3 Langage logique, Modélisation

Exercice 3.3 *Formalisation logique*

On modélise un monde dans lequel vivent des dragons. On utilisera les symboles de prédicats suivants :

— $\mathtt{B}(x), \mathtt{H}(x), \mathtt{V}(x)$: le dragon x est bleu, est heureux, vole

— $\mathtt{P}(x, y)$: le dragon x est parent du dragon y (ou encore le dragon y est un enfant du dragon x)

— $x = y$: les dragons x et y sont égaux

1. Donner les formules logiques correspondant aux phrases suivantes :
   (a) Tous les dragons bleus volent/ Il existe un dragon bleu qui vole
   (b) Tout dragon a exactement deux parents
   (c) Un dragon dont tous les enfants peuvent voler est heureux
   (d) Tout dragon qui a au moins un parent bleu est lui-même bleu
   (e) Il n'y a pas de dragon heureux qui ne vole pas
2. Traduire en français (ou anglais) les formules logiques suivantes :
   (a) $\forall x\, y\, z, \mathtt{V}(x) \Rightarrow \mathtt{P}(x, y) \Rightarrow \mathtt{P}(x, z) \Rightarrow y = z$
   (b) $\forall x, \exists y, \mathtt{P}(x, y) \wedge \mathtt{H}(y)$
   (c) $\exists y, \forall x, \mathtt{P}(x, y) \wedge \mathtt{H}(y)$

4 Interprétations, Validité, Satisfiabilité

Exercice 4.1 *Logique propositionnelle.*

Soit la formule A définie comme $((p \Rightarrow q \Rightarrow r) \Rightarrow r) \Rightarrow p \wedge q$.

1. Représenter la formule A sous forme d'arbre.
2. Donner la table de vérité de la formule A.
3. Dire si la formule est valide, satisfiable, insatisfiable ?
4. La formule $\neg A$ a-t-elle un modèle ? si oui donner un exemple.

Exercice 4.2 *Algorithmes satisfiabilité-validité*

— On suppose que l'on dispose d'un algorithme `valide` qui répond vrai si une formule est valide et faux sinon. Utiliser cette fonction pour implémenter un algorithme `satisfiable` qui répond vrai si une formule est satisfiable et faux sinon.

— On suppose que l'on dispose d'un algorithme `satisfiable` qui répond vrai si une formule est satisfiable et faux sinon. Utiliser cette fonction pour implémenter un algorithme `valide` qui répond vrai si une formule est valide et faux sinon.

Exercice 4.3 *Calcul booléen et programmation*

p, q et r sont 3 variables propositionnelles. Soit le pseudo-programme :

```
if (p⇒q) then print "a"
else if (q∨r) then print "b"
else if (¬p) then print "c"
else print "d"
```

1. Sachant que p, q et r peuvent prendre n'importe quelle valeur vrai ou faux, combien y-a-t-il d'entrées possibles pour ce programme ?
2. Quelles valeurs parmi "a", "b", "c" et "d" le programme peut-il afficher ?
3. Soit un langage dans lequel un programme P est une instruction élémentaire (comme `print "b"`) ou une conditionnelle **if** C **then** P_1 **else** P_2 avec C une condition logique et P_1 et P_2 des programmes dans le même langage. On dispose d'une fonction `satisfiable` qui pour une condition C renvoie vrai ssi C est satisfiable. Construire une fonction `nodeadcode` qui pour un programme P renvoie vrai s'il existe des entrées qui permettent d'exécuter chaque instruction élémentaire de P (c'est à dire qu'il n'y a pas de « code mort » dans P).
4. (bonus) Le problème de savoir si toutes les instructions élémentaires d'un programme peuvent être exécutées est-il décidable en général ?

Exercice 4.4 *Interprétation en calcul des prédicats*

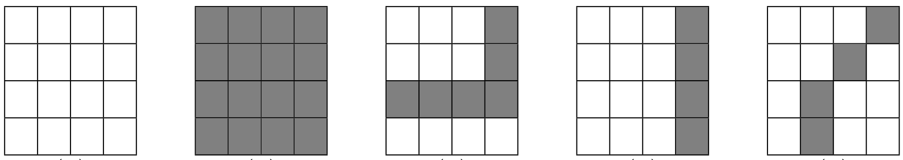
Pour connaître la valeur de vérité d'une formule du calcul des prédicats, il faut choisir un domaine et fixer une interprétation dans ce domaine des symboles de fonction et de prédicats du langage.

On considère un langage avec juste un symbole de prédicat binaire P.

On choisit une interprétation sur un domaine avec 4 éléments $\{1, 2, 3, 4\}$. On représente l'interprétation de P par un tableau 4×4. La formule atomique $P(t, u)$ aura la valeur vraie si t a la valeur i et u a la valeur j et que la case sur la ligne i et la colonne j est noir.

