Université de Montpellier - Master 2
Module **Contraintes**

Feuille TD 3 - 13/11/2023

Exercice 1

Soit $\Delta = \{c_0, c_1, c_2, c_3, \mu_0, \mu_1\}$ un langage de contraintes sur le domaine $D = \{0, 1\}$ dont les fonctions sont définies comme suit :

$$\begin{aligned}
c_0(x, y, z) &= x \vee y \vee z = D^3 \backslash \{(0, 0, 0)\} \\
c_1(x, y, z) &= \overline{x} \vee y \vee z = D^3 \backslash \{(1, 0, 0)\} \\
c_2(x, y, z) &= \overline{x} \vee \overline{y} \vee z = D^3 \backslash \{(1, 1, 0)\} \\
c_3(x, y, z) &= \overline{x} \vee \overline{y} \vee \overline{z} = D^3 \backslash \{(1, 1, 1)\} \\
\mu_0(x) &= \{(0)\} \\
\mu_1(x) &= \{(1)\}
\end{aligned}$$

Question 1. Démontrez que CSP(Δ) est NP-complet.

Correction. Le langage Δ n'admet aucune opération parmi $\{f_0, f_1, f_\vee, f_\wedge, \text{majority}, \text{minority}\}$ comme polymorphisme. En effet,

(f_0) $\tau_1 = (1, 1, 1) \in c_0$, $f_0(\tau_1) = (0, 0, 0) \notin c_0 \Rightarrow f_0 \notin \text{Pol}(\Delta)$
(f_1) $\tau_1 = (0, 0, 0) \in c_3$, $f_1(\tau_1) = (1, 1, 1) \notin c_3 \Rightarrow f_1 \notin \text{Pol}(\Delta)$
($f_\vee$) $\tau_1 = (1, 0, 1)$, $\tau_2 = (0, 1, 0) \in c_3$: $f_\vee(\tau_1, \tau_2) = (1, 1, 1) \notin c_3 \Rightarrow f_\vee \notin \text{Pol}(\Delta)$
($f_\wedge$) $\tau_1 = (1, 0, 0)$, $\tau_2 = (0, 1, 0) \in c_0$: $f_\wedge(\tau_1, \tau_2) = (0, 0, 0) \notin c_0 \Rightarrow f_\wedge \notin \text{Pol}(\Delta)$
(majority) $\tau_1 = (1, 0, 0)$, $\tau_2 = (0, 1, 0)$, $\tau_3 = (0, 0, 1) \in c_0$: $\text{majority}(\tau_1, \tau_2, \tau_3) = (0, 0, 0) \notin c_0$
$\Rightarrow \text{majority} \notin \text{Pol}(\Delta)$
(minority) $\tau_1 = (0, 1, 1)$, $\tau_2 = (1, 0, 1)$, $\tau_3 = (1, 1, 0) \in c_0$: $\text{minority}(\tau_1, \tau_2, \tau_3) = (0, 0, 0) \notin c_0$
$\Rightarrow \text{minority} \notin \text{Pol}(\Delta)$

Par le théorème de Schaefer, CSP(Δ) est donc NP-complet.

On suppose que P $\neq$ NP. On dit qu'un langage $\Gamma_1 \subseteq \Delta$ est Δ-*maximal* si CSP(Γ_1) est polynomial mais que CSP(Γ_2) est NP-complet pour tout Γ_2 tel que $\Gamma_1 \subset \Gamma_2 \subseteq \Delta$.

Question 2. Justifiez qu'il existe au plus six langages Δ-maximaux.

Correction. Supposons qu'il existe 7 langages Δ-maximaux distincts. Par le théorème de Schaefer, chacun de ces langages admet au moins une opération parmi $\{f_0, f_1, f_\vee, f_\wedge, \text{majority}, \text{minority}\}$ comme polymorphisme. Il n'y a que 6 opérations possibles, donc il existe deux langages Δ-maximaux distincts Γ', Γ'' qui admettent la même opération.

Par le théorème de Schaefer, $\mathrm{CSP}(\Gamma' \cup \Gamma'')$ est polynomial. De plus, on a $\Gamma' \subset \Gamma' \cup \Gamma'' \subseteq \Delta$, ce qui contredit le fait que Γ' est Δ-maximal.

