

**INFO501 : logique (et informatique)**  
**TD : logique propositionnelle**

Pierre Hyvernat  
 Laboratoire de mathématiques de l'université de Savoie  
 bâtiment Chablais, bureau 17, poste : 94 22  
 email : [Pierre.Hyvernat@univ-smb.fr](mailto:Pierre.Hyvernat@univ-smb.fr)  
 www : <http://www.lama.univ-smb.fr/~hyvernat/>

**Exercice 1 : formules**

*Question 1.* Donnez une formule propositionnelle (en introduisant des formules atomiques adéquates) pour les phrases suivantes.

- "S'il fait beau, je vais travailler en vélo, mais quand je suis en retard, je prend le bus."
- "Si tu ranges ta chambre, tu pourras aller au cinéma."
- "N'oublie pas ton écharpe, il fait froid."
- " $n$  est un multiple de 12 dès que  $n$  est un multiple de 2 et de 3."
- "Si  $n$  est un nombre premier, alors il est impair, sauf si c'est 2."

*Question 2.* Donnez le connecteur principal des formules suivantes, puis donnez l'arbre correspondant à la formule.

$$\begin{array}{cccc}
 A \wedge (B \Rightarrow C) & \neg(A \wedge (B \vee \neg C)) & A \vee B \vee C \vee D & A \Rightarrow B \Rightarrow C \\
 A \wedge B \vee C \wedge D & \neg\neg\neg(B \Leftrightarrow C) & \neg A \wedge B \Rightarrow C & A \Rightarrow B \vee \neg(C \wedge D \Rightarrow E)
 \end{array}$$

**Exercice 2 : connecteurs logiques et tables de vérité**

*Question 1.* Retrouvez la table de vérité du connecteur  $\Rightarrow$  en étudiant les différents cas correspondants à la formule (vraie) "*Si  $n$  est un multiple de 4, alors  $n$  est pair.*".

Vérifiez que cette table de vérité est la même que celle de  $\neg A \vee B$ .

*Question 2.* Calculez les tables de vérité pour vérifier que les 2 formules suivantes sont équivalentes :

- $A \vee (B \wedge C)$ ,
- $(A \vee B) \wedge (A \vee C)$ .

*Question 3.* Donnez une formule équivalente à  $A \wedge B$  en n'utilisant que la disjonction ( $\vee$ ) et la négation ( $\neg$ ).

*Question 4.* En utilisant les équivalences logique standard (lois de de Morgan, distributivité, etc.), montrez que les 2 formules suivantes sont équivalentes :

$$(A \vee B) \wedge (\neg A \vee \neg B) \quad \text{et} \quad (A \wedge \neg B) \vee (\neg A \wedge B)$$

*Question 5.* Le symbole de Sheffer ( $\uparrow$ ) est un connecteur binaire avec la table de vérité suivante :

$\uparrow$	0	1
0	1	1
1	1	0

Il agit comme la négation d'une conjonction : "pas les deux".

À quoi correspond  $A \uparrow A$  ?

Montrez que l'on peut redéfinir tous les opérateurs usuels uniquement à partir de  $\uparrow$ .

[?] *Question 6.* Combien de connecteurs binaires existe-t-il ? Énumérez les et donnez-leur un nom. Cherchez combien ont la propriété de complétude fonctionnelle (comme “↑”), à savoir de pouvoir redéfinir tous les connecteurs usuels, et donc toutes les fonctions booléennes.

*Question 7.* Les formules suivantes sont-elles des tautologies ? Sont-elles satisfiables ?

$$\begin{array}{ll} A \Rightarrow B \Rightarrow A & (A \Rightarrow A) \Rightarrow A \\ (A \vee B) \Rightarrow (A \wedge B) & (A \Rightarrow B) \vee (B \Rightarrow A) \end{array}$$

*Question 8.* Que pensez-vous de l’algorithme “bête” qui calcule la table de vérité d’une formule pour vérifier si elle est satisfiable ?

À partir de combien de variables est-ce que cet algorithme risque de ne plus fonctionner ?

*Question 9.* Justifiez l’assertion :  $\neg F$  est une tautologie si et seulement si  $F$  n’est pas satisfiable.

### Exercice 3 : satisfiabilité, résolution et algorithme DPLL

*Question 1.* Donnez les tables de vérité puis une formule en FNC (“forme normale conjonctive”) pour les formules suivantes à 4 variables :

- $s \Leftrightarrow (a \oplus b \oplus c)$
- $cc \Leftrightarrow ((a \wedge b) \vee (a \wedge c) \vee (b \wedge c))$ .

*Question 2.* Pour chacune des formules suivantes, donnez une formule en FNC équivalente sans passer par les tables de vérité. Donnez également la représentation en Python comme liste de listes d’entiers.

$$x_1 \wedge x_2 \quad x_1 \vee x_2 \quad x_1 \Leftrightarrow x_2 \quad (x_1 \wedge x_2) \vee \neg(x_3 \Rightarrow x_4)$$

*Question 3.* Décrivez une procédure (récursive) qui permet de transformer n’importe quelle formule en formule en FNC logiquement équivalente.

*Question 4.* Utilisez les lois de distribution pour transformer la formule suivante en FNC.

$$(x_1 \wedge y_1) \vee \dots \vee (x_k \wedge y_k)$$

Estimez la taille de la formule obtenue

*Indice :* commencez avec  $k = 2$  et  $k = 3$ .

[?] *Question 5.* Vérifiez que la formule

$$(z_1 \vee \dots \vee z_k) \wedge (\neg z_1 \vee x_1) \wedge (\neg z_1 \vee y_1) \wedge \dots \wedge (\neg z_k \vee x_k) \wedge (\neg z_k \vee y_k)$$

est équisatisfiable avec

$$(x_1 \wedge y_1) \vee \dots \vee (x_k \wedge y_k)$$

*Indice :* chaque variable  $z_i$  “représente” la clause  $(x_i \vee y_i)$ .

