et  $\{s_1, u_3, \dots, u_m\} = \{t_2, \dots, t_\ell\}$ . Cela permet bien de conclure sur l'égalité des multi-ensembles voulue.

On conclut par récurrence.

**Question 6.** L'existence est donnée par la question 4. L'unicité est donnée par la question 5 et la question 2 (les éboulements sont commutatifs et s'effectuent sur les mêmes sommets dans les deux cas).

## Implémentation

**Question 7.** On écrit une fonction auxiliaire qui déplace un grain de  $s$  vers un de ses voisins. On l'applique à l'ensemble de ses voisins avec `List.iter`.

```
let eboulement g c s =
  let grain t =
    c.(t) <- c.(t) + 1;
    c.(s) <- c.(s) - 1
  in
  List.iter grain g.(s);;
```

**Question 8.** On commence par calculer un tableau des degrés, pour vérifier efficacement si un sommet est stable, puis, tant qu'il reste un sommet instable, on l'éboule. On utilise un booléen pour savoir si on a parcouru l'ensemble des sommets sans trouver de sommet instable.

```
let avalanche g q c =
  let n = Array.length g in
  let degres = Array.init n (fun s -> List.length g.(s)) in
  let fini = ref false in
  while not !fini do
    fini := true;
    for s = 0 to n - 1 do
      while s <> q && c.(s) >= degres.(s) do
        fini := false;
        eboulement g c s
      done
    done
  done;
```

**Question 9.** La fonction `eboulement` a une complexité linéaire en le degré de  $s$ , majoré par  $n$ . Ainsi, l'ensemble des éboulements s'effectuent en complexité  $\mathcal{O}(kn)$ . Le fait qu'on fasse une boucle `for` qui parcourt tous les sommets à chaque fois n'augmente pas cette complexité, car il y a au plus  $k$  passages dans la boucle `while`.



## Configurations récurrentes

**Question 10.**  $G$  possède un arbre couvrant  $T$ . Cet arbre couvrant possède un sommet  $s$  de degré 1. En supprimant ce sommet de l'arbre, le reste de l'arbre reste connexe. On en déduit que  $G[-s]$  reste connexe.

**Question 11.** Comme les configurations contiennent un nombre positif de grains sur les sommets réguliers, un sommet instable dans  $c$  reste instable dans  $c + d$ . On en déduit, par une simple récurrence, que  $c + d \rightarrow^* c' + d$ . De même,  $c' + d \rightarrow^* c' + d'$ , d'où le résultat.

**Question 12.** Une configuration nulle ne peut pas être récurrente. En effet, supposons que  $c$  est une configuration nulle récurrente et soit  $c'$  non nulle telle que  $c + c' \rightarrow^* c$ . Notons  $c'' \rightarrow_s c$  le dernier éboulement d'une avalanche de  $c + c'$ . Comme  $G[-q]$  est connexe, le sommet  $s$  (qui est régulier) a au moins un voisin  $t$  régulier. On en déduit que  $c_t \geq 1$ , ce qui est absurde, car  $c$  est nulle.

**Question 13.** La configuration  $c$  étant non nulle, il existe  $s \in S$  un sommet régulier tel que  $c_s > 0$ . Notons, pour  $m \in \llbracket 0, n-1 \rrbracket$ ,  $S_m = \{t \in S \mid d(s, t) = m\}$ , l'ensemble des sommets à distance  $m$  du sommet  $s$ . Montrons qu'en posant  $k = n^{n-1}$ , on obtient le résultat attendu. En effet :

- ♦ en éboulant  $sn^{n-2}$  fois, les sommets de  $S_1$  gagnent ce nombre de grains et il reste au moins  $n^{n-1} - (n-1) \times n^{n-2} = n^{n-2}$  grains sur  $s$  (car  $\deg(s) \leq n-1$ );
- ♦ en éboulant les sommets de  $S_1$   $n^{n-3}$  fois, les sommets de  $S_2$  gagnent au moins ce nombre de grains et il reste au moins  $n^{n-3}$  grains sur les sommets de  $S_2$ ;
- ♦ ...
- ♦ en éboulant les sommets de  $S_{n-2}$  une fois, les sommets de  $S_{n-1}$  gagnent au moins un grain et il reste au moins un grain sur les sommets de  $S_{n-2}$ .

La récurrence cachée derrière ce raisonnement permet bien de conclure.