Question 3. Déterminez tous les langages Δ-maximaux.

Correction. Au vu de la réponse à la question précédente, les langages Δ-maximaux sont exactement les langages maximaux (par rapport à l'inclusion) qui admettent pour polymorphisme au moins une des six opérations du théorème de Schaefer. On commence donc par faire l'inventaire des polymorphismes de chaque fonction de Δ :

	c_0	c_1	c_2	c_3	μ_0	μ_1
f_0	**NON**	**OUI**	**OUI**	**OUI**	**OUI**	**NON**
f_1	**OUI**	**OUI**	**OUI**	**NON**	**NON**	**OUI**
$f_\vee$	**OUI**	**OUI**	**NON**	**NON**	**OUI**	**OUI**
$f_\wedge$	**NON**	**NON**	**OUI**	**OUI**	**OUI**	**OUI**
majority	**NON**	**NON**	**NON**	**NON**	**OUI**	**OUI**
minority	**NON**	**NON**	**NON**	**NON**	**OUI**	**OUI**

Voici une justification brève pour chaque entrée du tableau :

$\mathbf{c_0}$

(f_0, c_0) $\tau_1 = (1,1,1) \in c_0,\ f_0(\tau_1) = (0,0,0) \notin c_0 \Rightarrow f_0 \notin \mathrm{Pol}(\{c_0\})$
(f_1, c_0) $\forall \tau_1 \in c_0,\ f_1(\tau_1) = (1,1,1) \in c_0 \Rightarrow f_1 \in \mathrm{Pol}(\{c_0\})$
$(f_\vee, c_0)$ $\forall \tau_1, \tau_2 \in c_0,\ f_\vee(\tau_1, \tau_2) \neq (0,0,0)$ car $(0,0,0) \notin \{\tau_1, \tau_2\}$
donc $f_\vee(\tau_1, \tau_2) \in c_0$ et finalement $f_\vee \in \mathrm{Pol}(\{c_0\})$
$(f_\wedge, c_0)$ $\tau_1 = (1,0,0),\ \tau_2 = (0,1,0) : f_\wedge(\tau_1, \tau_2) = (0,0,0) \notin c_0 \Rightarrow f_\wedge \notin \mathrm{Pol}(\{c_0\})$
$(\mathrm{majority}, c_0)$ $\mathrm{majority}((1,0,0),(0,1,0),(0,0,1)) = (0,0,0) \notin c_0 \Rightarrow \mathrm{majority} \notin \mathrm{Pol}(\{c_0\})$
$(\mathrm{minority}, c_0)$ $\mathrm{minority}((0,1,1),(1,0,1),(1,1,0)) = (0,0,0) \notin c_0 \Rightarrow \mathrm{minority} \notin \mathrm{Pol}(\{c_0\})$

$\mathbf{c_1}$

(f_0, c_1) $\forall \tau_1 \in c_1,\ f_0(\tau_1) = (0,0,0) \in c_1 \Rightarrow f_0 \in \mathrm{Pol}(\{c_1\})$
(f_1, c_1) $\forall \tau_1 \in c_1,\ f_1(\tau_1) = (1,1,1) \in c_1 \Rightarrow f_1 \in \mathrm{Pol}(\{c_1\})$
$(f_\vee, c_1)$ $\forall \tau_1, \tau_2 \in c_1,\ f_\vee(\tau_1, \tau_2) \neq (1,0,0)$ car $(1,0,0) \notin \{\tau_1, \tau_2\}$
donc $f_\vee(\tau_1, \tau_2) \in c_1$ et finalement $f_\vee \in \mathrm{Pol}(\{c_1\})$
$(f_\wedge, c_1)$ $\tau_1 = (1,0,1),\ \tau_2 = (1,1,0) : f_\wedge(\tau_1, \tau_2) = (1,0,0) \notin c_1 \Rightarrow f_\wedge \notin \mathrm{Pol}(\{c_1\})$
$(\mathrm{majority}, c_1)$ $\mathrm{majority}((1,0,1),(1,1,0),(0,0,0)) = (1,0,0) \notin c_1 \Rightarrow \mathrm{majority} \notin \mathrm{Pol}(\{c_1\})$
$(\mathrm{minority}, c_1)$ $\mathrm{minority}((1,1,0),(0,0,0),(0,1,0)) = (1,0,0) \notin c_1 \Rightarrow \mathrm{minority} \notin \mathrm{Pol}(\{c_1\})$