En quoi est-ce que cette formule corrige le problème soulevé dans la question précédente ?

*Question 6.* Faites tourner l’algorithme naïf pour SAT sur les formules suivantes :

- $(x_1 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_3)$
- $(x_1 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2)$
- $(\neg x_1 \vee \neg x_3) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_3) \wedge (x_3 \vee x_1) \wedge (x_3 \vee \neg x_2) \wedge (x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_4 \vee \neg x_3)$

À chaque étape l’algorithme utilise le premier littéral de la première clause et le met à “vrai”.

*Question 7.* Appliquez l’algorithme vu en cours, en ajoutant la règle de propagation des clauses unitaires, sur les formules 2 et 3 de la question précédente.

Question 8. Proposez une fonction Python pour repérer les littéraux purs dans une formule en FNC.

Remarque : en pratique, on n'implémente pas la simplification des clauses pures car cela est trop lent.

Question 9. Faites tourner l'algorithme DPLL avec la simplification des clauses unitaires et celles des littéraux purs sur la formule

$$(\neg x_1 \vee x_2) \wedge (x_1 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_2 \vee x_3 \vee \neg x_5) \wedge (\neg x_3 \vee x_4 \vee x_5) \wedge (\neg x_5 \vee \neg x_1)$$

[?] Question 10. On s'intéresse à la transformation qui découpe une clause en 2 de la manière suivante :

$$F = (l_1 \vee \dots \vee l_k \vee \dots \vee l_n) \wedge \dots \mapsto F' = (l_1 \vee \dots \vee l_k \vee x) \wedge (\neg x \vee \dots \vee l_n) \wedge \dots$$

où  $x$  est une nouvelle variable propositionnelle.

- Les deux formules sont elles équivalentes ?
- Les deux formules sont elles équi-satisfiables ?
- En appliquant cette transformation plusieurs fois, peut-on réduire toutes les clauses à 4 littéraux ? À 3 littéraux ? À 2 littéraux ?

[?] Question 11. Donnez un exemple de formule satisfiable où l'algorithme naïf teste un nombre exponentiel de solution avant d'en trouver une correcte.

Faites la même chose pour l'algorithme DPLL.

#### Exercice 4 : déduction naturelle

Rappels : voici les règles de la déduction naturelle pour la logique propositionnelle.

$$\frac{\Gamma; F \vdash G}{\Gamma \vdash F \Rightarrow G} (\Rightarrow_c) \qquad \frac{\Gamma; F \Rightarrow G \vdash F}{\Gamma; F \Rightarrow G \vdash G} (\Rightarrow_h \dagger) / (\text{modus ponens} \dagger)$$

$$\frac{\Gamma \vdash F_1 \quad \Gamma \vdash F_2}{\Gamma \vdash F_1 \wedge F_2} (\wedge_c) \qquad \frac{\Gamma; F_1; F_2 \vdash G}{\Gamma; F_1 \wedge F_2 \vdash G} (\wedge_h)$$

$$\frac{\Gamma \vdash F_i}{\Gamma \vdash F_1 \vee F_2} (\vee_{c,i} \dagger) \qquad \frac{\text{????????????} (\vee_h)}{\Gamma; F_1 \vee F_2 \vdash G} / (\text{raisonnement par cas})$$

$$\frac{\Gamma; F \vdash \perp}{\Gamma \vdash \neg F} (\neg_c) \qquad \frac{\Gamma \vdash \neg F \quad \Gamma \vdash F}{\Gamma \vdash \perp} (\text{contradiction} \dagger)$$

$$\frac{}{\Gamma; F \vdash F} (\text{axiome}) \qquad \frac{\Gamma; \neg F \vdash \perp}{\Gamma \vdash F} (\text{raisonnement par l'absurde})$$

$$\frac{\Gamma \vdash G \quad \Gamma; G \vdash F}{\Gamma \vdash F} (\text{résultat intermédiaire} \dagger)$$

Question 1. Rappelez le sens "intuitif" de la notation " $H_1; H_2; \dots, H_k \vdash F$ " et d'une règle de la forme

$$\frac{\Delta_1 \vdash G_1 \quad \Delta_2 \vdash G_2 \quad \dots}{\Gamma \vdash F}$$

Question 2. Complétez la règle manquante et justifiez son nom "raisonnement par cas".

$$\frac{\text{????????????} (\vee_h)}{\Gamma; F_1 \vee F_2 \vdash G} / (\text{raisonnement par cas})$$

Question 3. Quelles sont les règles qui permettent de démontrer l'absurde (" $\perp$ "), c'est à dire d'obtenir un séquent de la forme  $\Gamma \vdash \perp$  ?

Question 4. Démontrez la loi de distributivité  $(A \vee (B \wedge C)) \Rightarrow (A \vee B) \wedge (A \vee C)$ .

Démontrez ensuite la réciproque  $(A \vee B) \wedge (A \vee C) \Rightarrow (A \vee (B \wedge C))$ .

*Question 5.* Démontrez la loi de “contraposition” :  $(A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A)$ .

Essayez de démontrer sa réciproque :  $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$ . (Il faudra utiliser un raisonnement par l'absurde.)

*Question 6.* Démontrez l'équivalence  $A \Leftrightarrow \neg\neg A$  en montrant les 2 implications.

*Indice* : il faudra utiliser un raisonnement par l'absurde pour l'implication  $\neg\neg A \Rightarrow A$ ...

[!] *Question 7.* Démontrez la tautologie  $A \vee \neg A$ .

*Indice* : il faudra utiliser un raisonnement par l'absurde...