

DS 2

Option informatique, deuxième année

Julien REICHERT

Partie 1 : Logique et calcul des propositions

Nous nous intéressons dans cet exercice à l'étude de quelques propriétés de la logique propositionnelle tri-valuée. En plus des deux valeurs classiques VRAI (\top) et FAUX (\perp) que peut prendre une expression, la logique propositionnelle tri-valuée introduit une troisième valeur INDETERMINE (?). \mathcal{V} est l'ensemble des variables propositionnelles et \mathcal{F} l'ensemble des formules construites sur \mathcal{V} . Pour $A, B \in \mathcal{V}$, les tables de vérité des opérateurs classiques dans cette logique propositionnelle sont les suivantes :

A	B	$A \wedge B$
\top	\top	\top
\top	\perp	\perp
\top	?	?
\perp	\top	\perp
\perp	\perp	\perp
\perp	?	\perp
?	\top	?
?	\perp	\perp
?	?	?

A	B	$A \vee B$
\top	\top	\top
\top	\perp	\top
\top	?	\top
\perp	\top	\top
\perp	\perp	\perp
\perp	?	?
?	\top	\top
?	\perp	?
?	?	?

A	B	$A \Rightarrow B$
\top	\top	\top
\top	\perp	\perp
\top	?	?
\perp	\top	\top
\perp	\perp	\top
\perp	?	\top
?	\top	\top
?	\perp	?
?	?	\top

A	$\neg A$
\top	\perp
\perp	\top
?	?

On appelle tri-valuation une fonction $f : \mathcal{V} \rightarrow \{\top, \perp, ?\}$, qu'on étend à une fonction \hat{f} définie sur \mathcal{F} . Elle satisfait une formule ϕ si $\hat{f}(\phi) = \top$. Une formule ϕ est une conséquence d'un ensemble de formule X si toute interprétation qui satisfait toutes les formules de X satisfait ϕ , et une tautologie si toute interprétation la satisfait. On pose $\top = 1$, $\perp = 0$ et $? = 0,5$.

Question 1 : Montrer que $A \vee \neg A$ n'est pas une tautologie.

Question 2 : Proposer une tautologie simple.

Question 3 : Proposer un calcul simple permettant de trouver la table de vérité de $A \wedge B$ en fonction de A et B . Même question pour $A \vee B$.

Question 4 : En logique bi-valuée classique, les propositions $\neg A \vee B$ et $A \Rightarrow B$ sont équivalentes. Qu'en est-il dans le cadre de la logique propositionnelle tri-valuée ?

Question 5 : En écrivant les tables de vérité, indiquer si les propositions $\neg B \Rightarrow \neg A$ et $A \Rightarrow B$ sont équivalentes.

Question 6 : Donner la table de vérité de la proposition $((A \Rightarrow B) \wedge ((\neg A) \Rightarrow B)) \Rightarrow B$. Cette proposition est-elle une tautologie ?

Un nouvel opérateur d'implication, noté \rightarrow , est alors défini, dont la table de vérité est la suivante :

A	B	$A \rightarrow B$
\top	\top	\top
\top	\perp	\perp
\top	?	?
\perp	\top	\top
\perp	\perp	\top
\perp	?	\top
?	\top	\top
?	\perp	?
?	?	?

Question 7 : $A \rightarrow A$ est-elle une tautologie ?

Question 8 : A-t-on équivalence entre le fait que B soit une conséquence de $\{A\}$ et le fait que $A \rightarrow B$ soit une tautologie ?

On définit un type `FormuleLogique` représentant les formules de la manière suivante :

```

type FormuleLogique
|Vrai (* Constante Vrai*)
|Faux (* Constante Faux*)
|Indétermine (* Constante Indeterminé*)
|Var of string (* Variable propositionnelle*)
|Non of FormuleLogique (* Négation d'une formule*)
|Et of FormuleLogique*FormuleLogique (* conjonction de deux formules*)
|Ou of FormuleLogique*FormuleLogique (* disjonction de deux formules*)
|Implique of FormuleLogique*FormuleLogique (* implication*)

```

Question 9 : Avec la représentation précédente, écrire en CaML la formule : $((A \Rightarrow B) \wedge ((\neg A) \Rightarrow B)) \Rightarrow B^1$.

1. Au passage, commenter la formule en bon mathématicien.

Partie 2 : Arbres binaires et occurrences

Il est possible de représenter un arbre binaire sous forme d'occurrences. Dans ce cas, si un nœud a pour occurrence μ , alors son fils gauche aura pour occurrence $\mu 0$ et son fils droit $\mu 1$.

Nous nous proposons donc de représenter un arbre binaire à l'aide d'une chaîne de caractères constituée des diverses occurrences de l'arbre séparées les unes des autres par une virgule. Nous représenterons la racine par E (l'occurrence vide).

Soit l'arbre B défini, sous forme d'occurrences et représenté à l'aide d'une chaîne de caractères de la manière suivante : $B = "E,0,1,00,01,10,11,001,101,1010,1011"$

Question 1 : Comment, sans représenter l'arbre B , peut-on déterminer sa taille, sa hauteur, le fait qu'un de ses nœuds soit une feuille, la somme des hauteurs de ses nœuds et la hauteur moyenne de ses nœuds ?

Question 2 : Représenter graphiquement l'arbre B .

Partie 3 : Chemin impossible

Soit un arbre binaire de recherche contenant des caractères. On désire chercher la valeur G . Parmi les séquences suivantes, lesquelles ne pourraient pas être la suite des nœuds parcourus ? Indiquer sur votre copie les numéros des séquences impossibles et justifiez-les.

- 1) S - A - Q - C - M - G
- 2) Q - A - K - C - O - G
- 3) A - C - K - I - E - G
- 4) C - K - I - A - E - G

Partie 4 : Les deux premières questions du sujet X-ENS de l'an dernier

Question 1 : Pour chaque formule qui suit, dire si elle est satisfiable ou non, en justifiant² :

- a) $x_1 \wedge (x_0 \vee \neg x_0) \wedge \neg x_1$
- b) $(x_0 \vee \neg x_1) \wedge (\neg x_0 \vee x_2) \wedge (x_1 \vee \neg x_2)$
- c) $x_0 \wedge \neg(x_0 \wedge \neg(x_1 \wedge \neg(x_1 \wedge \neg x_2)))$
- d) $(x_0 \vee x_1) \wedge (\neg x_0 \vee x_1) \wedge (x_0 \vee \neg x_1) \wedge (\neg x_0 \vee \neg x_1)$

En machine nous représenterons les formes normales conjonctives (FNC) sous la forme de listes de listes. Plus précisément, une FNC sera une liste de clauses et chaque clause sera une liste de littéraux :

```
type littéral =  
|V of int (* variable *)  
|NV of int;; (* négation de variable *)  
type clause == littéral list;;  
type fnc == clause list;;
```

Question 2 : Écrire une fonction `var_max` qui prend en entrée une FNC `f` et renvoie le plus grand indice de variable utilisé dans la formule. La complexité de la fonction doit être linéaire en la taille de `f`.

2. Dans le sujet de l'ENS, c'était sans justification...