Eléments de logique pour l'informatique
Exercices I

Uli Fahrenberg
uli@lmf.cnrs.fr

d'après Christine Paulin

11 septembre 2025

2 Syntaxe des formules

Exercice 2.1 Soient p et q deux symboles de prédicat binaire, r un symbole de prédicat unaire, f un symbole de fonction unaire, a un symbole de fonction 0-aire (ou constante) et g un symbole de fonction ternaire. Soit les formules du calcul des prédicats suivantes :

$$\begin{aligned} F &= (\exists x, p(x, f(y))) \vee \neg\forall y, q(y, g(a, z, f(z))) \\ G &= r(x) \vee ((\exists x, \forall y, p(f(x), z)) \wedge r(a)) \wedge \forall x, q(y, g(x, z, x)) \end{aligned}$$

1. Représenter chaque formule sous forme d'arbre dont les feuilles sont les formules atomiques.
2. Pour chaque occurrence de variable, dire si elle est libre ou liée et dans le cas où elle est liée, indiquer le quantificateur correspondant.
3. Donner les formules atomiques qui apparaissent dans chaque formule.
4. Donner tous les termes (en ignorant les sous-termes) qui apparaissent dans chaque formule.
5. Donner le résultat de la substitution dans chaque formule de la variable y par le terme $f(a)$ et de la variable z par le terme $f(x)$.

3 Langage logique, Modélisation

Exercice 3.1 Un logicien affirme "les personnes qui aiment la montagne aiment aussi la campagne".

1. Sachant que monsieur X n'aime pas la campagne, peut-on en déduire s'il aime ou non la montagne ?
2. Si Madame Y n'aime pas la montagne, peut-on en déduire si elle aime ou non la campagne ?
3. Que dire des phrases
   (a) "les personnes qui n'aiment pas la montagne n'aiment pas la campagne".
   (b) "les personnes qui n'aiment pas la campagne n'aiment pas la montagne".
   (c) "les personnes qui aiment la campagne aiment aussi la montagne".

   sont-elles équivalentes à l'affirmation du logicien ? lesquelles sont équivalentes entre elles ?
4. Traduire la phrase du logicien et les trois affirmations précédentes en des formules logiques qui utilisent des symboles de prédicat unaires `Mont` pour ceux qui aiment la montagne et `Camp` pour ceux qui aiment la campagne.

5. Donner deux formules logiques différentes qui expriment la négation de l'affirmation du logicien.

Exercice 3.2 Soit `joue` un prédicat binaire tel que $\texttt{joue}(x, y)$ représente le fait que x joue avec y. Dire si les affirmations suivantes sont correctes ou non :

1. $\forall x, \exists y, \neg\texttt{joue}(x, y)$ signifie qu'il existe quelqu'un avec qui personne ne joue.
2. La formule $(\exists x\, y, \texttt{joue}(x, y)) \Rightarrow \exists z, \texttt{joue}(z, z)$ est toujours vraie.
   qu'en est-il de la réciproque : $(\exists z, \texttt{joue}(z, z)) \Rightarrow (\exists x\, y, \texttt{joue}(x, y))$
3. $\forall x, \exists y, \exists z, (\texttt{joue}(x, y) \wedge \texttt{joue}(x, z))$ signifie que toute personne joue avec au moins deux personnes différentes.

Exercice 3.3 *Formalisation logique*

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

— $\texttt{B}(x), \texttt{H}(x), \texttt{V}(x)$: le dragon x est bleu, est heureux, vole
— $\texttt{P}(x, y)$: 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) $\forall x\, y\, z, \texttt{V}(x) \Rightarrow \texttt{P}(x, y) \Rightarrow \texttt{P}(x, z) \Rightarrow y = z$
   (b) $\forall x, \exists y, \texttt{P}(x, y) \wedge \texttt{H}(y)$
   (c) $\exists y, \forall x, \texttt{P}(x, y) \wedge \texttt{H}(y)$

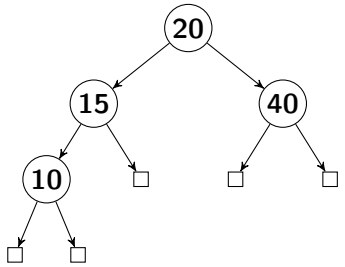
Exercice 3.4 *Modélisation*

La signature pour représenter des arbres binaires de recherche (ABR) contient une constante `nil` (feuille de l'arbre), et un symbole de fonction d'arité 3 : `node`.

