Université Paris Diderot – L3 Informatique 10 Janvier 2017

Outils Logiques – Examen Final (durée : 3 h)

Exercice 1 (DPLL) Soient $x_1, ..., x_n \in V$ $(n > 0)$ des variables propositionnelles. Une *clause disjonctive complète et ordonnée sur* $x_1, .., x_n$ est une disjonction de litteraux de la forme $(l_1 \vee \ldots \vee l_n)$ où, pour chaque $1 \leq i \leq n$, soit $l_i = x_i$, soit $l_i = \neg x_i$.

Nous allons écrire *n-cdco* pour abréger "clause disjonctive complète et ordonnée sur $x_1, .., x_n$". Par exemple, $(x_1 \vee \neg x_2 \vee \neg x_3)$ est une 3-cdco.

1. Écrire toutes les 2-cdco.

   Correction: Les quatre 2-cdco sont :
   (a) $(x_1 \vee x_2)$,
   (b) $(x_1 \vee \neg x_2)$,
   (c) $(\neg x_1 \vee x_2)$,
   (d) $(\neg x_1 \vee \neg x_2)$.

2. Combien y a-t-il de n-cdco distinctes ?

   Correction: Pour chaque $1 \leq i \leq n$, le i-ème litteral d'une n-cdco est soit x_i soit $\neg x_i$. Il y a donc 2^n n-cdco distinctes. Cette reponse, et sa version abrégé "2^n" ont obtenu la totalité des points. Un argument plus rigoureux pourrait se baser sur la bijection évidente entre les n-cdco et les séquences binaires de longueur n, qui sont au nombre de 2^n.

3. Montrer que si d et d' sont deux n-cdco distinctes, alors $d \not\models d'$, c'est à dire, exhiber une affectation v telle que $[\![d]\!]v = 1$ et $[\![d']\!]v = 0$.

   Correction: Pour une n-cdco $d = (l_1 \vee \ldots \vee l_n)$ donnée, soit v_d l'affectation définie par $v_d(x_i) = 1$ si $l_i = \neg x_1$, $v_d(x_i) = 0$ si $l_i = x_i$ (et $v_d(z) = 0$ si $z \neq x_1, ..., x_n$).

   Lemme : Pour toute paire de n-cdco $d = (l_1 \vee \ldots \vee l_n)$ et $d' = (l'_1 \vee \ldots \vee l'_n)$, on a que :

   $$[\![d']\!]v_d = 0 \Leftrightarrow d = d'$$

   Preuve :
   $(\Rightarrow)$ Soit $[\![d']\!]v_d = 0$. Vu qu'une clause disjonctive est fausse si et seulement si tous les litteraux qui la composent sont faux, on a, pour tout $1 \leq i \leq n$, $l'_i = x_i$ si $v_d(x_i) = 0$ et $l'_i = \neg x_i$ si $v_d(x_i) = 1$, donc $l'_i = l_i$, d'où $d' = d$.
   $(\Leftarrow)$ Chaque litteral de la clause disjonctive d est faux relativement à v_d, donc $[\![d]\!]v_d = 0$.

   Vu le lemme, pour toute n-cdco $d' \neq d$, on a $[\![d]\!]v_{d'} = 1$ et $[\![d']\!]v_{d'} = 0$, donc $d \not\models d'$.

Soit c_n la formule en forme conjonctive normale obtenue par conjonction de toutes les n-cdco (l'ordre est inessentiel). Ainsi par exemple, à l'ordre des 2-cdco près, on a :