**Question 14.** Par double implication :

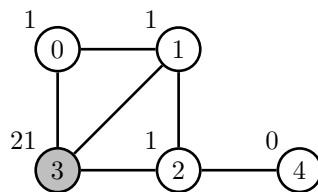
- ♦ si  $c$  est récurrente, soit  $c'$  non nulle telle que  $\text{av}(c + c') = c$ . On remarque, par la question 20 et par récurrence, que pour tout  $k \in \mathbb{N}$ ,  $\text{av}(c + kc') = c$ .  
Par la question précédente, il existe  $k$  et  $c''$  tels que  $kc' \rightarrow^* c''$  et  $c''_s > 0$  pour tout  $s$  régulier. Par un raisonnement similaire, on peut en fait choisir  $c''_s \geq \deg(s)$  (en augmentant la valeur de  $k$ ). On a alors :

$$c + kc' \rightarrow^* c + c'' = (c + c'' - \delta) + \delta \rightarrow^* c$$

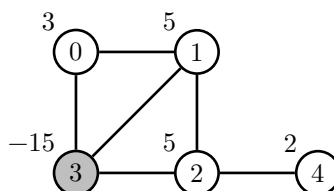
On en déduit bien le résultat attendu, car  $c + c'' - \delta$  est bien une configuration.

- ♦ réciproquement, s'il existe  $c'$  telle que  $\text{av}(c' + \delta) = c$ , alors  $\text{av}(c + (c' + \delta - c)) = c$ , et on remarque que  $\delta - c$  est bien une configuration, car  $c$  est stable (c'est le résultat d'une avalanche).

**Question 15.** On a  $\text{av}(2\delta)$  qui vaut :



On obtient alors  $\beta$  la configuration :



**Question 16.** On a  $\beta = \delta + (\delta - \text{av}(2\delta))$ . Or  $\delta - \text{av}(2\delta)$  est bien une configuration, car  $\text{bav}(2\delta)$  est stable. On en déduit que  $\beta$  est bien une configuration. Dès lors,  $\beta + \delta = 2\delta + (\delta - \text{av}(2\delta)) \rightarrow^* \text{av}(2\delta) + (\delta - \text{av}(2\delta)) = \delta$ .

**Question 17.** Le sens réciproque est direct, car  $\beta$  est une configuration non nulle (si c'était le cas, alors  $2\delta$  serait une configuration stable, ce qui est faux). Supposons que  $c$  est une configuration récurrente. Par la question 13, il existe une configuration  $c'$  telle que  $\text{av}(c' + \delta) = c$ . Dès lors,  $c' + \delta + \beta \rightarrow^* c' + \delta \rightarrow^* c$ . De plus,  $c' + \delta + \beta \rightarrow^* c + \beta \rightarrow^* c$ , par unicité de la configuration obtenue par avalanche. On en déduit bien que  $\text{av}(c + \beta) = c$ .



## Parcours décroissants

**Question 18.** Remarquons que  $\sigma^{<k} = \tau^{<k}$ , donc  $\sigma_k$  et  $\tau_k$  ne peuvent pas être tous deux une arête (car l'arête maximale vérifiant les conditions est unique). Si  $\sigma_k$  et  $\tau_k$  sont tous deux un sommet, alors  $\sigma_{k-1} = \tau_{k-1}$  est l'arête  $\{\sigma_k, \tau_k\}$ . Mais alors  $\sigma_{k-1}$  n'est incidente à aucun sommet de  $\sigma^{<k-1}$ . On conclut par l'absurde.

**Question 19.** Remarquons que  $\Phi(\sigma)$  est bien un ensemble d'arêtes, car un parcours décroissant ne peut pas contenir deux sommets consécutifs (un sommet apparaît toujours après une arête, s'il est régulier, ou au début du parcours pour le puits). Par ailleurs,  $|\Phi(\sigma)| = |S \setminus \{q\}| = n - 1$ . Enfin,  $(S, \Phi(\sigma))$  est connexe, car chaque sommet  $\sigma_k$  est adjacent à l'un des sommets de  $\sigma^{<k}$ . On en déduit que c'est un arbre, qui est également couvrant.

**Question 20.**

- ♦  $\Phi$  est injective : soient  $\sigma$  et  $\tau$  deux parcours décroissants et  $k$  l'indice minimal tel que  $\sigma_k \neq \tau_k$ . Sans perte de généralité, par la question 27,  $\sigma_k$  est un sommet et  $\tau_k$  est une arête. Alors l'arête  $\sigma_{k-1} = \tau_{k-1}$  est une arête de  $\Phi(\sigma)$ , mais pas de  $\Phi(\tau)$ , donc  $\Phi(\sigma) \neq \Phi(\tau)$ , ce qui donne l'injectivité.
- ♦  $\Phi$  est surjective : soit  $T = (S, B)$  un arbre couvrant de  $G$ . On pose  $\sigma$  le parcours arêtes-sommets défini par :
  - ◇  $\sigma_0 = q$
  - ◇ pour  $i \in [1, m + n - 1]$ , si  $\sigma_{i-1} \in B$ , on pose  $\sigma_i$  l'unique sommet incident à  $\sigma_{i-1}$  qui n'est pas dans  $\sigma^{<i}$ , sinon, on pose  $\sigma_i$  l'arête maximale selon  $<_A$  parmi les arêtes hors de  $\sigma^{<i}$  et incidentes à un sommet de  $\sigma^{<i}$ .