$\mathbf{c_2}$

(f_0, c_2) $\forall \tau_1 \in c_2,\ f_0(\tau_1) = (0,0,0) \in c_2 \Rightarrow f_0 \in \mathrm{Pol}(\{c_2\})$
(f_1, c_2) $\forall \tau_1 \in c_2,\ f_1(\tau_1) = (1,1,1) \in c_2 \Rightarrow f_1 \in \mathrm{Pol}(\{c_2\})$
$(f_\vee, c_2)$ $\tau_1 = (1,0,0),\ \tau_2 = (0,1,0) : f_\vee(\tau_1, \tau_2) = (1,1,0) \notin c_2 \Rightarrow f_\vee \notin \mathrm{Pol}(\{c_2\})$
$(f_\wedge, c_2)$ $\forall \tau_1, \tau_2 \in c_2,\ f_\wedge(\tau_1, \tau_2) \neq (1,1,0)$ car $(1,1,0) \notin \{\tau_1, \tau_2\}$
donc $f_\wedge(\tau_1, \tau_2) \in c_2$ et finalement $f_\wedge \in \mathrm{Pol}(\{c_2\})$
$(\mathrm{majority}, c_2)$ $\mathrm{majority}((1,0,0),(1,1,1),(0,1,0)) = (1,1,0) \notin c_2 \Rightarrow \mathrm{majority} \notin \mathrm{Pol}(\{c_2\})$
$(\mathrm{minority}, c_2)$ $\mathrm{minority}((1,0,0),(0,1,0),(0,0,0)) = (1,1,0) \notin c_2 \Rightarrow \mathrm{minority} \notin \mathrm{Pol}(\{c_2\})$

$\mathbf{c_3}$

(f_0, c_3) $\forall \tau_1 \in c_3,\ f_0(\tau_1) = (0,0,0) \in c_3 \Rightarrow f_0 \in \mathrm{Pol}(\{c_3\})$
(f_1, c_3) $\tau_1 = (0,0,0) \in c_3,\ f_1(\tau_1) = (1,1,1) \notin c_3 \Rightarrow f_1 \notin \mathrm{Pol}(\{c_3\})$
$(f_\vee, c_3)$ $\tau_1 = (1,0,1),\ \tau_2 = (0,1,0) : f_\vee(\tau_1, \tau_2) = (1,1,1) \notin c_3 \Rightarrow f_\vee \notin \mathrm{Pol}(\{c_3\})$
$(f_\wedge, c_3)$ $\forall \tau_1, \tau_2 \in c_3,\ f_\wedge(\tau_1, \tau_2) \neq (1,1,1)$ car $(1,1,1) \notin \{\tau_1, \tau_2\}$
donc $f_\wedge(\tau_1, \tau_2) \in c_3$ et finalement $f_\wedge \in \mathrm{Pol}(\{c_3\})$
$(\mathrm{majority}, c_3)$ $\mathrm{majority}((1,0,1),(1,1,0),(0,1,1)) = (1,1,1) \notin c_3 \Rightarrow \mathrm{majority} \notin \mathrm{Pol}(\{c_3\})$
$(\mathrm{minority}, c_3)$ $\mathrm{minority}((1,0,0),(0,1,0),(0,0,1)) = (1,1,1) \notin c_3 \Rightarrow \mathrm{minority} \notin \mathrm{Pol}(\{c_3\})$

$\boldsymbol{\mu_0, \mu_1}$

$f_\vee$, $f_\wedge$ (resp. majority et minority) sont des polymorphismes de μ_0 et μ_1 car ce sont des opérations idempotentes : si on leur donne deux fois (resp. trois fois) le même tuple τ en entrée, elles renvoient τ en sortie. Les cas f_0 et f_1 sont immédiats.