$$c_2 = ((x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2))$$

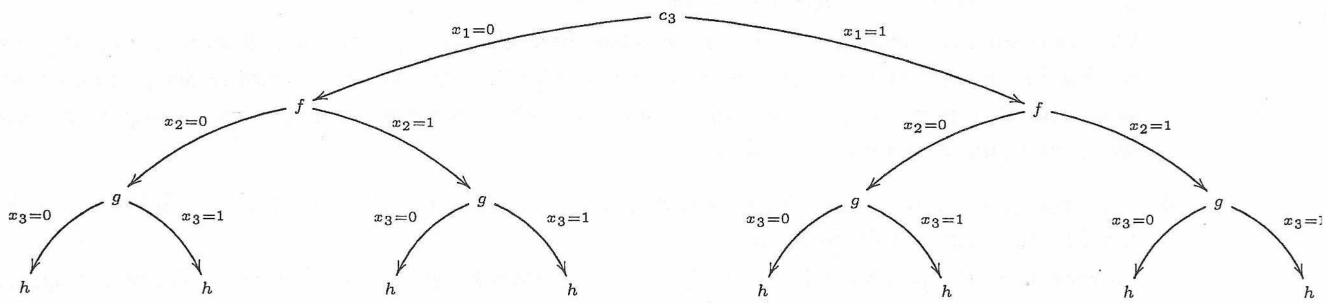
4. Écrire la formule c_3 et appliquer à cette formule la fonction `dpll` vue en cours *en suivant les indications ci-dessous.*

   Dessiner l'arbre des appels récursifs de la fonction :

- les nœuds de l'arbre seront étiquetés par des formes conjonctive normales (les arguments de `dpll`, donc c_3 sera l'étiquette de la racine de l'arbre) et les arêtes par les valeurs de vérité des variables de pivot.
- l'ordre des choix des variables de pivot est imposé : d'abord x_1, puis x_2 et enfin x_3.
- les feuilles de l'arbre seront soit de formes conjonctive normales contenant la clause disjonctive vide (echec), soit la forme conjonctive normale sans aucune clause disjonctive (succès).
- le résultat renvoyé par chaque appel de dpll sera écrit à côté du nœud correspondant à l'appel en question (donc par exemple, le résultat de l'appel initiale `dpll(`c_3`)` sera écrit à côté de la racine de l'arbre).

Correction:
$c_3 = ((x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3))$
Une bonne reponse à cette première question vaut 20% des points de (4).

Soient :
$f = ((x_2 \vee x_3) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3))$
$g = (x_3 \vee \neg x_3)$
$h = False =$ La clause conjonctive contenant une clause disjonctive vide.



La valeur renvoyée par DPLL est `False`, sur tous les sommets de l'arbre. Cette valeur devait être écrite à côté de chaque sommet.

5. Soit d une n-cdco et soit c_n^{-d} la forme conjonctive normale obtenue en retirant d de c_n.
   - Donner $c_2^{-(x_1 \vee x_2)}$.
   - Montrer que c_n^{-d} est satisfaisable pour tout n et tout d, c'est à dire, exhiber une affectation v telle que $[\![c_n^{-d}]\!]v = 1$.

   Correction:
   $c_2^{-(x_1 \vee x_2)} = ((x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \wedge \neg x_2))$. Une bonne reponse à cette première question vaut 20% des points de (5).

   On a vu dans le lemme de la correction du point 3 que chaque n-cdco différente de d est vrai pour v_d, donc $[\![c_n^{-d}]\!]v_d = 1$.

6. Montrer que c_n est une contradiction pour tout n, c'est à dire que pour toute affectation v on a $[\![c_n]\!]v = 0$.

   Correction: Soit v une affectation, et considérons la n-cdco $d_v = (l_1 \vee \ldots \vee l_n)$ définie par $l_i = x_i$ si $v(x_i) = 0$, $l_i = \neg x_i$ si $v(x_i) = 1$. On a $[\![d_v]\!]v = 0$, donc $[\![c_n]\!]v = 0$

Exercice 2 (Modélisation) Les étudiants de l'Université de Chapadmalal suivent un cursus leur permettant de valider un diplôme selon l'ensemble suivant de contraintes :

(a) Chaque étudiant doit choisir exactement un cours facultatif ;

(b) Chaque étudiant doit suivre au moins 5 cours obligatoires ;

(c) Un cours obligatoire peut accueillir au plus 20 étudiants.