Cette définition est correcte et donne un parcours décroissant. Par sa définition, on a bien  $\Phi(\sigma) = T$ , d'où la surjectivité.

**Question 21.** Notons  $T = (S, B) = \Phi(\sigma)$ . Notons de plus  $rg$  la fonction qui à un sommet ou une arête associe son rang dans  $\sigma$ , c'est-à-dire vérifiant, pour tout  $i$ ,  $rg(\sigma_i) = i$ . On raisonne par double implication.

- ♦ Supposons que  $a = \{s, t\}$  est forte pour  $\sigma$ . Dans un premier temps, par définition de  $\Phi$ ,  $a \notin B$ . Posons  $\gamma = (a_1, s_1, a_2, s_2, \dots, a_k, s_k)$  le cycle obtenu en rajoutant  $a$  à  $T$ , avec  $a_1 = a$ ,  $s_1 = s$  et  $s_k = t$ . Posons  $\ell$  l'indice tel que  $s_\ell$  est de rang minimal parmi les sommets du cycle. Nécessairement,  $rg(s_\ell)$  est inférieur au rang de ses deux arêtes incidentes (sinon une telle arête aurait son autre extrémité de rang inférieur, ce qui contredirait la minimalité de  $s_\ell$ ).

Par ailleurs, pour  $1 < i \leq k$ ,  $a_i$  est une arête de l'arbre, donc doit être de rang égal à celui d'une de ses extrémités moins un. En particulier, pour  $1 < i \leq \ell$ ,  $rg(a_i) = rg(s_{i-1}) - 1$  (par récurrence descendante, car  $s_\ell$  est minimal). De même, pour  $\ell < i \leq k$ ,  $rg(a_i) = rg(s_i) - 1$ . On a alors :

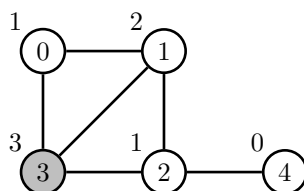
$$\begin{aligned} rg(a) &> rg(s_1) > rg(a_2) > rg(s_2) > \dots > rg(a_\ell) > rg(s_\ell) \\ rg(s_\ell) &< rg(a_{\ell+1}) < rg(s_{\ell+1}) < \dots < rg(a_k) < rg(s_k) < rg(a) \end{aligned}$$

Dès lors,  $a$  est de rang maximal dans le cycle. Posons  $r$  tel que  $a_r$  est l'arête du cycle minimale pour  $<_A$ . Supposons que  $r \neq 1$ . Sans perte de généralité, quitte à renuméroter le cycle dans l'autre sens, supposons  $\ell < r \leq k$ . Alors, après avoir choisi le sommet  $s_{r-1}$  dans le parcours  $\sigma$ , l'arête  $a_r$  n'est jamais l'arête maximale selon  $<_A$  parmi les arêtes qui ne sont pas dans  $\sigma^{<i}$  incidentes à un sommet de  $\sigma^{<i}$ , tant que toutes les autres arêtes du cycle n'ont pas été sélectionnées. On en déduit que le parcours n'est en fait pas un parcours décroissant, car on aurait dû avoir  $rg(a) < rg(a_r)$ . Par l'absurde on en déduit que  $a$  est bien minimale pour  $<_A$ , donc est active extérieurement.

- ♦ Réciproquement, supposons que  $a = \{s, t\}$  est active extérieurement. Supposons par l'absurde que  $a$  n'est pas forte, c'est-à-dire, sans perte de généralité, que  $rg(s) < rg(a) < rg(t)$ . Soit  $\sigma_j$  le sommet du cycle de rang minimal tel que  $rg(a) < rg(\sigma_j)$  ( $\sigma_j$  existe bien, car l'ensemble des tels sommets contient au moins  $t$ ). Alors  $\sigma_{j-1}$  est une arête du cycle incidente à  $\sigma_j$  (sinon par une preuve similaire à la précédente,  $\sigma_j$  serait de rang minimal dans le cycle, ce qui est impossible). On a donc  $rg(a) < j - 1$ . Mais alors, l'autre extrémité de  $\sigma_{j-1}$  est de rang inférieur au rang de  $a$  (car c'est un sommet du cycle, et par définition de  $\sigma_j$ ). On en déduit que dans le parcours  $\sigma$ , au moment où  $a$  a été ajouté au parcours, l'arête  $\sigma_{j-1}$  avait une extrémité déjà choisie et est  $>_A a$  (car  $a$  est active extérieurement). Par l'absurde, on en déduit que  $a$  est forte pour  $\sigma$ .

