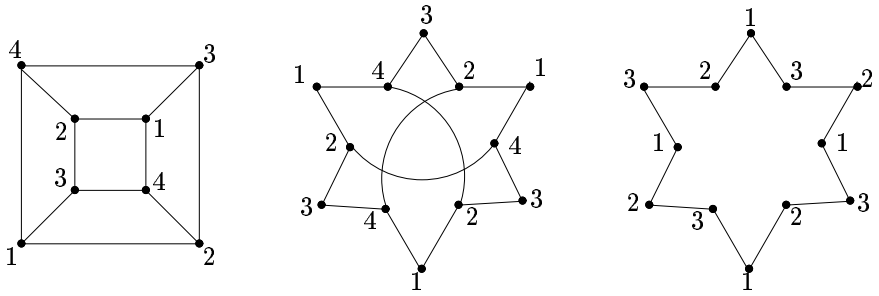
CORRIGÉ

1 Réseaux ambigus

1.1 La figure ci dessous indique des étiquetages localement bijectifs et non injectifs qui montrent que les réseaux correspondants sont ambigus.



1.2.1 Supposons que deux sommets similaires s et s' aient des voisinages non disjoints, soit $s'' \in V(s) \cap V(s')$ alors s et s' sont deux sommets de $V(s'')$ qui ont la même valeur par σ ce qui contredit le fait que σ est localement bijectif et donc injectif en restriction au voisinage de s''.

Deux arêtes non disjointes similaires sont des arêtes de la forme $\{s; s'\}$ et $\{s; s''\}$ avec s' et s'' similaires, ce qui signifie que s' et s'' sont des sommets similaires ayant des voisinages non disjoints ce qui est impossible comme on vient de le voir : cela contredit le fait que σ est injectif en restriction au voisinage de s.

1.2.2 Montrons $|\mathrm{sim}_\lambda(s)| = |\mathrm{sim}_\lambda(s')|$ lorsque s et s' sont *voisins*. Soit $\mathrm{sim}_\lambda(s) = \{s, s_1, \ldots, s_{k-1}\}$, deux sommets similaires ont des voisinages similaires et disjoints donc il existe des sommets $s'_i \in V(s_i)$ similaires à $s' \in V(s)$, ces sommets étant de plus distincts entre eux et distincts de s'. Ainsi $|\mathrm{sim}_\lambda(s')| \geq |\mathrm{sim}_\lambda(s)|$ et donc $|\mathrm{sim}_\lambda(s)| = |\mathrm{sim}_\lambda(s')|$ par symétrie. Comme le graphe est connexe, on en déduit que l'identité $|\mathrm{sim}_\lambda(s)| = |\mathrm{sim}_\lambda(s')|$ reste valide pour toute paire de sommets s et s'.

Les arêtes similaires à l'arête $a = \{s_1; s_2\}$ sont de la forme $\{s'_1; s'_2\}$ avec s_1 similaire à s'_1 et s_2 similaire à s'_2, comme deux arêtes similaires sont disjointes

on en déduit que $|\mathrm{sim}_\lambda(a)| \leq |\mathrm{sim}_\lambda(s_1)|$. Maintenant chaque sommet similaire à s_1 contient un voisin similaire à s_2 (car des sommets similaires ont des voisinages similaires) et ces sommets sont tous distincts (car les voisinages de deux sommets similaires sont disjoints); ceci nous donne $|\mathrm{sim}_\lambda(a)| \geq |\mathrm{sim}_\lambda(s_1)|$ et donc $|\mathrm{sim}_\lambda(a)| = |\mathrm{sim}_\lambda(s_1)|$.

Dire qu'une bijection locale est d'ordre 1 revient à dire qu'elle est injective, par conséquent un réseau dont toutes les bijections locales sont d'ordre 1 est non ambigu.