On supposera que $E = \{1, \ldots, n\}$ est l'ensemble d'étudiants, $CO = \{O_1, \ldots, O_k\}$ ($k \geq 5$) l'ensemble de cours obligatoires, et $CF = \{F_1, \ldots, F_h\}$ ($h \geq 1$) l'ensemble de cours facultatifs.

1. Donner l'ensemble des variables propositionnelles qui sera utilisé pour modéliser les contraintes, et expliquer brièvement la signification de ces variables.
   Correction: L'ensemble des variables est $(E \times CO) \cup (E \times CF)$, c'est à dire

$$\{[i, O_j] \mid 1 \leq i \leq n, 1 \leq j \leq k\} \cup \{[i, F_j] \mid 1 \leq i \leq n, 1 \leq j \leq h\}$$

   Pour $e \in E$ et $c \in CO$ (resp. $c \in CF$, la variable $[e, c]$ reçoit la valeur 1 si et seulement si l'étudiant e suit le cours obligatoire (resp. facultatf) c.

2. Donner une formule en **forme conjonctive normale (CNF)** pour modéliser chaque contrainte (a), (b) et (c) ci-dessus.
   Correction:
   (a) Tout étudiant e suit au moins un cours facultatif :

$$\bigwedge_{e \in E} \bigvee_{1 \leq i \leq h} [e, F_i]$$

   Aucun étudiant e ne suit deux cours facultatifs :

$$\bigwedge_{e \in E} \bigwedge_{1 \leq i_1 < i_2 \leq h} (\neg[e, F_{i_1}] \vee \neg[e, F_{i_2}])$$

   (b) Pour chaque étudiant e, pour n'importe quelle sélection de $k - 5 + 1$ cours obligatoires, l'étudiant suit au moins un de ces cours.

$$\bigwedge_{e \in E} \bigwedge_{O_1 \leq i_1 < \ldots < i_{k-4} \leq O_k} \bigvee_{j=1 \ldots k-4} [e, O_{i_j}]$$

   (c) Pour chaque cours obligatoire c, au moins $n - 20$ étudiants ne suivent pas ce cours. Ceci est équivalent à : pour chaque cours obligatoire c, pour chaque sélection de $n - (n - 20) + 1$ étudiants il y a au moins un étudiant parmi eux qui ne suit pas ce cours c.

$$\bigwedge_{O_1 \leq c \leq O_k} \bigwedge_{1 \leq i_1 < \ldots < i_{21} \leq n} \bigvee_{j=1 \ldots 21} \neg[e_{i_j}, c]$$

3. Rajouter les deux contraintes spécifiques suivantes, toujours en **CNF**.
   (d) `Ana` et `Brigitte` veulent suivre *tous* leurs cours ensemble (facultatifs et obligatoires).
   Correction: On écrit A pour `Ana` et B pour `Brigitte`.

$$\bigwedge_{1 \leq i \leq k} (\neg[A, O_i] \vee [B, O_i]) \wedge (\neg[B, O_i] \vee [A, O_i]) \wedge \bigwedge_{1 \leq i \leq h} (\neg[A, F_i] \vee [B, F_i]) \wedge (\neg[B, F_i] \vee [A, F_i])$$

(e) `Carlos` et `Leo` ne veulent pas suivre le même cours facultatif.
Correction: On écrit C pour `Carlos` et L pour `Leo`.

$$\bigwedge_{1 \leq F_i \leq h} (\neg[C, F_i] \vee \neg[L, F_i])$$

Exercice 3 (Formes Normales) Soit $\mathcal{F}$ l'ensemble de toutes les formules contenant uniquement les opérateurs $\{\texttt{True}, \texttt{False}, \oplus\}$ et soit $z \in V$ une variable propositionnelle donnée. On considère les règles de transformation suivantes :

$$\begin{array}{lcl} \texttt{True} & \rightsquigarrow & (z \vee \neg z) \\ \texttt{False} & \rightsquigarrow & (z \wedge \neg z) \\ (X \oplus Y) & \rightsquigarrow & ((X \wedge \neg Y) \vee (\neg X \wedge Y)) \end{array}$$

qu'on utilise pour transformer chaque formule de l'ensemble $\mathcal{F}$ en une formule du calcul propositionnel vu en cours (ne contenant aucune occurrence de `True`, `False` et $\oplus$).

