TD 0 - Introduction

Première partie – Problèmes et codages

Exercice 1. SAT et TAUTO

Les formules booléennes sur un ensemble fini de variables A sont définies par la grammaire suivante :

$$\varphi, \psi \in \mathsf{F}(A) ::= a \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \neg\varphi.$$

Étant donnée une fonction $\sigma : A \to \{\top, \bot\}$ qui donne à chaque variable une valeur de vérité, on peut définir une fonction d'interprétation :

$$\hat{\sigma}(a) := \sigma(a) \qquad \hat{\sigma}(\top) := \top \qquad \hat{\sigma}(\bot) := \bot$$
$$\hat{\sigma}(\varphi \vee \psi) := \hat{\sigma}(\varphi) \vee \hat{\sigma}(\psi) \qquad \hat{\sigma}(\varphi \wedge \psi) := \hat{\sigma}(\varphi) \wedge \hat{\sigma}(\psi) \qquad \hat{\sigma}(\neg\varphi) := \neg\hat{\sigma}(\varphi)$$

où $\vee$, $\wedge$ et $\neg$ sont interprétés respectivement par la disjonction, conjonction et négation sur les booléens.

On dit d'une formule $\varphi \in \mathsf{F}(A)$ qu'elle est satisfiable s'il existe une fonction d'évaluation $\sigma : A \to \{\top, \bot\}$ telle que $\hat{\sigma}(\varphi) = \top$.

On considère, pour un ensemble A donné, le problème suivant :

SAT *Satisfiabilité d'une formule booléenne*
Étant donnée une formule $\varphi \in \mathsf{F}(A)$, est-elle satisfiable ?

Question 1. Déterminer pour chacune des formules suivantes si elles sont satisfiables :
a - $(a \vee b) \wedge (\neg a \vee \neg b)$
b - $a \wedge (b \wedge \neg b)$

Solution :

a- Satisfiable, par exemple avec $\sigma = [a \mapsto \top;\ b \mapsto \bot]$.

b- Non-satisfiable :

a	b	$\neg b$	$b \wedge \neg b$	$a \wedge (b \wedge \neg b)$
$\top$	$\top$	$\bot$	$\bot$	$\bot$
$\top$	$\bot$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$\bot$	$\bot$	$\top$	$\bot$	$\bot$

Question 2. Définir l'ensemble d'instances et l'ensemble d'instances positives de SAT.

Solution :

instances $I_{\mathsf{SAT}} := \mathsf{F}(A)$.

instances positives $E_{\mathsf{SAT}} := \left\{ \varphi \in \mathsf{F}(A) \;\middle|\; \exists \sigma \in \{\top, \bot\}^A :\ \hat{\sigma}(\varphi) = \top \right\}$.

Une tautologie est une formule $\varphi \in \mathsf{F}(A)$ qui est universellement vraie, autrement dit telle que

$$\forall \sigma \in \{\top, \bot\}^A,\ \hat{\sigma}(\varphi) = \top.$$

On considère maintenant le problème TAUTO :

TAUTO *Universalité d'une formule booléenne*
La formule $\varphi \in \mathsf{F}(A)$ est-elle une tautologie ?

Question 3. Définir l'ensemble d'instances et l'ensemble d'instances positives de TAUTO.

Solution :

instances $I_{\text{TAUTO}} := \mathsf{F}(A)$.
Remarquons donc que $I_{\text{TAUTO}} = I_{\text{SAT}}$.
instances positives $E_{\text{TAUTO}} := \left\{ \varphi \in \mathsf{F}(A) \;\middle|\; \forall \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(\varphi) = \top \right\}$.

Question 4. Trouvez une fonction $f : I_{\text{TAUTO}} \to I_{\text{SAT}}$ telle que

$$\forall \varphi \in I_{\text{TAUTO}}, \; \varphi \in E_{\text{TAUTO}} \Leftrightarrow f(\varphi) \notin E_{\text{SAT}}.$$

Solution :

Commençons par comprendre la question, en dépliant les définitions. On demande une fonction $f : I_{\text{TAUTO}} \to I_{\text{SAT}}$. Rappelons nous que $I_{\text{TAUTO}} = I_{\text{SAT}} = \mathsf{F}(A)$. On cherche donc une fonction qui à une formule f associe une autre formule, qu'on appellera $f(\varphi)$.
On demande à ces formules de respecter la relation suivante :

$$\forall \varphi \in I_{\text{TAUTO}}, \; \varphi \in E_{\text{TAUTO}} \Leftrightarrow f(\varphi) \notin E_{\text{SAT}}.$$