1.3 Comme toute paire de sommets d'un graphe complet sont voisins et que (par 1.2.1) deux sommets similaires par une bijection locale ne peuvent être voisins on en déduit qu'une bijection locale d'un réseau complet est nécessairement injective, c'est à dire qu'un tel réseau est non ambigu.

1.4 La relation de similarité définie sur l'ensemble des sommets du réseau est une relation d'équivalence, par 1.2.2 toutes ses classes ont la même cardinalité on en déduit que si un réseau a un nombre premier de sommets, soit il y a une seule classe ou chaque classe ne contient qu'un élément. Par hypothèse le réseau possède au moins deux sommets voisins et ceux ci ne sont pas similaires donc le premier cas est exclu et on en déduit que toute bijection locale est d'ordre 1 et donc qu'un tel réseau est non ambigu.

De la même façon, la relation de similarité définie sur l'ensemble des arêtes du réseau est une relation d'équivalence, par 1.2.2 toutes ses classes ont la même cardinalité on en déduit que si un réseau a un nombre premier d'arêtes, soit il y a une seule classe ou chaque classe ne contient qu'un élément. Dans le premier cas il s'agit du réseau réduit à une arête qui est non ambigu car il s'agit d'un réseau complet (1.3). Dans le second cas on en déduit que la bijection locale est d'ordre 1. Un réseau ayant un nombre premier d'arêtes est donc non ambigu.

1.5 D'après 1.2.2 l'ordre d'une bijection locale d'un réseau divise à la fois le nombre de ses sommets et le nombre de ses arêtes. Or si le réseau est un arbre ces deux nombres sont premiers entre eux (car $|A| = |S| - 1$) et donc toute bijection locale est nécessairement d'ordre 1, ainsi un arbre est non ambigu.

1.6 Si $\sigma_1 - r/s \to \sigma_2$ est une dérivation et σ_1' est un état tel que $\forall s' \in$

$S \setminus V(s) \quad \sigma'_1(s') = \sigma_1(s')$ alors $\sigma'_1 - r/s \to \sigma'_2$ est une dérivation avec

$$\sigma'_2(s') = \begin{cases} \sigma_2(s') & \text{si} \quad s' \in V(s) \\ \sigma'_1(s') & \text{sinon} \end{cases}$$

Ceci nous permet de construire par récurrence la suite de dérivations requises avec

$$\sigma_{\tau,j}(s') = \begin{cases} \sigma_{\tau(i)}(s') & \text{si} \quad s' \in V(s_{\tau(i)}) \quad \text{pour} \quad 1 \leq i \leq j \\ \sigma(s') & \text{sinon} \end{cases}$$

et donc $\sigma_{\tau,n}$ ne dépend pas de la permutation τ:

$$\sigma_{\tau,n}(s') = \begin{cases} \sigma_i(s') & \text{si} \quad s' \in V(s_i) \\ \sigma(s') & \text{sinon} \end{cases}$$

1.7 Soit σ un état λ-compatible et $\sigma - r/s \to \sigma'$ une dérivation d'un protocole. Posons $\mathrm{sim}_\lambda(s) = \{s_1, \ldots, s_n\}$ l'ensemble des sommets similaires à s, avec disons $s = s_1$. Si une règle d'un protocole est applicable en un état λ-compatible σ au sommet s, elle l'est également en cet état en chacun des sommets similaires à s et la dérivation correspondante en s_i, $\sigma - r/s_i \to \sigma_i$ est telle que $\sigma_i(s') = \begin{cases} \sigma'(\varphi_i(s')) & \text{si} \quad s' \in V(s_i) \\ \sigma(s') & \text{sinon} \end{cases}$ où $\varphi : V(s_i) \to V(s)$ est l'application qui associe à tout sommet de $V(s_i)$ le sommet de $V(s)$ qui lui est similaire. Par 1.2.1 les sommets similaires à s ont des voisinages disjoints, on en déduit par 1.6 la dérivation parallèle $\sigma - \{r/s_1, \ldots, r/s_n\} \to \sigma''$ avec $\sigma''(s') = \begin{cases} \sigma_i(s') & \text{si} \quad s' \in V(s_i) \\ \sigma(s') & \text{sinon} \end{cases}$. Montrons que σ'' est λ-compatible. Pour cela considérons s' et s'' deux sommets λ-similaires. Supposons que $s' \in V(s_i)$, comme s'' est similaire à s' on a $s'' \in V(s_j)$ pour un certain j et alors $\sigma''(s') = \sigma''(s'') = \sigma'(s''')$ où s''' est le sommet de $V(s)$ similaire à s' et à s''. Si $s' \notin \bigcup_{1 \leq i \leq n} V(s_i)$ alors il en est de même de s'' et $\sigma''(s') = \sigma(s') = \sigma(s'') = \sigma''(s'')$.

