

INFO501 : logique (et informatique)
TD 1 : logique propositionnelle

Pierre Hyvernât
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 17, poste : 94 22
email : Pierre.Hyvernât@univ-smb.fr
www : <http://www.lama.univ-smb.fr/~hyvernât/>

Exercice 1 : formules

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

- "Si j'ai assez d'argent, je prendrais du fromage et un dessert."
- "Quand il pleut, je prend mon parapluie."
- "Quand il pleut, je prend mon parapluie, et quand il fait froid, je prend mon bonnet."
- "Quand il fait beau, je vais travailler en vélo, mais quand je suis en retard, je prend le bus."
- "Tu pourras aller au cinéma quand tu auras rangé ta chambre."
- "N'oublie pas ton écharpe, il fait froid."
- "Si f est une fonction dérivable, alors c'est aussi une fonction continue."
- "Si n est un nombre premier, alors il est impair, sauf si c'est 2."

Question 2. Donnez le connecteur principal des formules suivantes :

- $A \wedge (B \Rightarrow C)$
- $\neg(A \wedge (B \vee \neg C))$
- $A \vee B \vee C \vee D$
- $A \Rightarrow B \Rightarrow C$
- $A \wedge B \vee C \wedge D$
- $\neg\neg\neg(B \Leftrightarrow C)$
- $\neg A \wedge B \Rightarrow C$
- $A \Rightarrow B \vee \neg(C \wedge D \Rightarrow E)$

Question 3. Dessinez l'arbre correspondant à chaque formule de la question précédente.

Exercice 2 : connecteurs logiques et tables de vérité

Question 1. Rappelez les tables de vérité des connecteurs \wedge , \vee , \Leftrightarrow et \neg .

Question 2. 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 3. 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 4. La table de vérité du XOR (noté \oplus) est

\oplus	0	1
0	0	1
1	1	0

Donnez une formule ayant la même table de vérité que $A \oplus B$ en utilisant les connecteurs usuels. N'oubliez pas de vérifier...

Question 5. En mathématique, on dit en général “ A ssi B ssi C ” pour dire que les trois formules A , B et C sont équivalentes.

Peut-on mettre des parenthèses dans $A \Leftrightarrow B \Leftrightarrow C$ pour avoir cette interprétation ?

Sinon, comment peut-on traduire “ A ssi B ssi C ” en logique propositionnelle ?

Question 6. Donnez une formule équivalente à $A \wedge B$ en n'utilisant que la disjonction (\vee) et la négation (\neg).

Question 7. En utilisant les équivalences logiques standard (lois 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 8. 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 9. Combien de connecteurs binaires existe-t-il ? Énumérez les et donnez-leur un nom. Cherchez combien ont la propriété de complétude fonctionnelle (comme “ \uparrow ”), à savoir de pouvoir redéfinir tous les connecteurs usuels, et donc toutes les fonctions booléennes.

Question 10. Rappelez les définitions de “tautologie” et de “formule satisfiable”.

Question 11. Les formules suivantes sont-elles des tautologies ? Sont-elles satisfiables ?

- $A \vee \neg A$
- $(A \Rightarrow A) \Rightarrow A$
- $(A \vee B) \Rightarrow (A \wedge B)$
- $(A \Rightarrow B) \vee (B \Rightarrow A)$

Question 12. Vérifiez que la “loi de Peirce” $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ est une tautologie en calculant sa table de vérité.

Question 13. Au pire cas, combien faut-il calculer de lignes d'une table de vérité pour vérifier si une formule est une tautologie ?

Et au meilleur cas ?

Et pour vérifier si une formule est satisfiable ?

Question 14. 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 15. Justifiez l'assertion : $\neg F$ est une tautologie si et seulement si F n'est pas satisfiable.

Exercice 3 : déduction naturelle

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

$$\frac{\Gamma, F \vdash G}{\Gamma \vdash F \Rightarrow G} (\Rightarrow_c^*) \quad \frac{\Gamma \vdash F \Rightarrow G \quad \Gamma \vdash F}{\Gamma \vdash G} (\text{modus ponens})$$

$$\frac{\Gamma \vdash F_1 \quad \Gamma \vdash F_2}{\Gamma \vdash F_1 \wedge F_2} (\wedge_c^*) \quad \frac{\Gamma \vdash F_1 \wedge F_2}{\Gamma \vdash F_i} (\wedge_i)$$

$$\frac{\Gamma \vdash F_i}{\Gamma \vdash F_1 \vee F_2} (\vee_{c,i}) \quad \frac{\Gamma \vdash F_1 \vee F_2 \quad ????????????}{\Gamma \vdash G} \text{ (raisonnement par cas)}$$

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

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

Les règles avec une étoile “*” sont celle que l’on peut toujours appliquer sans risque de rendre la preuve impossible.