1. Soient les formules :
   (a) $\forall x\, y, P(x, y)$
   (b) $\exists x\, y, P(x, y)$
   (c) $\exists x, \forall y, P(x, y)$
   (d) $\exists y, \forall x, P(x, y)$
   (e) $\forall x, \exists y, P(x, y)$
   (f) $\forall y, \exists x, P(x, y)$

   Dire si ces formules sont ou non vraies dans chacune des interprétations de P suivantes :

   (1) (2) (3) (4) (5)

2. Expliquer pour chaque formule comment on reconnait sur le tableau si la formule est vraie ou non dans l'interprétation correspondante.

Exercice 4.5 *Modèles de relation, examen session 2 2014-15*

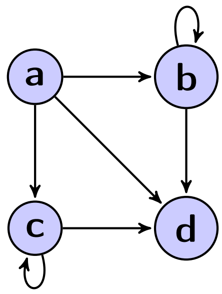
On se place dans un langage avec un symbole de prédicat binaire R. Soient les quatre formules de la logique du premier ordre suivantes :

— F_1 : $\forall x, ((\exists y, \neg R(x, y)) \Rightarrow \exists y, (R(x, y) \land R(y, x)))$
— F_2 : $\forall x, \exists y, (R(x, y) \lor R(y, x))$
— F_3 : $\forall x\, y\, z, ((R(x, y) \land R(y, z)) \Rightarrow R(x, z))$
— F_4 : $\exists x, R(x, x)$

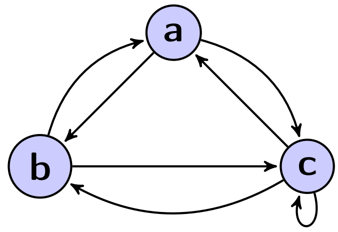
1. On se donne des interprétations de la relation R sous forme de graphes. Le domaine est l'ensemble des sommets du graphe et on a une arête du sommet x au sommet y exactement lorsque la relation $R(x, y)$ est vérifiée dans l'interprétation. Dans chacune des deux interprétations suivantes :

(a) donner la liste des couples (x, y) tels que $R(x, y)$ est vraie dans l'interprétation ;

(b) dire lesquelles des formules F_1, F_2, F_3, F_4 précédentes sont vraies et lesquelles sont fausses. Justifier votre réponse.



modèle (A)



modèle (B)

2. Montrez que

   (a) la formule F_2 est conséquence logique de la formule F_1 ;

   (b) la formule F_4 est conséquence des deux formules F_1 et F_3.

3. A l'aide d'une variante du modèle A, montrer que F_4 n'est pas conséquence logique de F_2 et F_3.

Exercices d'approfondissement

Exercice 4.6 *Un peu de théorie.*

On considére des formules propositionnelles sur 2 variables p, q.

1. Montrer qu'il y a une infinité de formules utilisant ces deux variables.
2. Combien y-a-t-il d'interprétations possibles utilisant ces 2 variables ?
3. Combien y-a-t-il de tables de vérité possibles pour des formules utilisant ces 2 variables ?
4. On se donne 20 formules propositionnelles qui utilisent p et q, montrer qu'il y a forcément deux formules parmi les 20 qui sont logiquement équivalentes.
5. Si on considère des formules construites sur n variables, combien de formules faut-il prendre pour être sûr d'en avoir deux différentes qui sont équivalentes ?

Exercice 4.7 Cet exercice a pour objectif de prouver un résultat théorique qui s'appelle le théorème de compacité.

L'énoncé de ce théorème est le suivant : soit un ensemble infini $\mathcal{E}$ de formules propositionnelles. On suppose que cet ensemble est insatisfiable. Alors il existe un sous-ensemble fini de $\mathcal{E}$ qui est insatisfiable.

1. Redonner la définition de la propriété $\mathcal{E}$ est insatisfiable.
2. Montrer que si un sous-ensemble fini de $\mathcal{E}$ est insatisfiable alors $\mathcal{E}$ est insatisfiable.
3. Dire pourquoi le résultat est vrai dans le cas où il n'y a qu'un nombre fini de variables propositionnelles. On étudiera l'exemple de l'ensemble $\mathcal{E} = \{B \Rightarrow A, A \Rightarrow \neg B, \neg A \Rightarrow B\} \cup (M_i)_{i \in \mathbb{N}}$ avec $M_0 = A \Rightarrow B$ et $M_{n+1} = M_n \Rightarrow B$
4. On suppose maintenant qu'il y a un ensemble dénombrable de variables propositionnelles $(X_i)_{i \in \mathbb{N}}$. On construit un arbre binaire de décision possiblement infini pour l'ensemble de formule $\mathcal{E}$ de la manière suivante :
   — Les sommets sont étiquetés par les variables propositionnelles, la racine de l'arbre est étiquetée par la variable X_0, les fils d'un nœud étiqueté par la variable X_i sont étiquetés par la variable X_{i+1}.

- Le fils gauche d'un nœud étiqueté X_i correspond au cas $X_i = V$ et le fils droit au cas $X_i = F$.
- Chaque branche correspond à une interprétation partielle $X_0 = V/F, \ldots, X_i = V/F$
- Une feuille de l'arbre est étiquetée par une formule qui est fausse dans l'interprétation qui correspond à la branche.

Montrer le résultat dans ce cas.