1.8 soit λ une bijection locale d'ordre $k > 1$ d'un réseau $\mathcal{R} = (S, A)$. Supposons que $\mathcal{P} = (L, R)$ soit un protocole d'énumération pour $\mathcal{R}$, c'est à dire qu'il existe un état local $\ell_0 \in L$ pour lequel toute exécution du protocole $\mathcal{P}$ sur le réseau $\mathcal{R}$ à partir de l'état σ_0, tel que $\forall s \in S \quad \sigma_0(s) = \ell_0$, termine et fournit une énumération de S, ce qui signifie que toute exécution à partir

de l'état σ_0 termine et fournit un résutat qui est une *injection*. Par conséquent tout état λ-compatible ne peut être le résultat d'une exécution du protocole $\mathcal{P}$ à partir de σ_0 (puisque lui même non injectif). Par conséquent si $(\sigma_i - r_i/s_i \to \sigma_{i+1}, 0 \leq i \leq m-1)$ est une exécution finie du protocole $\mathcal{P}$ à partir de σ_0 conduisant à un état σ_m compatible avec la bijection locale λ, il existe une règle $r \in R$ qui peut s'appliquer en un sommet $s \in S$ pour l'état σ_m. Par 1.7 cette règle peut également s'appliquer en chacun des sommets similaires à s et fournir une dérivation parallèle $\sigma_m - \{r/s' | s' \in \mathrm{sim}_\lambda\} \to \sigma'$ où σ' est un état λ-compatible. Puisque l'état initial est constant et donc λ-compatible, on peut en itérant cette construction obtenir une suite infinie de dérivation du protocole à partir de l'état σ_0, ce qui fournit la contradiction.

2 Un protocole d'énumération universel

2.1.1 Si $f : X \to I\!N$ est une fonction connexe non injective, soient x_1 et x_2 deux éléments distincts de X ayant la même valeur par f et $k = \max\{f(x) |\ x \in X\}$, la fonction $g : X \to$ donnée par $g(x) = \begin{cases} f(x) & \text{si} \quad x \neq x_1 \\ k+1 & \text{si} \quad x = x_1 \end{cases}$ est connexe et plus grande que f.

2.1.2 $g(X) \cap (I\!N \setminus \{0\}) = [f(X) \cap (I\!N \setminus \{0\})] \cup \{k+1\}$ où $k = \max\{f(y) |\ y \in Y\}$, puisque $k \in f(X)$ la fonction g est connexe.

2.1.3 $A \prec B \Leftrightarrow \forall a \in A \setminus B\ \exists b \in B \setminus A \quad a < b$. Supposons $A \prec B \prec C$ et $a \in A \setminus C$ on doit exhiber $c \in C \setminus A$ tel que $a < c$. Posons $\beta = \max(B \setminus A)$ et $\gamma = \max(C \setminus B)$.

- supposons $a \notin B$
  alors $a < \beta$ puisque $A \prec B$. Si $\beta \in C$ alors le résultat est obtenu avec $c = \beta$. Si $\beta \notin C$ alors $\beta < \gamma$ et $\gamma \notin A$ car $\gamma \in A$ contredit $A \prec B$, le résultat est alors obtenu avec $c = \gamma$.

