Eléments de logique pour l'informatique
Exercices I

Uli Fahrenberg
uli@lmf.cnrs.fr

d'après Christine Paulin

30 septembre 2025

Table des matières

1 Calcul Propositionnel

Exercice 1.1 La contraposée d'une formule $P \Rightarrow Q$ est la formule $\neg Q \Rightarrow \neg P$, la réciproque est la formule $Q \Rightarrow P$, la contraposée de la réciproque est la formule $\neg P \Rightarrow \neg Q$.

1. Exprimer par des phrases ces 4 formules en prenant pour P la propriété « il est midi » et pour Q la propriété « j'ai faim ».
2. Pour P et Q arbitraires construire les tables de vérité de ces quatre formules, lesquelles sont équivalentes ?
3. Attention aux usages courants qui ne respectent pas toujours les règles de la logique. Soit la formule P « Max a bu » et la formule Q « Max ne peut pas conduire ». On suppose que la formule $P \Rightarrow Q$ est vérifiée. Peut-on en déduire que si Max n'a pas bu alors il peut conduire ?

Exercice 1.2 *Table de vérité*
Soient les formules $A \stackrel{\text{def}}{=} P \wedge Q \Rightarrow R$ et $B \stackrel{\text{def}}{=} P \Rightarrow Q \Rightarrow R$.

1. Ajouter des parenthèses autour des connecteurs sans changer le sens de ces formules.
2. Donner les tables de vérité de ces deux formules. Que constate-t-on ?
3. Reprendre les mêmes questions avec les formules $P \vee Q \Rightarrow R$ et les formules $(P \Rightarrow R) \wedge (Q \Rightarrow R)$.

Exercice 1.3 *Partiel 2012.*
Un homme qui semble divaguer déclare à toute la clientèle d'un café :

1. Le jour où je ne bois pas et où je dors, je ne suis pas content.
2. Le jour où je bois, je ne suis pas content et je dors.
3. Le jour où je ne mange pas, ou bien je ne suis pas content, ou bien je dors ou les deux.
4. Le jour où je mange, ou bien je suis content, ou bien je bois ou les deux.
5. Aujourd'hui, je suis content.

Questions

1. Introduire des variables propositionnelles pour représenter les principales notions et donner les formules correspondantes à chacune des affirmations précédentes. Combien y-a-t-il a priori de situations possibles ?
2. On considère que toutes les affirmations précédentes sont vraies.
   (a) Par raisonnement élémentaire à partir des affirmations, montrer qu'il n'a pas bu.
   (b) Répondre en les justifiant par un raisonnement ou une table de vérité aux questions suivantes : a-t-il mangé ? a-t-il dormi ?

Exercice 1.4 *Structure arborescente des formules.*
Soit la formule A définie comme $\neg P \Rightarrow Q \vee \neg(P \vee R)$.

1. Parenthéser la formule A en préservant le sens
2. Donner la forme arborescente de cette formule.
3. Pour quelles valeurs de P, Q et R, la formule A est-elle vraie ? (on essaiera de répondre sans construire l'ensemble de la table de vérité).

Exercice 1.5 Dans la suite X, Y et Z sont des variables propositionnelles (symboles de prédicat d'arité 0) **distinctes**. Dire pour chacune des affirmations suivantes si elle est vraie ou fausse.

Affirmation	vrai	faux
$A = B$ représente le fait que les deux formules sont syntaxiquement égales (même représentation).		
$X \vee Y = Y \vee X$		
$X \Rightarrow Y \wedge Z = X \Rightarrow (Y \wedge Z)$		
$A \equiv B$ représente le fait que les deux formules sont sémantiquement égales (même table de vérité).		
$(X \vee Y) \wedge X \equiv X$		
$(X \wedge Y) \vee Z \equiv X \vee Z$		
$\neg(X \Rightarrow Y) \equiv \neg X \wedge \neg Y$		
Si $X \Rightarrow Y$ est faux alors X est vrai.		
Si $X \vee Y$ est vrai alors Y est vrai.		
Si X est vrai alors $(\neg X) \Rightarrow Y$ est faux.		
Une formule est *satisfiable* s'il existe des valeurs pour les variables propositionnelles qui la rendent vraie. Une formule est *valide* si elle est vraie quelles que soient les valeurs des variables propositionnelles.		
La formule $X \Rightarrow \neg Y$ est satisfiable		
La formule $X \Rightarrow \neg Y$ est valide		

Autres énigmes

Exercice 1.6 *Enigme, d'après Smullyan, partiel 2013*
Une aventurière découvre 3 coffres numérotés de 1 à 3. Un seul de ces coffres contient un trésor qu'il faut découvrir, les autres coffres sont piégés. Chaque coffre comporte une inscription :

1. Le trésor est dans ce coffre
2. Le trésor n'est pas dans ce coffre
3. Le trésor n'est pas dans le coffre 1

Questions. On introduit des variables propositionnelles P_1 pour représenter le fait que le trésor est dans le coffre 1 et P_2 pour représenter le fait que le trésor est dans le coffre 2.