En conclusion, d'après le tableau on a :

$$\Gamma \text{ est } \Delta\text{-maximal} \iff \begin{cases} \Gamma = \{c_1, c_2, c_3, \mu_0\}, ou \\ \Gamma = \{c_0, c_1, c_2, \mu_1\}, ou \\ \Gamma = \{c_0, c_1, \mu_0, \mu_1\}, ou \\ \Gamma = \{c_2, c_3, \mu_0, \mu_1\} \end{cases}$$

Exercice 2

Pour tout entier naturel $n > 0$ on définit $X_n = \{x_i \mid 1 \leq i \leq n\}$, $Y_n = \{y_i \mid 1 \leq i \leq n\}$, $\mathcal{Y}_n^{+1} = \{Y_n \cup \{x\} \mid x \in X_n\}$ et $\mathcal{X}_n^{+1} = \{X_n \cup \{y\} \mid y \in Y_n\}$. On considère la famille $\mathcal{H}$ des hypergraphes dont l'ensemble des sommets est de la forme $X_n \cup Y_n$ (pour un n qui peut varier d'un hypergraphe à un autre) et dont les arêtes sont des éléments de $\mathcal{Y}_n^{+1} \cup \mathcal{X}_n^{+1}$.

Question 1. La treewidth des hypergraphes de $\mathcal{H}$ est-elle bornée par une constante ? Justifiez.

Correction. Si un hypergraphe H a une arête qui contient au moins k sommets, alors sa treewidth est au moins $k-1$ car dans toute décomposition arborescente de H il existe un sac qui contient entièrement cette arête. Pour tout $k > 0$ la famille $\mathcal{H}$ contient des hypergraphes dont les arêtes contiennent plus de k sommets, donc la treewidth de $\mathcal{H}$ n'est pas bornée.

Question 2. Démontrez que tout $H \in \mathcal{H}$ a une hypertreewidth d'au plus 2.

Correction. Soit $H \in \mathcal{H}$ un hypergraphe dont l'ensemble des sommets est $X_n \cup Y_n$. Si H ne contient aucune arête, son hypertreewidth est 0.

Si H contient deux arêtes e, f telles que $e \in \mathcal{Y}_n^{+1}$ et $f \in \mathcal{X}_n^{+1}$, alors e et f couvrent H : la décomposition arborescente dont le seul sac est $X_n \cup Y_n$ a une c-width de 2.

Supposons au contraire que H ne contient aucune arête de $\mathcal{Y}_n^{+1}$. Alors, H a une décomposition arborescente composée d'une racine dont le sac est X_n, et de feuilles dont les sacs coincident avec les arêtes de H. La c-width de cette décomposition est 1 car chaque sac est couvert par une arête.

Le dernier cas (aucune arête de $\mathcal{X}_n^{+1}$) est symétrique.

Question 3. Soient $n > 0$ et H_n l'hypergraphe dont les sommets sont $X_n \cup Y_n$ et les arêtes sont $\mathcal{Y}_n^{+1} \cup \mathcal{X}_n^{+1}$. Montrer que l'hypertreewidth fractionnaire de H_n est strictement inférieure à 2.

Correction. Si l'on donne un poids $w(e) = 1/(n+1)$ à chaque arête e de H_n, pour chaque sommet v on a $\sum_{e:v\in e} w(e) = 1$. C'est donc une couverture fractionnaire de H_n. H_n contient $2n$ arêtes, donc le poids total de cette couverture est $2n \times (1/(n+1)) < 2$: la décomposition arborescente dont le seul sac est $X_n \cup Y_n$ a une fc-width strictement inférieure à 2.

Question 4. La famille $\mathcal{H}$ a la propriété d'être *hypertreewidth-monotone*, c'est-à-dire que retirer une arête d'un hypergraphe de H ne peut pas augmenter son hypertreewidth. Est-ce vrai en général, pour tous les hypergraphes ? Est-ce vrai en général pour la treewidth ?

Correction. C'est faux en général. Si c'était le cas, tous les hypergraphes auraient une hypertreewidth 1 : ajouter une arête complète (qui couvre tous les sommets) à un hypergraphe produit toujours un hypergraphe d'hypertreewidth 1. C'est en revanche vrai pour la treewidth : si l'on obtient H' en supprimant une arête d'un hypergraphe H, toute décomposition arborescente de H reste valable pour H' et sa largeur (en termes de nombre de sommets par sac) ne change pas.