- supposons $a \in B$
  Puisque $B \prec C$, $a < \gamma$. $\gamma \in A$ contredit $A \prec B$; donc $\gamma \in C \setminus A$ et le résultat est obtenu avec $c = \gamma$.

l'antisymétrie est évidente.

Si A et B sont deux ensembles distincts, alors au moins un des deux ensembles $A \setminus B$ et $B \setminus A$ est non vide, ces ensembles n'ont aucun éléments en commun on a donc $\max(A \setminus B) \neq \max(B \setminus A)$ et donc $A \prec B$ ou $B \prec A$.

2.2.1

$$\sigma_0 = \left(\begin{array}{l} (0,\emptyset,\emptyset), \\ (0,\emptyset,\emptyset), \\ (0,\emptyset,\emptyset) \end{array} \right)$$

$$\sigma_1 = \left(\begin{array}{l} (1,\emptyset,\{(1,\emptyset);(0,\{1\})\}), \\ (0,\{1\},\{(1,\emptyset);(0,\{1\})\}), \\ (0,\emptyset,\emptyset) \end{array} \right)$$

$$\sigma_2 = \left(\begin{array}{l} (1,\emptyset,\{(1,\emptyset);(0,\{1\})\}), \\ (0,\{1\},\{(1,\emptyset);(0,\{1\})\}), \\ (0,\emptyset,\{(1,\emptyset);(0,\{1\})\}) \end{array} \right)$$

$$\sigma_3 = \left(\begin{array}{l} (1,\emptyset,\{(1,\emptyset);(0,\{1\})\}), \\ (0,\{1;2\},\{(1,\emptyset);(0,\{1\});(0.\{1;2\});(2,\emptyset)\}), \\ (2,\emptyset,\{(1,\emptyset);(0,\{1\});(0.\{1;2\});(2,\emptyset)\}) \end{array} \right)$$

$$\sigma_4 = \left(\begin{array}{l} (1,\emptyset,\{(1,\emptyset);(0,\{1\});(0.\{1;2\});(2,\emptyset)\}), \\ (0,\{1;2\},\{(1,\emptyset);(0,\{1\});(0.\{1;2\});(2,\emptyset)\}), \\ (2,\emptyset,\{(1,\emptyset);(0,\{1\});(0.\{1;2\});(2,\emptyset)\}) \end{array} \right)$$

$$\sigma_5 = \left(\begin{array}{l} (1,\{3\},\{(1,\emptyset);(0,\{1\});(0.\{1;2\});(2,\emptyset)(1,\{3\});(3,\{1;2\});(2,\{3\})\}), \\ (3,\{1;2\},\{(1,\emptyset);(0,\{1\});(0.\{1;2\});(2,\emptyset)(1,\{3\});(3,\{1;2\});(2,\{3\})\}), \\ (2,\{3\},\{(1,\emptyset);(0,\{1\});(0.\{1;2\});(2,\emptyset)(1,\{3\});(3,\{1;2\});(2,\{3\})\}) \end{array} \right)$$

L'exécution est terminée car les trois boîtes à lettres sont égales et en chaque sommet le registre est plus grand que les messages ayant le même identificateur que le sommet.

2.3.1 Si $n_i(s) \neq n_{i+1}(s)$ c'est que $n_{i+1}(s) = \max(N_i(s)) + 1$ et comme $n_i(s) \in N_i(s))$ on déduit $n_i(s) < n_{i+1}(s)$. Si $R_i(s) \neq R_{i+1}(s)$ cela signifie que $R_{i+1}(s) = (R_i(s) \setminus \{k\}) \cup \{k'\}$ avec $0 \leq k \leq k'$ et donc $R_i(s) \preceq R_{i+1}(s)$. Les boîtes à lettres quant à elles ne font que croître.