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

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

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

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

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 démontrez sa réciproque : $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$. (C’est plus difficile !)

Question 6. Démontrez la “loi de Peirce” $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$.

Indice : il faudra utiliser un raisonnement par l’absurde...

Question 7. 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 8. Difficile : démontrez la tautologie $A \vee \neg A$.

Indice : il faudra utiliser un raisonnement par l’absurde...

Question 9. On pourrait rajouter de nombreuses règles de démonstration, mais ce n’est pas nécessaire car on peut retrouver ces règles à partir des règles de bases.

Montrer par exemple que les règles suivantes sont dérivables :

$$\frac{\Gamma \vdash T \quad \Gamma; T \vdash C}{\Gamma \vdash C} \text{ (coupure)} \quad \frac{\Gamma; A; B \vdash C}{\Gamma; A \wedge B \vdash C}$$

Exercice 4 : satisfiabilité, résolution et algorithme DPLL

Question 1. Donnez une formule en FNC (“forme normale conjonctive”) pour les tables de vérité suivantes

A	B	F
0	0	1
0	1	1
1	0	0
1	1	1

A	B	C	G
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Question 2. Donnez les tables de vérité puis une formule en FNC pour les formules

- $S \Leftrightarrow (A \oplus B \oplus C)$,
- $C' \Leftrightarrow ((A \wedge B) \vee (A \wedge C) \vee (B \wedge C))$.

Question 3. Pour chacune des formules suivantes, donnez une formule en FNC équivalente

- $A \wedge B$
- $A \vee B$
- $A \Rightarrow B$
- $\neg(A \Rightarrow B)$
- $(A \wedge B) \vee \neg(C \Rightarrow D)$

Question 4. Décrivez une procédure (récursive) qui permet de transformer n'importe quelle formule en formule en FNC logiquement équivalente.

Question 5. Estimez la taille minimale d'une formule en FNC logiquement équivalente à

$$(A_1 \wedge B_1) \vee \dots \vee (A_k \wedge B_k)$$

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

Question 6. Vérifiez que la formule

$$(C_1 \vee \dots \vee C_k) \wedge (\neg C_1 \vee A_1) \wedge (\neg C_1 \vee B_1) \wedge \dots \wedge (\neg C_k \vee A_k) \wedge (\neg C_k \vee B_k)$$

est équivalent à la formule de la question précédente.

Indice : chaque variable C_i "représente" la clause $(A_i \vee B_i)$.

Question 7. On regarde la transformation qui réduit le nombre de littéraux des clauses de 4 à 3, en ajoutant des clauses.

$$F = (l_1 \vee l_2 \vee l_3 \vee l_4) \wedge \dots \mapsto F' = (l_1 \vee l_2 \vee x_2) \wedge (\neg x_2 \vee l_3 \vee l_4) \wedge \dots$$

où x_2 est une nouvelle variable propositionnelle.

- Les deux formules sont-elles équivalentes ?
- Les deux formules sont-elles équivalentes-satisfiables ?
- Généralisez la transformation pour réduire une clause arbitraire à plusieurs clauses de 3 littéraux.

Question 8. Faites tourner l'algorithme naïf pour SAT sur les formules suivantes :

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

Cet algorithme instancie les variables dans l'ordre (x_1 , puis x_2 , etc.) et essaie en premier la valeur 1.

Question 9. Donnez un exemple de formule où l'algorithme naïf est exponentiel car seule la dernière solution testée est valide.

Question 10. Appliquez les règles de propagation des clauses unitaires et de simplification des littéraux purs sur les formules suivantes.

Le résultat est-il satisfiable ?

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

Question 11. En pratique, on n'implémente pas la simplification des clauses pures car cela prend trop de temps.

Faites tourner l'algorithme DPLL en utilisant uniquement la simplification des clauses unitaires sur les formules de la question 8.