Question 5. Dans l'autre sens, il est évidemment possible qu'ajouter une arête à un hypergraphe augmente son hypertreewidth. Montrez que dans ce cas, son hypertreewidth augmente de 1 au maximum.

Correction. On prend une décomposition arborescente de H de c-width minimum, et on ajoute les sommets de la nouvelle arête à tous les sacs. C'est une décomposition arborescente du nouvel hypergraphe, et sa c-width n'a augmenté que d'au plus 1.

Exercice 3

Pour tout entier naturel $k > 0$ et tout ensemble $S \subseteq \{0, \ldots, k\}$, on définit la fonction booléenne $c_S^k = \{\tau \in \{0,1\}^k \mid \sum_{i=1}^k \tau[i] \in S\}$. Par exemple, on a

$$c^3_{\{1,3\}}(x,y,z) = \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \qquad c^3_{\{0,1\}}(x,y,z) = \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \qquad c^2_{\{1\}}(x,y) = \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

Question 1. Soient $k > 0$ et $S \subseteq \{0, \ldots, k\}$ tel que $S \neq \emptyset$ et $0 \notin S$. Démontrez que l'opération $\vee$ ("ou" logique) est un polymorphisme de c_S^k si et seulement si il existe $j \in \{1, \ldots, k\}$ tel que $S = \{j, \ldots, k\}$.

Correction. Première implication : supposons $S = \{j, \ldots, k\}$. Soient $\tau_1, \tau_2 \in c_S^k$ et $\tau_3 = \tau_1 \vee \tau_2$. Alors,

$$\sum_{i=1}^k \tau_3[i] = \sum_{i=1}^k (\tau_1[i] \vee \tau_2[i]) \geq \sum_{i=1}^k \tau_1[i] \geq j$$

et donc $\tau_3 \in c_S^k$: $\vee$ est un polymorphisme de c_S^k.

Réciproque : supposons que $\vee$ est un polymorphisme de c_S^k. L'ensemble S est non vide, donc $j = \min(S)$ existe. De plus, $0 \notin S$ donc $j \in \{1, \ldots, k\}$. On va prouver la proposition suivante par récurrence sur $n \geq j$:

$$P(n) : \{j, \ldots, n\} \subseteq S$$

$P(j)$ est vraie. Supposons maintenant que $P(n)$ est vraie pour un certain n tel que $j \leq n < k$. Soit un tuple $\tau_1 \in c_k^S$ tel que $\sum_{i=1}^k \tau_1[i] = n$, et deux indices p_0, p_1 tels que $\tau_1[p_0] = 0$ et $\tau_1[p_1] = 1$. (Ces deux indices existent car $n \geq j > 0$ et $n < k$.) On construit un tuple τ_2 comme suit :

$$\tau_2[i] = \begin{cases} 1 & \text{si } i = p_0 \\ 0 & \text{si } i = p_1 \\ \tau_1[i] & \text{sinon} \end{cases}$$

Par construction, on a $\tau_2 \in c_k^S$ car $\sum_{i=1}^k \tau_2[i] = \sum_{i=1}^k \tau_1[i] = n \in S$. De plus, par hypothèse $\vee$ est polymorphisme de c_k^S et $\sum_{i=1}^k (\tau_1 \vee \tau_2)[i] = n + 1$, ce qui implique que $n + 1 \in S$ et donc $P(n + 1)$ est vraie. Finalement, par récurrence on conclut que $P(k)$ est vraie, cqfd.

Question 2. Soient $k > 0$ et $S \subseteq \{0, \ldots, k\}$. Montrez que l'opération $\vee$ est un polymorphisme de c_S^k si et seulement si c'est un polymorphisme de $c^k_{S \setminus \{0\}}$. Déduisez-en une caractérisation précise des fonctions c_S^k qui admettent le polymorphisme $\vee$, en fonction de l'ensemble S.