1. Donner une formule qui utilise les variables P_1 et P_2 et qui est vraie exactement lorsque le trésor est dans le coffre 3.
2. Donner une formule qui utilise les variables P_1 et P_2 et qui représente le fait que le trésor est exactement dans un des coffres.
3. Donner des formules I_1, I_2 et I_3 qui utilisent les variables P_1 et P_2 et qui représentent les inscriptions sur chacun des coffres.
4. Sachant qu'une seule des formules I_1, I_2 et I_3 est vraie, en déduire dans quel coffre est caché le trésor.

Exercice 1.7 *Enigme.* Trois collègues, Albert, Bernard et Charles déjeunent ensemble chaque jour ouvrable. Les affirmations suivantes sont vraies :

1. Si Albert commande un dessert, Bernard en commande un aussi.
2. Chaque jour, soit Bernard, soit Charles, mais pas les deux, commandent un dessert.
3. Albert ou Charles, ou les deux, commandent chaque jour un dessert.
4. Si Charles commande un dessert, Albert fait de même.

Questions

1. Exprimer les données du problème comme des formules propositionnelles
2. Que peut on en déduire sur qui commande un dessert ?
3. Pouvait-on arriver à la même conclusion en supprimant l'une des quatre affirmations ?

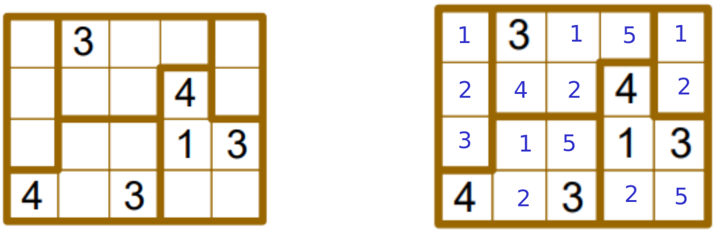
Exercice d'approfondissement

Exercice 1.8 *Modélisation propositionnelle du jeu de Tectonic (d'après session 2 2022-23)*

Le but de cet exercice est de modéliser le jeu de tectonic en logique propositionnelle.

Ce jeu se joue sur une grille découpée en zones de 1 à 5 cases. Il s'agit de compléter chaque case de chaque zone par un chiffre entre 1 et la taille de la zone. Deux cases qui se touchent par un bord ou un coin ne peuvent pas avoir le même chiffre. Par exemple si le chiffre 4 se trouve sur une case, il ne peut par apparaître sur les 8 cases qui l'entourent.

La figure ci-dessous présente un exemple de grille et sa solution.



On manipule des *cases*, avec une relation de *deux cases qui se touchent par un bord ou un coin* et des *valeurs* (les chiffres entre 1 et 5).

L'organisation spatiale des cases dans des *zones* et leur taille varie d'un problème à l'autre, on va donc les traiter de manière abstraite.

En logique propositionnelle, il n'y a pas de quantificateur dans les formules (juste les connecteurs $\neg, \wedge, \vee, \Rightarrow$) et les seuls symboles dans la signature sont des variables propositionnelles.

On introduit une variable propositionnelle X_p^i pour chaque case p et chaque valeur i possible. La variable X_p^i sera vraie dans une grille où la solution pour la case p est le chiffre i.

Les notions de case, de zones, d'adjacence ou de max sont des objets mathématiques qui dépendent du problème :

— `case` est l'ensemble des cases
— `max` est ue fonction qui associe à chaque case un entier entre 1 et 5
— $\texttt{adj} \subseteq \texttt{case} \times \texttt{case}$ est l'ensemble des couples de cases qui se touchent
— $\texttt{zone} \subseteq \texttt{case} \times \texttt{case}$ est l'ensemble des couples de cases qui sont dans la même zone

L'ensemble des variables propositionnelles pour modéliser une grille est

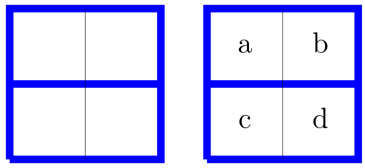
$$\{X_p^i \mid p \in \texttt{case}, 1 \leq i \leq \texttt{max}(p)\}$$

L'ensemble $\mathcal{E}$ des formules propositionnelles qui expriment le fait qu'on peut compléter toutes les cases de la grille par au moins un chiffre compris entre 1 et la taille de la zone s'écrira :

$\{X_p^1 \vee \ldots \vee X_p^{\texttt{max}(p)} | p \in \texttt{case}\}$

1. Sur l'exemple de grille donné, indiquer le nombre de variables propositionnelles nécessaires et le cardinal de l'ensemble des formules $\mathcal{E}$
2. Donner dans le cas général l'ensemble des formules qui expriment que deux cases différentes qui sont dans la même zone ou qui sont adjacentes ne peuvent pas avoir la même valeur (attention, en calcul propositionnel, il n'y a pas d'égalité dans les formules engendrées).