À nouveau, en remplaçant les I, E_x par leurs définitions, on obtient :

$$\forall \varphi \in \mathsf{F}(A), \; \left(\forall \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(\varphi) = \top \right) \Leftrightarrow \neg \left(\exists \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(f(\varphi)) = \top \right).$$

On peut tranformer l'énoncé de droite avec l'équivalence logique $\neg \exists x, P(x) \Leftrightarrow \forall x, \neg P(x)$, ce qui nous donne :

$$\forall \varphi \in \mathsf{F}(A), \; \left(\forall \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(\varphi) = \top \right) \Leftrightarrow \left(\forall \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(f(\varphi)) = \bot \right).$$

On voit qu'il s'agit de transformer des $\top$ en $\bot$, ce qui nous donne l'idée d'examiner la négation, c'est à dire la fonction $\neg : \mathsf{F}(A) \to \mathsf{F}(A)$ qui associe à une formule φ la formule $\neg\varphi$. Remarquons alors que pour toute fonction $\sigma : A \to \{\top, \bot\}$, on a :

$$\hat{\sigma}(\neg\varphi) = \begin{cases} \top & \text{si } \hat{\sigma}(\varphi) = \bot \\ \bot & \text{si } \hat{\sigma}(\varphi) = \top \end{cases}$$

Cela signifie qu'on a

$$\forall \varphi \in \mathsf{F}(A), \forall \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(\varphi) = \top \Leftrightarrow \hat{\sigma}(\neg(\varphi)) = \bot.$$

On peut maintenant montrer que la fonction définie par $f(\varphi) := \neg\varphi$ est une réponse correcte à la question. Soit une formule $\varphi \in I_{\text{TAUTO}}$:

$$\begin{aligned} \varphi \in E_{\text{TAUTO}} &\Leftrightarrow \forall \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(\varphi) = \top \\ &\Leftrightarrow \forall \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(f(\varphi)) = \hat{\sigma}(\neg\varphi) = \bot \\ &\Leftrightarrow \neg \left(\exists \sigma \in \{\top, \bot\}^A : \; \hat{\sigma}(f(\varphi)) = \top \right) \\ &\Leftrightarrow f(\varphi) \notin E_{\text{SAT}}. \end{aligned}$$

Exercice 2. REACH

Définir l'ensemble d'instances et l'ensemble d'instances positives du problème suivant :

REACH *Accessibilité entre deux noeuds d'un graphe*
Soit $G = \langle V, E \rangle$ un graphe orienté, et $u, v \in V$ deux de ses noeuds. Peut-on trouver un chemin dans G allant de u à v ?

Solution :

instances $I_{\text{REACH}} := \{\langle G, u, v\rangle \mid G = \langle V, E\rangle \text{ est un graphe fini orienté, et } u, v \in V\}$

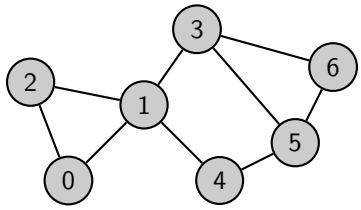
instances positives $E_{\text{REACH}} := \left\{ \langle G, u, v\rangle \;\middle|\; \exists n, \exists (u_i)_{1 \leqslant i \leqslant n} \in V^n : \begin{array}{l} u_1 = u,\ u_n = v, \\ \forall i < n,\ \langle u_i, u_{i+1}\rangle \in E \end{array} \right\}.$

Exercice 3. COLOUR

Soit $k \in \mathbb{N}$ un entier. Dans cet exercice, on appelle « couleurs » des entiers tirés d'un ensemble fini de la forme $\{1, 2, \ldots, k\}$. Un coloriage est une fonction $\kappa : V \to \{1, \ldots, k\}$ associant à chaque sommet une couleur Graphiquement, on peut représenter un coloriage en associant k couleurs distinctes aux entiers $\{1, \ldots, k\}$, et en coloriant chaque sommet $u \in V$ du graphe de la couleur associée à l'entier $\kappa(u)$.

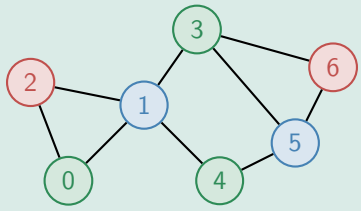
COLOUR *k-coloriabilité d'un graphe*
Soit $G = \langle V, E\rangle$ un graphe non-orienté, et k un entier, existe-t'il un coloriage tel que deux sommet adjacents aient toujours deux couleur différentes ?