Correction. On a $(0,\dots,0) \vee \tau_1 = \tau_1$ pour tout tuple τ_1, donc

$$\vee \text{ polymorphisme de } c^k_{S\setminus\{0\}} \Rightarrow \vee \text{ polymorphisme de } c^k_S$$

Réciproquement, si $\vee$ n'est pas polymorphisme de $c^k_{S\setminus\{0\}}$ alors il existe $\tau_1, \tau_2 \in c^k_{S\setminus\{0\}}$ tels que $\tau_3 = \tau_1 \vee \tau_2 \notin c^k_{S\setminus\{0\}}$. Puisque $0 \notin S$, on a $(0,\dots,0) \notin \{\tau_1, \tau_2\}$ et donc $\tau_3 \neq (0,\dots,0)$. Cela implique $\tau_3 \notin c^k_S$, et finalement $\vee$ n'est pas polymorphisme de c^S_k. En combinant ce résultat avec la question précédente on obtient :

$$\vee \text{ polymorphisme de } c^S_k \iff S = \emptyset,\ S = \{0\} \text{ ou } S\setminus\{0\} = \{j,\dots,k\} \text{ pour un certain } j > 0.$$

Question 3. Soit $k > 0$. Pour tout ensemble $S \subseteq \{0,\dots,k\}$, on définit $S^\perp = \{k - i \mid i \in S\}$. Montrez que $\vee$ est un polymorphisme de c^k_S si et seulement si $\wedge$ est un polymorphisme de $c^k_{S^\perp}$.

Correction. On commence par observer que pour tout tuple τ, $\tau \in c^k_S \iff \overline{\tau} \in c^k_{S^\perp}$ (ici, $\overline{\tau}$ désigne la négation du tuple τ). Soient $\tau_1, \tau_2 \in c^k_S$. Alors,

$$\begin{aligned}
\vee \in \mathrm{Pol}(c^k_S) &\iff \forall \tau_1, \tau_2 \in c^k_S,\ \tau_1 \vee \tau_2 \in c^k_S \\
&\iff \forall \overline{\tau_1}, \overline{\tau_2} \in c^k_{S^\perp},\ \overline{\tau_1 \vee \tau_2} \in c^k_{S^\perp} \\
&\iff \forall \overline{\tau_1}, \overline{\tau_2} \in c^k_{S^\perp},\ \overline{\tau_1} \wedge \overline{\tau_2} \in c^k_{S^\perp} \\
&\iff \wedge \in \mathrm{Pol}(c^k_{S^\perp})
\end{aligned}$$

On rappelle que CSP(Γ) est *résolu* par AC si la fermeture arc cohérente de toute instance insatisfiable de CSP(Γ) contient un domaine vide.

Question 4. Soit Γ un langage booléen. Démontrez que CSP(Γ) est résolu par AC si et seulement si Γ admet un polymorphisme parmi $\{0, 1, \wedge, \vee\}$.

Correction. D'après la version du théorème de Schaefer présentée en cours, si Γ admet un polymorphisme parmi $\{0, 1, \wedge, \vee\}$ alors CSP(Γ) est résolu par AC. Pour la réciproque, on va utiliser le théorème de Dalmau-Pearson : si CSP(Γ) est résolu par AC, alors pour tout $k > 0$ le langage Γ admet un polymorphisme totalement symétrique d'arité k.

Soit f un polymorphisme totalement symétrique de Γ d'arité 2. Si $f(1,1) = f(0,0)$, alors $g : \{0,1\} \to \{0,1\}$ défini par $g(a) = f(a,a)$ est un polymorphisme constant de Γ (0 ou 1). Si $f(1,1) = 1$ et $f(0,0) = 0$, alors on a deux cas possibles : soit $f(1,0) = f(0,1) = 1$, auquel cas $f(a,b) = a \vee b$, soit $f(1,0) = f(0,1) = 0$ et donc $f(a,b) = a \wedge b$. Enfin, si $f(1,1) = 0$ et $f(0,0) = 1$ alors $g(a,b) = f(f(a,b), f(a,b))$ est totalement symétrique et satisfait $g(1,1) = 1$, $g(0,0) = 0$, ce qui nous ramène au deuxième cas.

Question 5. Déduisez-en une caractérisation des langages finis $\Gamma \subset \{c^k_S \mid k > 0, S \subseteq \{0,\dots,k\}\}$ résolus par AC.

