OPTION INFORMATIQUE
Devoir surveillé 3

Toutes les fonctions sont à écrire en OCaml. On attachera le plus grand soin à leur lisibilité. On pourra toujours librement utiliser une fonction demandée à une question précédente, même si cette question n'a pas été traitée.

1 Cherchez les Clés du Paradis (d'après CCP 2018)

1.1 Préparation

Dans la suite on note $\mathbb{B} = \{F, V\}$ l'ensemble des booléens, avec F désignant le faux et V le vrai.
On utilisera la notation $\overline{\phi}$ pour désigner $\neg\phi$, où ϕ est une formule.
On introduit dans la syntaxe du calcul propositionnel les constantes $\perp$ et $\top$, $\perp$ étant toujours interprétée par F, et $\top$ par V.

1. Pour A une variable propositionnelle, simplifier les formules :
   $A \vee \perp \quad A \vee \top \quad A \wedge \perp \quad A \wedge \top \quad A \vee \overline{A} \quad A \wedge \overline{A} \quad A \to \perp \quad A \to \top \quad \perp \to A \quad \top \to A$

On introduit également le constructeur binaire $\oplus$ (lire xor), dont la sémantique est définie ainsi : une valuation v satisfait $\phi \oplus \psi$ si et seulement si v satisfait exactement une formule parmi ϕ et ψ. On considérera dans la suite que c'est ce connecteur logique que désigne un « ou bien » en français.

1. Pour A, B deux variables propositionnelles, donner une forme normale conjonctive de $A \oplus B$, puis une forme normale disjonctive.

1.2 Les épreuves

Vous avez été sélectionné(e) pour participer au jeu "Cherchez les Clés du Paradis (CCP)".

Première épreuve

Jean-Pierre Pendule, le célèbre animateur, vous accueille pour la première épreuve. Il vous explique la règle du jeu. Devant vous, deux boîtes et sur chacune d'entre elles une inscription. Chacune des boîtes contient soit une clé verte, soit une clé rouge. Vous devez choisir l'une des boîtes : si le résultat est une clé rouge, alors vous quittez le jeu, si c'est une clé verte vous remportez l'épreuve. Jean-Pierre Pendule dévoile les inscriptions sur chacune des boîtes et vous affirme qu'elles sont soit vraies toutes les deux, soit fausses toutes les deux :

- sur la boîte 1, il est écrit "Une au moins des deux boîtes contient une clé verte" ;
- sur la boîte 2, il est écrit "Il y a une clé rouge dans l'autre boîte".

Dans toute cette partie, on note P_i la proposition affirmant qu'il y a une clé verte dans la boîte i.

1. Donner une formule représentant la phrase écrite sur la boîte 1. Donner une seconde formule pour la boîte 2.
2. Donner une formule représentant l'affirmation de l'animateur. Simplifier cette formule de sorte à n'obtenir qu'une seule occurrence de chaque P_i.
3. Quel choix devez-vous faire pour continuer le jeu à coup sûr ?

Deuxième épreuve

Bravo, vous avez obtenu la première clé verte, et parvenez à la deuxième épreuve. Avec les mêmes règles du jeu, l'animateur vous propose alors deux nouvelles boîtes portant les inscriptions suivantes :

- sur la boîte 1, il est écrit "Il y a une clé rouge dans cette boîte, ou bien il y a une clé verte dans la boîte 2" ;
- sur la boîte 2, il est écrit "Il y a une clé verte dans la boîte 1".

1. Donner une formule de la logique des propositions pour chaque affirmation.
2. Dresser la table de vérité de ces formules et en déduire l'ensemble des valuations satisfaisant les règles du jeu. En déduire la boîte que vous devez choisir.

Troisième épreuve

Sous les acclamations du public, vous parvenez à l'épreuve finale. Jean-Pierre Pendule sourit et vous demande simplement :

1. Montrer que $\{\oplus, \rightarrow\}$ est un système complet de connecteurs.

2 Implémentation en Caml

On représente les formules du calcul propositionnel (sans les constantes $\top$ et $\bot$) par le type Caml suivant :

```
type 'a formule =
      V of 'a
    | Neg of 'a formule
    | Et of 'a formule * 'a formule
    | Ou of 'a formule * 'a formule
    | Imp of 'a formule * 'a formule
    | Eq of 'a formule * 'a formule;;
```

Le type `'a` correspond au type des variables apparaissant dans les formules.
Par exemple, la formule $A \vee \neg A$ est représentée dans ce type par le terme `Ou(V('A'), Neg(V('A')))`.

1. Représenter la formule $A \rightarrow ((A \rightarrow B) \rightarrow B)$ en Caml.
2. Écrire une fonction `interpretation : 'a formule -> ('a -> bool) -> bool` prenant en argument une formule et une valuation (ie une fonction qui à chaque variable associe un booléen) et qui renvoie la valeur de vérité de cette formule sous cette valuation.
3. Rappeler la définition d'une *clause*, puis d'une *forme normale conjonctive*.
4. Écrire une fonction `est_une_clause` prenant en argument une formule et testant si c'est une clause.
5. Écrire une fonction `est_une_fnc` prenant en argument une formule et testant si c'est une forme normale conjonctive.
6. Déterminer la complexité des deux fonctions précédentes.
7. Mettre la formule $(A \rightarrow B) \leftrightarrow (B \rightarrow C)$ en forme normale conjonctive.
8. Écrire une fonction `substitution` prenant en argument une formule `phi`, un nom de variable `a` et une formule `psi`, et renvoyant `phi[psi/a]`, ie la formule obtenue en substituant chaque occurrence de la variable `a` par `psi` dans `phi`. Donner la complexité de cette fonction.
9. On dit qu'un graphe $G = (S, A)$ est **biparti** si son ensemble de sommets S admet une partition (U, V) telle qu'il n'existe aucune arête entre deux sommets de U, ou entre deux sommets de V.
   (a) On considère que la variable propositionnelle `V(s)` code "le sommet s appartient à U". Pour un arc $(s1, s2)$, donner une formule codant la propriété respectée par cette arête dans un graphe biparti.
   (b) En déduire une fonction `formule_graphe_biparti : int list array -> int formule` prenant en argument un graphe `g` représenté par un tableau de listes d'adjacences, et renvoyant une formule satisfiable si et seulement si `g` est biparti.