Question 1. Le graphe suivant est-il 2-coloriable ? Est-il 3-coloriable ?



Solution :

Le graphe n'est pas 2-coloriable, à cause des triangles : par exemple, le triangle $(0, 1, 2)$ nécessite l'utilisation d'au moins trois couleurs différentes.
Il est en revanche 3-coloriable :



Question 2. Définir l'ensemble d'instances et l'ensemble d'instances positives de COLOUR.

Solution :

instances $I_{\text{COLOUR}} := \{\langle G, k\rangle \mid G = \langle V, E\rangle \text{ est un graphe fini non-orienté, et } k \in \mathbb{N}\}$.

instances positives $E_{\text{SAT}} := \left\{ \langle G, k\rangle \;\middle|\; \exists \kappa \in \{1, \ldots, k\}^V : \forall \langle a, b\rangle \in E,\ \kappa(a) \neq \kappa(b) \right\}$.

Exercice 4. Codages

Question 1. Proposer un codage pour chacun des problèmes SAT, REACH et COLOUR.

Solution :

SAT On encode le problème sur l'alphabet $\Sigma := A \cup \{(,),\mathtt{|},\&,!\}$, avec la fonction $\lfloor - \rfloor : \mathsf{F}(A) \to \Sigma^\star$ définie par induction sur les formules comme suit :

$$\lfloor a \rfloor := \mathtt{a} \qquad \lfloor \neg\varphi \rfloor := !\lfloor\varphi\rfloor$$
$$\lfloor \varphi \vee \psi \rfloor := (\lfloor\varphi\rfloor\mathtt{|}\lfloor\varphi\rfloor) \qquad \lfloor \varphi \wedge \psi \rfloor := (\lfloor\varphi\rfloor\&\lfloor\varphi\rfloor)$$

REACH On encode le problème sur l'alphabet $\Sigma := \{0,1,[,],(,),\mathtt{,}\}$. On va utiliser deux codages auxiliaires :

- $\lfloor - \rfloor_2 : \mathbb{N} \to \{0,1\}^\star$ qui encode les entiers par leur représentation binaire ;
- $\lfloor - \rfloor_* : \left(\{0,1,(,),\mathtt{,}\}^\star\right)^\star \to \Sigma^\star$, qui encode les listes de mots sur l'alphabet (privé des symboles [,]). Cette fonction va simplement insérer des , entre chaque mot, et ajouter de part et d'autre les délimiteurs [et].

Ensuite, on numérote les sommets du graphe avec une fonction $\mathrm{nb} : V \to \mathbb{N}$. Pour représenter le graphe $G = \langle V, E\rangle$, on calcule la liste L des mots $(\lfloor \mathrm{nb}(u)\rfloor_2\mathtt{,}\lfloor \mathrm{nb}(v)\rfloor_2)$ pour toute arrête $\langle u, v\rangle \in E$. Enfin, l'instance G, u, v est codée par le mot $\lfloor L\rfloor_*\mathtt{,}\lfloor \mathrm{nb}(u)\rfloor_2\mathtt{,}\lfloor \mathrm{nb}(v)\rfloor_2$.

COLOUR On utilise le même alphabet que pour le problème REACH. Comme précédemment, on numérote les sommets et on utilise un codage binaire des entiers. Pour les arrêtes non-orientées, on choisit de les noter avec le sommet d'indice minimal en premier, i.e. $\langle 0,1\rangle$ et pas $\langle 1,0\rangle$. À cette différence près, on peut comme dans le cas précédent obtenir une liste $L \in \left(\{0,1,(,),\mathtt{,}\}^\star\right)^\star$ qui code l'ensemble E. Le code d'une instance $\langle G, k\rangle$ est alors le mot $\lfloor L\rfloor_*\mathtt{,}\lfloor k\rfloor_2 \in \Sigma^\star$.

Question 2. Donner les codages des instances données en exemple dans les exercices 1 et 3.

Solution :

- $\lfloor (a \vee b) \wedge (\neg a \vee \neg b)\rfloor = ((\mathtt{a}|\mathtt{b})\&(!\mathtt{a}|!\mathtt{b}))$.
- $\lfloor a \wedge (b \wedge \neg b)\rfloor = (\mathtt{a}\&(\mathtt{b}|!\mathtt{b}))$.
- $\mathrm{code}(G,2) =$ [(000,010),(000,001),(001,010),(011,110),(001,101),(100,101),(001,100),(011,101),(101,110)],010.
- $\mathrm{code}(G,3) =$ [(000,010),(000,001),(001,010),(011,110),(001,101),(100,101),(001,100),(011,101),(101,110)],011.