Correction. En combinant les réponses aux questions précédentes, on déduit :

$$\text{CSP}(\Gamma) \text{ résolu par AC} \iff \begin{cases} \forall c^k_S \in \Gamma, S = \emptyset \text{ ou } 0 \in S, \text{ ou} \\ \forall c^k_S \in \Gamma, S = \emptyset \text{ ou } k \in S, \text{ ou} \\ \forall c^k_S \in \Gamma, S = \emptyset, S = \{0\} \text{ ou } \exists j > 0 : S\setminus\{0\} = \{j,\dots,k\}, \text{ ou} \\ \forall c^k_S \in \Gamma, S = \emptyset, S = \{k\} \text{ ou } \exists j < k : S\setminus\{k\} = \{0,\dots,j\}. \end{cases}$$

Exercice 4

On dit qu'une fonction booléenne $c : D^2 \to \{0,1\}$ est ZOA (*Zero-One-All*) s'il existe $D_1, D_2 \subseteq D$ tels que

$$(1) : \forall (d_1, d_2) \in c,\ \text{ on a } d_1 \in D_1 \text{ et } d_2 \in D_2$$
$$(2) : \forall \alpha \in D_1, |\{d \in D_2 \mid (\alpha, d) \in c\}| \in \{1, |D_2|\}$$
$$(3) : \forall \beta \in D_2, |\{d \in D_1 \mid (d, \beta) \in c\}| \in \{1, |D_1|\}$$

Par exemple, considérons les fonctions

$$c_1(x,y) = \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 2 & 1 \end{bmatrix} \end{array} \qquad c_2(x,y) = \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 2 & 1 \\ 3 & 3 \end{bmatrix} \end{array}$$

définies sur $D = \{1,2,3\}$. Informellement, la propriété ZOA signifie que chaque valeur qui apparaît dans une colonne est compatible avec soit *toutes* les valeurs apparaissant dans l'autre colonne, soit *une seule*. Ainsi, la fonction c_1 est ZOA avec $D_1 = \{1,2\}$ et $D_2 = \{1,2,3\}$, mais la fonction c_2 ne l'est pas car on a $(1,3),(3,3) \in c_2$ mais $(2,3) \notin c_2$.

Question 1. Soit D un domaine fini et l'opération $f_{dd} : D^3 \to D$ définie par

$$f_{dd}(x,y,z) = \begin{cases} x & \text{si } y \neq z \\ z & \text{sinon.} \end{cases}$$

Démontrez que si une fonction booléenne $c : D^2 \to \{0,1\}$ est ZOA, alors f_{dd} est un polymorphisme de c.

Correction. Soient $\tau_1 = (\alpha_1, \beta_1)$, $\tau_2 = (\alpha_2, \beta_2)$ et $\tau_3 = (\alpha_3, \beta_3)$ trois tuples de c. Notons que par la condition (1), chaque α_i appartient à D_1 et chaque β_i appartient D_2. On va démontrer que $f_{dd}(\tau_1, \tau_2, \tau_3)$ est également un tuple de c. On distingue quatre cas :

1) $\alpha_2 \neq \alpha_3$ et $\beta_2 \neq \beta_3$. Dans ce cas, $f_{dd}(\tau_1, \tau_2, \tau_3) = (\alpha_1, \beta_1) = \tau_1 \in c$.
2) $\alpha_2 \neq \alpha_3$ et $\beta_2 = \beta_3$. Comme $\tau_2, \tau_3 \in c$, on a $\{\alpha_2, \alpha_3\} \subseteq \{d \in D_1 \mid (d, \beta_2) \in c\}$ et par la condition (3), $|\{d \in D_1 \mid (d, \beta_2) \in c\}| = |D_1|$. Avec la condition (1) cela implique $\{d \in D_1 \mid (d, \beta_2) \in c\} = D_1$ et finalement $f_{dd}(\tau_1, \tau_2, \tau_3) = (\alpha_1, \beta_2) \in c$.
3) $\alpha_2 = \alpha_3$ et $\beta_2 \neq \beta_3$. Ce cas est symétrique au cas précédent, en utilisant la condition (2) au lieu de la condition (3).
4) $\alpha_2 = \alpha_3$ et $\beta_2 = \beta_3$. Dans ce cas, $f_{dd}(\tau_1, \tau_2, \tau_3) = (\alpha_3, \beta_3) = \tau_3 \in c$.