2.3.2 On fait la preuve par récurrence sur i. Pour $i = 0$ il n'y a rien à

prouver puisque $n_0(s) = 0$ pour tout $s \in S$. Supposons cette propriété pour tout k tel que $0 \leq k < i$. Supposons $n_i(s) = n > 0$, si $n_i(s) = n_{i+1}(s)$ alors la propriété est prouvée par 2.3.1 en prenant $s' = s$. Si $n_i(s) \neq n_{i+1}(s)$, alors il existe $k \leq i$ et $s_k \in S$ tels que $n_k(s_k) = n$ et $R_i(s) \prec R_k(s_k)$. Si $k = i$ alors $s \neq s_k$ et donc $n_i(s_k) = n_{i+1}(s_k)$ et $R_i(s) \preceq R_{i+1}(s_k)$ ce qui prouve le résultat. Si $k < i$, par hypothèse de récurrence appliquée $i - k$ fois on déduit un s' tel que $n_i(s') = n$ et $R_i(s) \preceq R_i(s')$, on déduit comme précédemment $s \neq s'$ et $n_{i+1}(s') = n$ et $R_i(s) \preceq R_{i+1}(s')$.

2.3.3 On fait la preuve par récurrence sur i. Le cas $i = 0$ est trivialement vérifié. Soit $s' \in V(s)$ tel que $n_{i+1}(s') = n > 0$. Si $n = n_i(s')$, par hypothèse de récurrence on déduit $n \in N_i(s)$ et donc $n \in N_{i+1}(s)$. Si $n \neq n_i(s')$ alors par application du protocole en s', $(n, R_{i+1}(s')) \in B_{i+1}(s')$ et donc $n \in N_{i+1}(s)$.

2.3.4 On fait la preuve par récurrence sur i. Le cas $i = 0$ est trivialement vérifié. Supposons par l'absurde qu'il existe $s', s'' \in V(s)$ distints tela que $n_{i+1}(s') = n_{i+1}(s'') > 0$. Si $n_i(s') = n_{i+1}(s')$ et $n_i(s'') = n_{i+1}(s'')$ l'hypothèse de récurrence nous fournit la contradiction. Si par exemple $n_i(s') < n_{i+1}(s')$ (et donc $n_i(s'') = n_{i+1}(s'')$) alors $n_{i+1}(s') > \max(N_i(s'))$. Puisque $s'' \in V(s)$, par 2.3.3 il vient $n_i(s'') = n_{i+1}(s'') \in N_i(s)$, par ls condition du renommage $N_i(s) = N_i(s')$ d'où $n_{i+1}(s') > n_{i+1}(s'')$.

2.3.5 On fait la preuve par récurrence sur i. Le cas $i = 0$ est trivialement vérifié. Supposons $R_i(s) = n_i(V(s) \setminus \{s\}) \setminus \{0\}$. Si aucun des sommets voisins de s n'a changé son identificateur en σ_i, alors $R_{i+1}(s) = R_i(s) = n_i(V(s) \setminus \{s\}) \setminus \{0\} = n_{i+1}(V(s) \setminus \{s\}) \setminus \{0\}$. Si au contraire il existe un sommet s' voisin de s pour lequel $n_i(s') \neq n_{i+1}(s')$, alors $R_{1+1} = (R_i(s) \setminus \{n_i(s')\}) \cup \{n_{i+1}(s')\}$ comme par ailleurs $n_{i+1}(V(s) \setminus \{s\}) = n_i(V(s) \setminus \{s, s'\}) \cup \{n_{i+1}(s')\}$ la conclusion $R_{i+1}(s) = n_{i+1}(V(s) \setminus \{s\}) \setminus \{0\}$ suit lorsque $n_i(s') = 0$ mais aussi lorsque $n_i(s') > 0$ en utilisant 2.3.4.