3. Donner dans le cas général l'ensemble des formules qui expriment qu'une case ne peut pas avoir deux valeurs différentes.
4. On suppose que l'on a construit un ensemble X de formules propositionnelles pour toutes les conditions énoncées ci-dessus dans le cas de la grille de l'exemple. On veut maintenant résoudre le problème en utilisant des outils logiques.
   (a) Quelles formules faut-il ajouter à la modélisation ?
   (b) A quelle question logique sur l'ensemble de formules obtenu correspond une solution du jeu ?
5. On s'intéresse à une grille très simple avec deux zones de deux cases.



   On note les cases a, b, c, d (voir schéma).
   (a) Pour cette grille, expliciter les relations d'adjacence, de zones ainsi que la fonction `max`.
   (b) On veut montrer que cette grille n'a pas de solution. Donner un ensemble de formules propositionnelles qui correspondent aux règles du jeu (comme vu aux questions précédentes), qui concernent uniquement les cases a, b et c et à partir desquelles vous établirez une contradiction (par la méthode de votre choix).

Exercice 1.9 *Coloriage de graphe*

Le problème de coloriage de graphe consiste à se donner un ensemble fini de couleurs et à associer à chaque sommet d'un graphe une couleur de manière à ce que deux sommets adjacents n'aient pas la même couleur.

Pour chaque couleur $c \in C$ on se donne un ensemble de variables propositionnelles (x_i^c) avec (x_i^c) qui sera vrai si le sommet i a la couleur c.

— Combien y-a-t-il de variables pour un graphe de n sommets et k couleurs ?
— Proposer un ensemble de formules tel que toute interprétation qui rend vraies ces formules correspond à un coloriage du graphe.
— Soit un graphe infini avec un nombre dénombrable de sommets, montrer que si tous les sous-graphes finis sont coloriables alors le graphe complet est coloriable.

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) $\exists y, \forall x, \texttt{P}(x, y) \wedge \texttt{H}(y)$

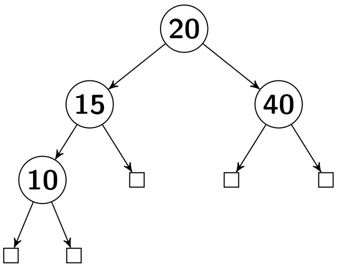
Exercice 3.4 *Modélisation*

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 $\texttt{node}(\texttt{node}(\texttt{node}(\texttt{nil}, 10, \texttt{nil}), 15, \texttt{nil}), 20, \texttt{node}(\texttt{nil}, 40, \texttt{nil}))$.

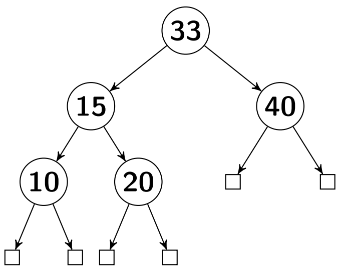
Les symboles de prédicats sont $=$, $<$ et $\in$ d'arité 2, notés de manière infixe.

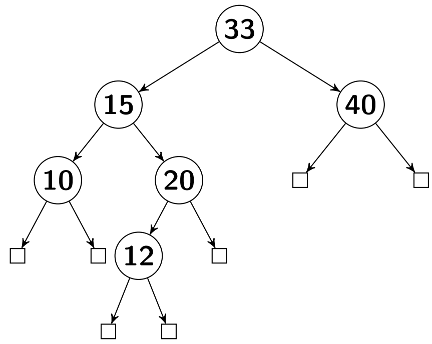
— $x = y$ représente l'égalité des objets x et y,

— $x < y$ 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.

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.

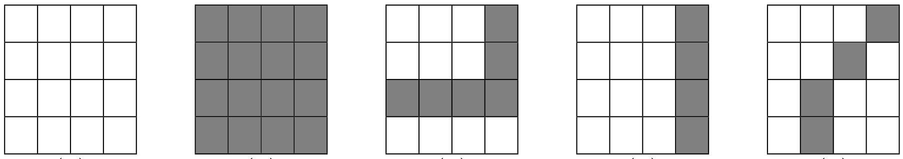
On considère un langage avec juste un symbole de prédicat binaire P.