Deuxième partie – Modèles de calcul

Exercice 5.

Soit $L \subseteq \Sigma^\star$ un langage, et $\mathscr{M} = \langle \mathbb{M}, \models \rangle$ un modèle.

Question. Parmi les affirmations suivantes, lesquelles sont équivalentes à l'énoncé « le langage L est expressible par le modèle $\mathscr{M}$ » ?

1. $L \in \mathscr{M}$
2. $\exists M \in \mathbb{M} :\ L \in M$
3. $L \in \mathrm{Lang}(\mathscr{M})$
4. $L \in \mathbb{M}$
5. $\exists M \in \mathbb{M} :\ L = L_M$
6. $\exists M \in \mathbb{M} :\ L_M \in L$
7. $\exists M \in \mathbb{M} :\ \forall w,\ w \in L \Leftrightarrow w \models M$
8. $\exists M \in \mathbb{M} :\ \forall w,\ w \in L \Rightarrow w \in L_M$

Solution :

1. N'a aucun sens : $\mathscr{M}$ est une paire, il ne contient pas directement d'éléments.
2. N'a probablement aucun sens : $\mathbb{M}$ est un ensemble de machines, donc $M \in \mathbb{M}$ est une machine. Cela signifie que $L \in M$ n'a de sens que si M est un ensemble en plus d'être une machine, ce qui n'est en général pas le cas.
3. C'est vrai.
4. Pas de sens en général, à moins que les « machines » du modèle soient elle-mêmes des langages.
5. Ça aussi, c'est vrai :
$$L \in \mathrm{Lang}(\mathscr{M}) = \{L_M \subseteq \Sigma^\star \mid M \in \mathbb{M}\} \Leftrightarrow \exists M \in \mathbb{M} : L = L_M.$$
6. N'a pas de sens : L_M est un langage (un ensemble de mots), et L aussi. Donc L contient des mots, et pas des langages.
7. Encore vrai, car $L = L_M$ est équivalent à $\forall w,\ w \in L \Leftrightarrow w \in L_M$, et $w \in L_M = \{w \in \Sigma^\star \mid w \models M\}$ est par définition équivalent à $w \models M$.
8. Ça a du sens, mais c'est faux : cet énoncé est équivalent à
$$\exists L' \in \mathrm{Lang}(\mathscr{M}) :\ L \subseteq L'.$$
Supposons par exemple que le modèle contient une machine M^* qui dit « oui » sur n'importe quelle entrée. Le langage de cette machine est donc $L_{M^*} = \Sigma^\star$, l'ensemble de tous les mots. Ce langage inclut tous les autres, et donc l'énoncé est vrai pour n'importe quel langage L. Autrement dit, si notre notion de « un modèle $\mathscr{M}$ peut exprimer le langage L » était cet énoncé, alors n'importe quel modèle capable de répondre bêtement « oui » sans lire la question serait universel, c'est à dire qu'il pourrait exprimer tous les langages possibles...

Exercice 6.

Montrer que pour toute paire de modèles $\mathscr{M}_1$, $\mathscr{M}_2$ on a :

$$\mathscr{M}_1 \preceq \mathscr{M}_2 \Leftrightarrow \text{Lang}(\mathscr{M}_1) \subseteq \text{Lang}(\mathscr{M}_2)$$

Solution :

$$\begin{aligned}
\mathscr{M}_1 \preceq \mathscr{M}_2 &\Leftrightarrow \forall M_1 \in \mathbb{M}_1, \exists M_2 \in \mathbb{M}_2 : \ \forall w \in \Sigma^\star, \ w \models_1 M_1 \Leftrightarrow w \models_2 M_2 \\
&\Leftrightarrow \forall M_1 \in \mathbb{M}_1, \exists M_2 \in \mathbb{M}_2 : \ L_{M_1} = L_{M_2} \\
&\Leftrightarrow \forall L \in \text{Lang}(\mathscr{M}_1), \exists M_2 \in \mathbb{M}_2 : \ L = L_{M_2} \\
&\Leftrightarrow \forall L \in \text{Lang}(\mathscr{M}_1), \ L \in \text{Lang}(\mathscr{M}_2) \\
&\Leftrightarrow \text{Lang}(\mathscr{M}_1) \subseteq \text{Lang}(\mathscr{M}_2)
\end{aligned}$$