1. Compléter les séquences de transformations suivantes :

p_1		p_2		p_3
$(\texttt{False} \oplus x)$	$\rightsquigarrow$		$\rightsquigarrow$	
$(x \oplus \texttt{True})$	$\rightsquigarrow$		$\rightsquigarrow$	

Correction:

$$\texttt{False} \oplus x \rightsquigarrow (z \wedge \neg z) \oplus x \rightsquigarrow [(z \wedge \neg z) \wedge \neg x] \vee [\neg(z \wedge \neg z) \wedge x]$$

$$x \oplus \texttt{True} \rightsquigarrow x \oplus (z \vee \neg z) \rightsquigarrow [x \wedge (z \vee \neg z)] \vee [x \wedge (z \vee \neg z)]$$

2. Est-ce que les formules dans la colonne p_2 sont dans l'ensemble $\mathcal{F}$? Pourquoi ?
Correction: Non, la formule $(z \wedge \neg z) \oplus x$ n'est pas dans $\mathcal{F}$ car elle contient les symboles $\vee, \wedge, \neg$.

3. On considère l'ensemble $\mathcal{G}$ de toutes les formules construites à partir d'un ensemble de variables V et des opérateurs $\{\texttt{True}, \texttt{False}, \neg, \vee, \wedge, \oplus\}$. On considère également la fonction Φ sur les formules de l'ensemble $\mathcal{G}$:

$$\begin{array}{lcl@{\qquad}lcl} \Phi(x) & := & 0 & \Phi(\neg p) & := & \Phi(p) \\ \Phi(\texttt{True}) & := & 1 & \Phi(p \vee q) & := & \Phi(p) + \Phi(q) \\ \Phi(\texttt{False}) & := & 1 & \Phi(p \wedge q) & := & \Phi(p) + \Phi(q) \\ & & & \Phi(p \oplus q) & := & 2 \cdot \Phi(p) + 2 \cdot \Phi(q) + 1 \end{array}$$

(a) Pour chaque séquence de transformations $p_1 \rightsquigarrow p_2 \rightsquigarrow p_3$ du point 1, vérifier que $\Phi(p_1) > \Phi(p_2) > \Phi(p_3)$.
Correction:

$$\begin{array}{lcl@{\qquad}lcl} \Phi(\texttt{False} \oplus x) & = & 3 & \Phi(x \oplus \texttt{True}) & = & 3 \\ \Phi((z \wedge \neg z) \oplus x) & = & 1 & \Phi(x \oplus (z \vee \neg z)) & = & 1 \\ \Phi([(z \wedge \neg z) \wedge \neg x] \vee [\neg(z \wedge \neg z) \wedge x]) & = & 0 & \Phi([x \wedge (z \vee \neg z)] \vee [x \wedge (z \vee \neg z)]) & = & 0 \end{array}$$

(b) Montrer par induction sur les formules de $\mathcal{G}$ que : $p \rightsquigarrow q$ implique $\Phi(p) > \Phi(q)$.
Correction:
- Les trois cas de base :
  - $p = x$ ne se transforme pas donc la propriété est trivialement vraie.

- $p = \texttt{True} \rightsquigarrow (z \vee \neg z) = q$, donc $\Phi(p) = 1 > 0 = \Phi(q)$.
- $p = \texttt{False} \rightsquigarrow (z \wedge \neg z) = q$, donc $\Phi(p) = 1 > 0 = \Phi(q)$.