## Parcours décroissants et configurations récurrentes

**Question 22.** On obtient la configuration suivante :



**Question 23.** On pose  $c = \Psi(\sigma)$ . On remarque que pour un sommet régulier  $\sigma_i$ ,  $I(\sigma_i)$  contient au moins une arête dans  $\sigma^{<i}$ . On en déduit que  $c_{\sigma_i} < |I(\sigma_i)| = \deg(\sigma_i)$ , donc  $c$  est une configuration stable. Par ailleurs, posons  $c'$  la configuration telle que  $c'_q = -\deg(q)$ ,  $c'_s = 1$  si  $s \in V(q)$ , et  $c'_s = 0$  pour les autres sommets. Montrons que  $c + c' \rightarrow^* c$ . En effet, en éboulant les sommets dans l'ordre de  $\sigma$ , on obtient exactement  $c$ . Remarquons que configuration  $c + c'$  correspond à une configuration obtenue où on a éboulé le puits  $q$ . Dès lors, si on suppose qu'on a éboulé tous les sommets de  $\sigma^{<i}$  quand on atteint le sommet  $\sigma_i$ , alors les grains sur ce sommet sont :

- ♦ les  $|I(\sigma_i) \setminus \sigma^{<i}|$  grains initialement présents ;
- ♦ les  $|I(\sigma_i) \cap \sigma^{<i}|$  grains reçus par les voisins de  $\sigma_i$  déjà éboulés.

Il y a donc  $|I(\sigma_i)| = \deg(\sigma_i)$  grains sur  $\sigma_i$ , qui est bien instable et peut être éboulé.

Finalement, si on éboule chaque sommet exactement une fois, chaque sommet  $s$  perd  $\deg(s)$  grains lors de son éboulement et reçoit  $\deg(s)$  grains lors de l'éboulement de chacun de ses voisins, ce qui montre bien qu'on obtient la configuration  $c$  au final.

**Question 24.**

- ♦  $\Psi$  est injective : soient  $\sigma$  et  $\tau$  deux parcours décroissants et  $k$  l'indice minimal tel que  $\sigma_k \neq \tau_k$ . Sans perte de généralité, par la question 18,  $\sigma_k$  est un sommet et  $\tau_k$  est une arête. Supposons que  $\sigma_k = \tau_\ell$ , avec  $\ell > k$ . Alors  $I(\tau_\ell) \setminus \tau^{<\ell} \subseteq I(\sigma_k) \setminus \sigma^{<k}$ . On en déduit que  $\Psi(\sigma)(\sigma_k) > \Psi(\tau)(\sigma_k)$ , ce qui montre bien l'injectivité.
- ♦  $\Psi$  est surjective : soit  $c$  une configuration récurrente telle que  $c_q = \deg(q)$ . On définit le parcours arêtes-sommets  $\sigma$  par :
  - ◇  $\sigma_0 = q$
  - ◇ pour  $i \in \llbracket 1, m+n-1 \rrbracket$ , si  $\sigma_{i-1}$  est une arête incidente à un sommet  $s$  tel que  $s \notin \sigma^{<i}$  et  $|I(s) \setminus \sigma^{<i}| = c_s$ , on pose  $\sigma_i = s$ , sinon on pose  $\sigma_i$  l'arête maximale selon  $<_A$  parmi les arêtes hors de  $\sigma^{<i}$  et incidentes à un sommet de  $\sigma^{<i}$ .

Cette définition est correcte et donne un parcours décroissant. Par sa définition, on a bien  $\Psi(\sigma) = c$ , d'où la surjectivité.

**Question 25.** Une arête forte apparaît dans  $\sigma$  après ses deux extrémités, donc elle ajoute un grain à chacune de ses extrémités. Une arête qui n'est pas forte ajoute un grain à seulement une de ses deux extrémités (celle qui apparaît avant dans  $\sigma$ ). On en déduit que le poids de  $\Psi(\sigma)$  est bien la somme du nombre d'arêtes plus le nombre d'arêtes fortes (les arêtes fortes sont comptées deux fois).

**Question 26.** TODO