2.3.6 On fait la preuve par récurrence sur i. Le cas $i = 0$ est trivialement vérifié. Si $n_{i+1} = n_i$ alors elle est connexe par hypothèse de récurrence. Sinon cela signifie qu'il existe $s \in S \quad n_i(s) < n_{i+1}(s)$ et alors ou bien $n_i(s) = 0$ ou bien $\exists j < i \quad \exists s_i \in S \quad n_i(s) = n_j(s_1) > 0$ et $R_i(s) \prec R_j(s')$, par application successive de 2.3.2 on déduit $\exists s_2 \in S \quad n_i(s_2) = n_i(s)$ et $R_i(s) \prec R_i(s_2)$ et

donc en particulier $s \neq s_2$. Par conséquent on a

$$n_{i+1}(s') = \begin{cases} n_i(s') & \text{si} \quad s' \neq s \\ \max\{n_i(s'')|\ s'' \in V(s)\} + 1 & \text{si} \quad s' = s \end{cases}$$

et $n_i(s) = 0$ ou $n_i(s) = n_i(s_2)$ pour un certain $s_2 \neq s$; on en déduit par 2.1.2 que n_{i+1} est connexe.

2.3.7 On ordonne les états locaux du protocole par

$$(n, R, B) \leq (n', R', B') \quad \Leftrightarrow \quad n \leq n' \wedge R \preceq R' \wedge B \subseteq B'$$

l'ordre sur les états qui s'en déduit est défini point à point:

$$\sigma \leq \sigma' \quad \Leftrightarrow \quad \forall s \in S \quad \sigma(s) \leq \sigma'(s)$$

. Soit $\sigma_0 - s_1 \rightarrow \ \sigma_1 - s_2 \rightarrow \sigma_2 \ldots \ \sigma_n - s_{n+1} \rightarrow \sigma_{n+1} \ \ldots$ une exécution du protocole. Par 2.3.1, $\sigma_i \leq \sigma_{i+1}$ pour tout $i \geq 0$. De plus $\sigma_i \neq \sigma_{i+1}$ car il existe au moins un sommet s pour lequel $\sigma_i(s) < \sigma_{i+1}(s)$ ou $B_i(s) \overset{\subset}{\neq} B_{i+1}(s)$. On en déduit que la suite des états

$$\sigma_0 < \sigma_1 < \ldots < \sigma_n < \ldots$$

est strictement croissante. Par 2.3.6 $\sigma_i(s) \leq |S|$, par ailleurs tout $B_i(s)$ est contenu dans l'ensemble $\{(k, W)|\ 0 \leq k \leq |S| \wedge W \subseteq \{1, 2, \ldots, |S|\}\}$, il s'en suit que l'exécution est nécessairement finie.

2.3.8 Si σ_m est le résultat d'une exécution du protocole à partir de l'état σ_0, alors *(i)* toutes les boîtes à lettres sont égales et *(ii)* tous les identificateurs sont non nuls en σ_m sinon une dérivation du protocole serait possible dans cet état. *(iii)* découle de 2.3.5. *(iv)* supposons $n_m(s') = n_m(s'')$ et $n_m(V(s') \setminus \{s'\}) \neq n_m(V(s'') \setminus \{s''\})$ par exemple $n_m(V(s') \setminus \{s'\}) \prec n_m(V(s'') \setminus \{s''\})$, i.e. par *(iii)* $R_m(s') \prec R_m(s'')$ par ailleurs $(n_m(s''), R_m(s'')) \in B_m(s')$ donc le protocole s'applique en s' ce qui nous donne la contradiction. *(v)* suit de 2.3.4 et du point *(iv)* ci dessus.

2.3.9 Soit $\mathcal{R} = (S, A)$ un réseau et n_m le résultat d'une exécution du protocole sur $\mathcal{R}$. L'application n_m est connexe par 2.3.6 et localement bijective par 2.3.8*(v)*. Par conséquent, ou bien n_m est une énumération (i.e. par 2.1.2 une application connexe sur S maximale) ou bien il s'agit d'un étiquetage localement bijectif d'ordre $k > 1$ montrant l'ambiguïté du réseau.