- Les cas inductifs
  - $p = \neg p_1$.
    HI : si $p_1 \rightsquigarrow q_1$, alors $\Phi(p_1) > \Phi(q_1)$.
    À démontrer : Si $\neg p_1 \rightsquigarrow q$, alors $\Phi(\neg p_1) > \Phi(q)$.
    Si $\neg p_1 \rightsquigarrow q$, alors $q = \neg q_1$ et $p_1 \rightsquigarrow q_1$. On peut appliquer l'HI et on conclut donc ainsi : $\Phi(p) = \Phi(\neg p_1) = \Phi(p_1) >_{H.I} \Phi(q_1) = \Phi(\neg q_1) = \Phi(q)$.
  - $p = p_1 \vee p_2$.
    HI : si $p_i \rightsquigarrow q_i$, alors $\Phi(p_i) > \Phi(q_i)$ $(i = 1, 2)$.
    À démontrer : Si $p_1 \vee p_2 \rightsquigarrow q$, alors $\Phi(p_1 \vee p_2) > \Phi(q)$.
    Maintenant on analyse les deux cas possibles pour q :
    - $p \rightsquigarrow q_1 \vee p_2 = q$, où $p_1 \rightsquigarrow q_1$.
      Alors $\Phi(p_1 \vee p_2) = \Phi(p_1) + \Phi(p_2) >_{HI} \Phi(q_1) + \Phi(p_2) = \Phi(q)$.
    - $p \rightsquigarrow p_1 \vee q_2 = q$, où $p_2 \rightsquigarrow q_2$.
      Alors $\Phi(p_1 \vee p_2) = \Phi(p_1) + \Phi(p_2) >_{HI} \Phi(p_1) + \Phi(q_2) = \Phi(q)$.
  - $p = p_1 \wedge p_2$, raisonnement similaire au cas précédent.
  - $p = p_1 \oplus p_2$.
    HI : si $p_i \rightsquigarrow q_i$, alors $\Phi(p_i) > \Phi(q_i)$ $(i = 1, 2)$.
    À démontrer : Si $p_1 \oplus p_2 \rightsquigarrow q$, alors $\Phi(p_1 \oplus p_2) > \Phi(q)$.
    Maintenant on analyse les trois cas possibles pour q :
    - $p \rightsquigarrow q_1 \oplus p_2 = q$, où $p_1 \rightsquigarrow q_1$.
      Alors $\Phi(p_1 \oplus p_2) = 2 \cdot \Phi(p_1) + 2 \cdot \Phi(p_2) + 1 >_{HI} 2 \cdot \Phi(q_1) + 2 \cdot \Phi(p_2) + 1 = \Phi(q)$.
    - $p \rightsquigarrow p_1 \oplus q_2 = q$, où $p_2 \rightsquigarrow q_2$.
      Alors $\Phi(p_1 \oplus p_2) = 2 \cdot \Phi(p_1) + 2 \cdot \Phi(p_2) + 1 >_{HI} 2 \cdot \Phi(p_1) + 2 \cdot \Phi(q_2) + 1 = \Phi(q)$.
    - $p \rightsquigarrow (p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2) = q$.
      Alors $\Phi(p) = 2 \cdot \Phi(p_1) + 2 \cdot \Phi(p_2) + 1 > 2 \cdot \Phi(p_1) + 2 \cdot \Phi(p_2) = \Phi(q)$.

(c) Comment conclure que toute séquence de transformations $\rightsquigarrow$ à partir d'une formule quelconque de l'ensemble $\mathcal{G}$ est de longueur finie ?

Correction: On remarque d'abord que pour toute formule p dans $\mathcal{G}$ on a $\Phi(p) \geq 0$ (on peut le montrer formellement par induction structurelle). Supposons maintenant qu'il existe une séquence infinie à partir d'une formule p :

$$p \rightsquigarrow p_1 \rightsquigarrow p_2 \dots$$

Par le point (b) et la première remarque nous avons donc une séquence infinie sur les entiers *naturels* :

$$\Phi(p) > \Phi(p_1) > \Phi(p_2) \dots$$

Ce qui donne une contradiction.