$f_{dd}(\tau_1, \tau_2, \tau_3)$ est donc un tuple de c pour tout choix de tuples $\tau_1, \tau_2, \tau_3 \in c$: f_{dd} est un polymorphisme de c.

Question 2. Soit Γ un langage dont toutes les fonctions sont ZOA. En vous appuyant sur la réponse à la question précédente, démontrez que Γ a la propriété de *bounded width*. Quel algorithme utiliseriez-vous pour résoudre CSP(Γ) ?

Correction. L'opération f_{dd} de la question précédente satisfait l'identité

$$f_{dd}(x,x,y) = f_{dd}(x,y,x) = f_{dd}(y,x,x) = x$$

pour tout $x, y \in D$ et est polymorphisme de Γ. De plus, l'opération $g : D^4 \to D$ définie par

$$g(x,y,z,w) = f(f(x,y,z),z,w)$$

satisfait l'identité

$$g(x, x, x, y) = g(x, x, y, x) = g(x, y, x, x) = g(y, x, x, x) = x$$

et est également polymorphisme de Γ (car Pol(Γ) est un clone concret, donc invariant par composition). Par le théorème de Barto-Kozik, on déduit donc que Γ a la propriété de *bounded width* et CSP(Γ) est résolu par SAC.

A priori, il est possible que CSP(Γ) ne soit pas résolu par AC (par exemple, le langage $\{c_\oplus\}$ est ZOA mais n'est pas résolu par AC). En l'absence d'information additionnelle sur Γ, SAC est donc un bon choix d'algorithme pour CSP(Γ).

Exercice 5

Pour tout entier naturel $n \geq 2$ on définit l'hypergraphe H_n dont l'ensemble des sommets est $\{x_1, \ldots, x_n, y_1, \ldots, y_n\}$ et l'ensemble des arêtes est $\{\{x_i, x_{i+1}, y_{i+1}\} \mid 1 \leq i \leq n-1\} \cup \{\{x_i, y_i, y_{i+1}\} \mid 1 \leq i \leq n-1\}$.

Question 1. Démontrez que pour tout $n \geq 2$, l'hypertreewidth de H_n est égale à 1.

Correction. On construit une décomposition arborescente $(T, (B_t)_{t \in V(T)})$ de H_n comme suit. T est un chemin $(t_1^y, t_1^x, t_2^y, t_2^x, \ldots, t_{n-1}^y, t_{n-1}^x)$ de taille $2n-2$ dont chaque sommet est associé à une arête de H_n :

$$\forall i \in \{1, \ldots, n-1\}, B_{t_i^y} = \{x_i, y_i, y_{i+1}\}$$
$$\forall i \in \{1, \ldots, n-1\}, B_{t_i^x} = \{x_i, x_{i+1}, y_{i+1}\}$$

Notons que chaque arête de H_n est effectivement contenue dans au moins un sac. Il reste donc à vérifier que tout sommet de H_n induit un sous-arbre connexe de T. y_1 appartient à un seul sac, celui de t_1^y. Pour $1 < i < n-1$, y_i appartient à exactement trois sacs : ceux de t_{i-1}^y, t_{i-1}^x et t_i^y. Ces sacs sont consécutifs dans T donc le sous-arbre induit est connexe. Finalement, y_n appartient aux sacs de t_{n-1}^y, t_{n-1}^x, qui sont également consécutifs dans T. Le cas des x_i est symétrique.

Chaque sac de cette décomposition arborescente de H_n est entièrement contenu dans une arête, donc l'hypertreewidth de H_n est au plus 1. Comme H_n contient au moins une arête, son hypertreewidth est au moins 1, ce qui conclut la démonstration.

Question 2. Existe-t-il un entier $n \geq 2$ et un réseau de contraintes N dont l'hypergraphe est H_n, tels que N n'a pas de solution mais appliquer la 3-cohérence forte sur N ne vide aucun domaine ? Justifiez votre réponse.

Correction. Non. Par la réponse à la question précédente et le fait que les arêtes de H_n contiennent toutes 3 sommets, la treewidth de H_n est d'au plus 2. Par le cours, appliquer la $(2+1)$-cohérence forte sur N permet de déterminer l'existence d'une solution, c'est-à-dire qu'un domaine sera forcément vidé si N n'a pas de solution.