On a une constante pour chaque entier naturel $0, 1, \ldots, 10, \ldots, 15 \ldots$

Dans l'expression $\texttt{node}(l, v, r)$, l représente le sous-arbre gauche, r le sous-arbre droit et v la valeur stockée dans le nœud.

Pour chaque nœud interne $\texttt{node}(l, v, r)$ d'un ABR, les valeurs dans le sous-arbre gauche l sont strictement plus petites que v qui est lui-même strictement plus petit que toutes les valeurs dans le sous-arbre droit r.

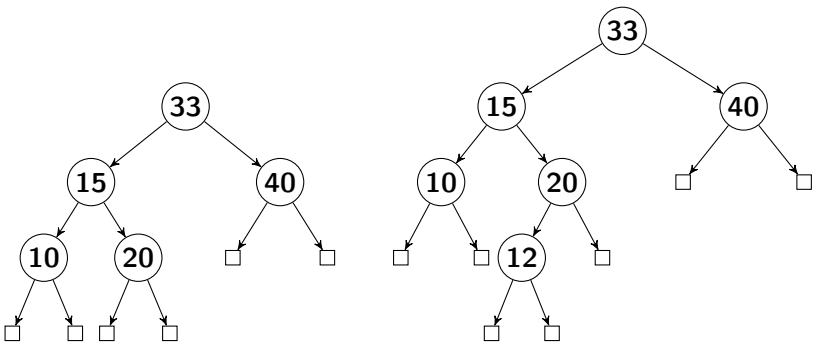


est un arbre binaire de recherche. Il correspond au terme $\texttt{node}(\texttt{node}(\texttt{node}(\texttt{nil}, 10, \texttt{nil}), 15, \texttt{nil}), 20, \texttt{node}(\texttt{nil}, 40, \texttt{nil}))$.

Les symboles de prédicats sont $=$, $<$ et $\in$ d'arité 2, notés de manière infixe.

— $x = y$ représente l'égalité des objets x et y,
— $x < y$ représente le fait que x est strictement plus petit que y
— $v \in t$ est vrai si la valeur v apparaît dans un des nœuds de l'arbre t.

1. Dire si les arbres suivants sont bien des ABR sur les entiers avec l'ordre standard.



   Si c'est un arbre binaire de recherche, écrire le terme correspondant. Sinon justifier pourquoi.

2. Ecrire en utilisant les prédicats de la signature une formule logique avec deux variables libres t et n qui exprime le fait que n est la valeur maximum dans l'arbre t.

3. On suppose vérifiées les propriétés suivantes
   (a) $\forall x, \neg(x \in \texttt{nil})$
   (b) $\forall x\, l\, v\, r, (x \in \texttt{node}(l, v, r) \Leftrightarrow x \in l \vee x = v \vee x \in r)$
   (c) $\forall t_1\, t_2, \exists t, \forall x, (x \in t \Leftrightarrow x \in t_1 \wedge x \in t_2)$

   Expliquer chacune de ces propriétés en langage naturel.

4. On ajoute à la signature un prédicat unaire `abr`. On veut que $\texttt{abr}(t)$ représente la propriété que t est un arbre binaire de recherche.

   En vous inspirant de la question précédente, proposer deux propriétés du prédicat `abr` (l'une dans le cas $t = \texttt{nil}$ et l'autre dans le cas $t = \texttt{node}(l, v, r)$) qui caractérisent le fait que l'arbre t vérifie les propriétés des arbres binaires de recherche.

5. On ajoute à la signature une fonction binaire `union`. Ecrire une propriété qui spécifie que si t et u sont des arbres binaires de recherche alors $\texttt{union}(t, u)$ est un arbre binaire de recherche qui contient *exactement* les éléments de t et les éléments de u.