Travaux pratiques

option informatique

Tautologies propositionnelles

Rappelons que la logique des propositions est définie par la donnée :

- de *constantes* Faux et Vrai ;
- de *variables* en nombre dénombrable ;
- d'un opérateur unaire de négation $\neg$;
- de trois opérateurs binaires de conjonction ($\wedge$), de disjonction ($\vee$) et d'implication ($\Rightarrow$) ;

avec la règle suivante :

si f_1 et f_2 sont deux propositions alors $\neg f_1$, $f_1 \wedge f_2$, $f_1 \vee f_2$ et $f_1 \Rightarrow f_2$ sont des propositions.

Les propositions sont représentées en Caml par le type :

```
type proposition = Const of bool
                 | Var of char
                 | Neg of proposition
                 | Et of proposition * proposition
                 | Ou of proposition * proposition
                 | Imp of proposition * proposition ;;
```

On sait que la logique propositionnelle est décidable : on peut par exemple examiner les tables de vérité des propositions. On se propose ici d'implémenter un autre algorithme de décision, qui donne de plus une réfutation lorsque la proposition n'est pas valide.

IF-expressions

Pour cela, on introduit la notion de IF-expression : une IF-expression est soit une proposition atomique (c'est-à-dire une constante ou une variable) soit une proposition de la forme :

$$\text{if } m \text{ then } p \text{ else } q$$

où m, p et q sont des IF-expressions. Comme on s'en doute, la valeur d'une IF-expression est la valeur de p si m est vrai et la valeur de q sinon.

On représente les IF-expressions par le type suivant :

```
type ifexpression = Cst of bool
                  | Vr of char
                  | If of ifexpression * ifexpression * ifexpression ;;
```

Toute proposition est équivalente à une IF-expression. En effet, on dispose des équivalences suivantes :

$$\begin{aligned}
\neg p &\iff \text{if } p \text{ then Faux else Vrai} \\
p \wedge q &\iff \text{if } p \text{ then } q \text{ else Faux} \\
p \vee q &\iff \text{if } p \text{ then Vrai else } q \\
p \Rightarrow q &\iff \text{if } p \text{ then } q \text{ else Vrai}
\end{aligned}$$

Question 1. Écrire une fonction `prop_to_if` qui transforme une proposition en une IF-expression équivalente.

```
prop_to_if : proposition -> ifexpression
```

En déduire des IF-expressions équivalentes aux trois propositions suivantes :

$$p_1 = (a \wedge b) \Rightarrow a, \qquad p_2 = a \Rightarrow (a \wedge b), \qquad p_3 = (a \Rightarrow (a \Rightarrow b)) \Rightarrow b$$

On dit qu'une IF-expression est *normale* si elle est soit atomique, soit de la forme

$$\text{if } s \text{ then } p \text{ else } q$$

où s est une variable et p et q deux IF-expressions normales.

Question 2. Écrire une fonction **`est_normale`** qui détermine si une IF-expression est normale.

```
est_normale : ifexpression -> bool
```

Toute IF-expression est équivalente à une IF-expression normale. Pour le voir il suffit de constater que :

$$\begin{aligned}
\text{if Vrai then } p \text{ else } q &\iff p \\
\text{if Faux then } p \text{ else } q &\iff q \\
\text{if (if } m \text{ then } p \text{ else } q) \text{ then } r \text{ else } s &\iff \text{if } m \text{ then (if } p \text{ then } r \text{ else } s) \text{ else (if } q \text{ then } r \text{ else } s)
\end{aligned}$$

Question 3. Écrire une fonction **`normalise`** qui transforme toute IF-expression en une IF-expression normale équivalente.

```
normalise : ifexpression -> ifexpression
```

Donner les IF-expressions normales correspondant à p_1, p_2 et p_3.

Décision sur les IF-expressions

On appelle *assignation partielle* une fonction des propositions atomiques dans les valeurs de vérité dont le domaine est fini. On représentera une assignation partielle par une liste d'association :

```
type assignation == (char * bool) list ;;
```

On cherche à décider si une IF-expression est une tautologie ou si au contraire elle est réfutable. On définit donc le type suivant :

```
type resultat = Tautologie | Refutation of assignation ;;
```

Pour décider si une IF-expression est ou n'est pas une tautologie, on définit plus généralement la notion d'*être une tautologie par rapport à une assignation partielle* α de la manière suivante :

> une IF-expression est une tautologie par rapport à α si elle est vraie pour toute les assignations qui coïncident avec α sur son domaine.

Ainsi, une IF-expression est une tautologie si elle est une tautologie par rapport à l'assignation partielle de domaine vide (la liste vide).

Pour décider si une IF-expression normale m est une tautologie par rapport à α, on procède de la manière suivante :

- si m est Vrai ou Faux, le résultat est immédiat ;
- si m est une variable s, deux cas se présentent : soit $\alpha(s)$ est défini et alors m est une tautologie par rapport à α si et seulement si $\alpha(s)$ est vrai ; soit $\alpha(s)$ n'est pas défini et on a alors une réfutation en étendant α en α' par $\alpha'(s) = \text{Faux}$;
- enfin, si m est une IF-expression normale (if s then p else q) deux cas se présentent : soit $\alpha(s)$ est défini et vaut Vrai (respectivement Faux) et alors m est une tautologie par rapport à α si et seulement si p (respectivement q) est une tautologie par rapport à α ; soit $\alpha(s)$ n'est pas défini et on définit α_T (respectivement α_F) comme l'expression de α avec Vrai (respectivement Faux) assigné à s. Alors m est une tautologie par rapport à α si et seulement si p est une tautologie par rapport à α_T et q une tautologie par rapport à α_F.

Question 4. Écrire une fonction **`decision_partielle`** qui décide si une IF-expression normale est une tautologie par rapport à une assignation ou au contraire si elle est réfutable (et dans ce cas donnant une réfutation).

```
decision_partielle : assignation -> ifexpression -> resultat
```

Question 5. En déduire une fonction **`decision`** qui décide si une proposition est une tautologie ou au contraire si elle est réfutable.

```
decision : proposition -> resultat
```

Appliquer cette procédure de décision aux trois propositions p_1, p_2 et p_3.