

L2/LD – TP 6 – Tables de vérité

Ce TP doit être fini avant la prochaine séance de TP, éventuellement sur votre temps de travail personnel.

Vous devez rendre sous la forme d'un unique fichier source compressé au format .zip avant 23h30 la veille de votre prochain TP l'ensemble des fonctions demandées dans ce TP

1 Introduction

L'objectif de ce TP est de construire toute la table de vérité d'une formule propositionnelle, c'est-à-dire l'interpréter avec toutes les valuations possibles de ses propositions atomiques.

Par exemple, la table de vérité de la formule propositionnelle $f = P \wedge (Faux \vee (Vrai \wedge Q))$ est la suivante :

P	Q	$P \wedge (Faux \vee (Vrai \wedge Q))$
false	false	false
false	true	false
true	false	false
true	true	true

Chaque ligne de cette table est l'interprétation de f pour une valuation des propositions atomiques de la formule, qui est affichée avant la double barre verticale.

Pour faire ce tp, nous allons utiliser les fonctions, formules propositionnelles et valuations définies lors du tp précédent.

2 Liste des propositions atomiques

La première étape pour déterminer la table de vérité d'une formule propositionnelle consiste à déterminer la liste des propositions atomiques contenues dans cette formule.

La fonction `List.mem` est une fonction prédéfinie qui prend en argument un élément puis une liste et retourne `true` si l'élément appartient à la liste, `false` sinon :

À taper : `List.mem 2 [3;2];;`

Réponse : `- : bool = true`

À taper : `List.mem 5 [3;2];;`

Réponse : `- : bool = false`

1. Proposer une fonction `listePropositionsAtomiques` qui prend en argument une formule propositionnelle (`pf`) et retourne la liste de ses propositions atomiques sans doublon (`ap list`).

Vous pourrez commencer par définir une fonction `listeDePropositionsAtomiques` qui prend en argument une formule propositionnelle et une liste de propositions atomiques sans doublon et retourne la liste complétée des propositions atomiques de la formule.

Vérifier que vous obtenez :

```
À taper : let l=listePropositionsAtomiques f;;
Réponse : val l : ap list = ["P"; "Q"]
```

2. Pour composer la première ligne de la table de vérité, on commence par afficher les propositions atomiques de la formule. Proposer une fonction `affichePremiereLignedebut` qui prend en argument la liste des propositions atomiques d'une formule et écrit le début de la première ligne de la table de vérité :

```
À taper : affichePremiereLignedebut l;;
Réponse : P    /Q    //- : unit = ()
```

3. En utilisant la fonction précédente, proposer une fonction qui prend en argument une formule propositionnelle et affiche la première ligne de sa table de vérité :

```
À taper : affichePremiereLigne f;;
Réponse : P    /Q    //P ^ (Faux v (Vrai ^ Q))- : unit = ()
À taper : affichePremiereLigne f3;;
Réponse : R    /P    //Faux v ((R ^ Vrai) ^ P)- : unit = ()
```

3 Une ligne de la table de vérité

Proposer une fonction `afficheValuation` qui prend en argument une formule, sa liste de propositions atomiques et une valuation et qui affiche la ligne de la table de vérité de cette formule correspondant à cette valuation. On pourra définir une fonction auxiliaire qui affiche la valeur de chacun des atomes d'une liste pour une valuation donnée.

```
À taper : afficheLigne f l valuation1;;
Réponse : true   /false   //false- : unit = ()
À taper : afficheLigne f2 l valuation2;;
Réponse : true   /true    //false- : unit = ()
```

4 Affichage de toutes les valuations

La fonction

```
listeValuations : ap list -> (ap -> bool) list
```

du fichier `tp6.ml` retourne, dans une liste, l'ensemble des valuations possibles d'une liste de propositions atomiques donnée. Chacune de ces valuations est une fonction analogue, sur la forme, à la fonction `valuation1`.

```
À taper : let listeval = listeValuations l;;
Réponse : val listeval : (ap -> bool) list = [<fun>; <fun>; <fun>; <fun>]
```

Définir une fonction `afficheAutresLignes` qui prend en argument une formule, sa liste de propositions atomiques et la liste des valuations correspondant et qui affiche toutes les lignes de la table de vérité correspondant à ces valuations.

On doit obtenir :

```
À taper : afficheAutresLignes f l listeval;;
Réponse : false    /false    //false
           false    /true     //false
           true     /false    //false
           true     /true     //true
           - : unit = ()
```

5 Affichage de toutes les valuations

1. Définir une fonction

```
afficheTableVerite : pf -> unit
```

qui affiche la table de vérité d'une formule propositionnelle.

```
À taper : afficheTableVerite f;;
Réponse : P    /Q    //P ^ (Faux v (Vrai ^ Q))
           false  /false  //false
           false  /true   //false
           true   /false  //false
           true   /true   //true
           - : unit = ()
```

2. Construire sur papier les tables de vérité des formules `f`, `f2` et `f3` et vérifier que l'on obtient bien les bons résultats lorsqu'on applique la fonction `afficheTableVerite` à ces formules propositionnelles.