TD 7. Feuille de révision sur la logique propositionnelle

Exercice 1. Algorithme DPLL

Considérons la formule propositionnelle φ_1 ci-dessous.

$$\varphi_1 \stackrel{\text{def}}{=} \big((P_2 \Rightarrow P_1) \Rightarrow P_3\big) \wedge \neg P_3 \wedge \big(P_2 \Rightarrow (P_1 \vee P_3)\big)\ .$$

(a) Mettre φ_1 sous forme normale négative : calculer $\mathrm{nnf}(\varphi_1)$.

(b) Mettre $\mathrm{nnf}(\varphi_1)$ sous forme clausale.

(c) Appliquer l'algorithme DPLL à la formule obtenue en (b). Plus précisément, dessiner un arbre de recherche DPLL comme vu en cours (c.f. figures 15 à 18 des notes de cours).

Exercice 2. Calcul des séquents propositionnel

Considérons la formule propositionnelle φ_2 ci-dessous.

$$\varphi_2 \stackrel{\text{def}}{=} \big((P_2 \Rightarrow P_1) \wedge \neg P_3\big) \vee \big((P_2 \Rightarrow (P_1 \vee P_3)) \Rightarrow P_3\big)\ .$$

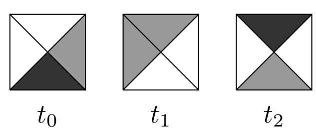
(a) Mettre φ_2 sous forme normale négative : calculer $\mathrm{nnf}(\varphi_2)$.

(b) Faire une recherche de preuve dans le calcul des séquents propositionnel vu en cours sur la formule $\mathrm{nnf}(\varphi_2)$ obtenue en (b).

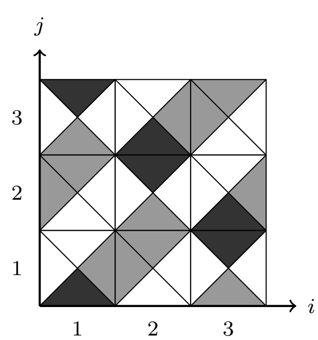
(c) Quel lien y a-t-il entre les résultats obtenus à la question (c) de l'exercice 1 et à la question (b) de cet exercice ?

Exercice 3. Modélisation en logique propositionnelle

Le problème qui nous intéresse est le problème de *pavage carré*. L'entrée du problème est un entier naturel $n \in \mathbb{N}$ et un *catalogue*, qui est un ensemble fini C de *tuiles* carrées avec une couleur par côté comme les trois tuiles t_0, t_1 et t_2 ci-dessous :



Le but pour un catalogue C donné est de déterminer s'il est possible de couvrir un carré de dimension $n \times n$ en respectant les couleurs. On peut pour cela réutiliser les tuiles du catalogue, mais celles-ci ne peuvent pas être tournées. Voici ci-dessous un exemple de pavage 3×3 avec le catalogue $C \stackrel{\text{def}}{=} \{t_0, t_1, t_2\}$:



Formellement, un catalogue C est associé à deux relations binaires $H \subseteq C \times C$ de contraintes horizontales et $V \subseteq C \times C$ de contraintes verticales, où $(t, t') \in H$ si la couleur de droite de t est la même que la couleur de gauche de t' et $(t, t') \in V$ si la couleur du haut de t est la même que la couleur du bas de t'. Par exemple, pour notre catalogue $C \stackrel{\text{def}}{=} \{t_0, t_1, t_2\}$, on a les contraintes suivantes :

$$H = \{(t_0, t_1), (t_1, t_0), (t_1, t_2), (t_2, t_0), (t_2, t_2)\}\ , \qquad V = \{(t_0, t_1), (t_1, t_2), (t_2, t_0)\}\ .$$

Un *pavage carré de dimension n par C* est alors une fonction $p\colon (\{1, \dots, n\} \times \{1, \dots, n\}) \to C$ telle que

1. pour tout $1 \leq i < n$ et $1 \leq j \leq n$, si $p(i, j) = t$ et $p(i + 1, j) = t'$ alors $(t, t') \in H$ et
2. pour tout $1 \leq i \leq n$ et $1 \leq j < n$, si $p(i, j) = t$ et $p(i, j + 1) = t'$ alors $(t, t') \in V$.

Fixons un catalogue C avec ses relations H et V ainsi qu'une dimension n. Notre objectif est d'écrire une formule propositionnelle φ_3 (qui dépend de C, H, V et n) telle que φ_3 soit satisfiable si et seulement s'il existe un pavage carré de dimension n par C. On utilisera pour cela des propositions $P_{i,j,t}$ où $1 \leq i \leq n$, $1 \leq j \leq n$ et $t \in C$, qui exprimeront le fait que $p(i, j) = t$ dans notre pavage. Dans les questions suivantes, on cherche des formules propositionnelles qui dépendent de C, H, V, et n.

(a) Écrire une formule propositionnelle $\varphi_{3,\geq 1}$ qui impose que chaque case du carré $n \times n$ reçoit au moins une tuile.

(b) Écrire une formule propositionnelle $\varphi_{3,\leq 1}$ qui impose que chaque case du carré $n \times n$ reçoit au plus une tuile.

(c) Écrire une formule propositionnelle $\varphi_{3,H}$ qui impose que deux cases adjacentes horizontalement du carré $n \times n$ respectent la contrainte H et une formule propositionnelle $\varphi_{3,V}$ qui impose que deux cases adjacentes verticalement du carré $n \times n$ respectent la contrainte V.

(d) Écrire la formule propositionnelle φ_3 en utilisant les réponses précédentes.