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)

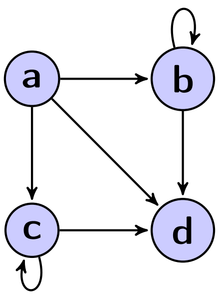
2. Expliquer pour chaque formule comment on reconnait sur le tableau si la formule est vraie ou non dans l'interprétation correspondante.

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

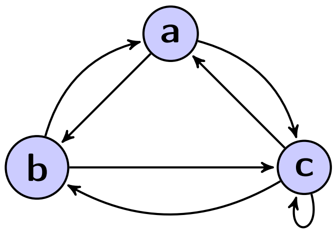
On se place dans un langage avec un symbole de prédicat binaire R. Soient les quatre formules de la logique du premier ordre suivantes :

— F_1 : $\forall x, ((\exists y, \neg R(x,y)) \Rightarrow \exists y, (R(x,y) \wedge R(y,x)))$
— F_2 : $\forall x, \exists y, (R(x,y) \vee R(y,x))$
— F_3 : $\forall x\, y\, z, ((R(x,y) \wedge R(y,z)) \Rightarrow R(x,z))$
— 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.

Exercice 4.8 *Modélisation du jeu de Tectonic (d'après session 2 2022-23)*

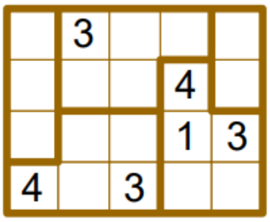
Le but de cet exercice est de modéliser le jeu de tectonic en logique du premier ordre. Ce jeu se joue sur une grille découpée en zones de 1 à 5 cases. Il s'agit de compléter chaque case de chaque zone par un chiffre entre 1 et la taille de la zone. Deux cases qui se touchent par un bord ou un coin ne peuvent pas avoir le même chiffre. Par exemple si le chiffre 4 se trouve ur une case, il ne peut par apparaître sur les 8 cases qui l'entourent.

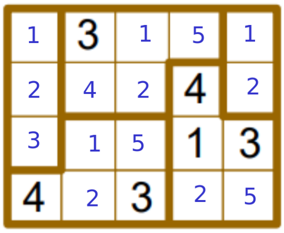
La figure ci-dessous présente un exemple de grille et sa solution.

On manipule des *cases*, avec une relation de *deux cases qui se touchent par un bord ou un coin* et des *valeurs* (les chiffres entre 1 et 5).

L'organisation spatiale des cases dans des *zones* et leur taille varie d'un problème à l'autre, on va donc les traiter de manière abstraite.

La signature du langage se compose :





— d'un symbole de fonction unaire `max` qui associe à chaque case sa *valeur maximale*, à savoir l'entier le plus grand entre 1 et 5 qui peut être mis dans la case (correspond au nombre de cases dans une zone) ;

— de deux symboles de prédicats unaires `case` et `val`. Le formule $\texttt{case}(x)$ est vraie exactement lorsque x représente une case et $\texttt{val}(x)$ est vraie exactement lorsque x représente une valeur ;

— de cinq symboles de prédicats binaires
  - — l'égalité $t = u$ est vraie lorsque t et u sont le même objet
  - — l'ordre $n \leq m$ est vrai lorsque n est une valeur inférieure ou égale à la valeur m.
  - — l'adjacence $\texttt{adj}(p, q)$ est vraie lorsque p et q sont deux cases qui se touchent (par un bord ou un coin)
  - — les zones : $\texttt{zone}(p, q)$ est vraie lorsque p et q sont deux cases dans la même zone
  - — la solution : $\texttt{X}(p, n)$ est vraie lorsque la case p a la valeur n.

1. On modélise la grille de notre exemple en représentant une case par un couple (i, j) avec i le numéro de ligne et j le numéro de colonne.
   - — Le domaine de notre interprétation est donc formé des éléments suivants :
     $\mathcal{D} = \{(i, j) \in \mathbb{N} \times \mathbb{N} \mid 1 \leq i \leq 4 \text{ et } 1 \leq j \leq 5\} \cup \{1, 2, 3, 4, 5\}$
   - — L'interprétation du prédicat `val` est l'ensemble $\{1, 2, 3, 4, 5\}$ et l'interprétation du prédicat `case` est l'ensemble $\{(i, j) \in \mathbb{N} \times \mathbb{N} \mid 1 \leq i \leq 4, 1 \leq j \leq 5\}$
   - — L'interprétation du symbole d'égalité est l'égalité du domaine et l'ordre sur les valeurs est interprété comme l'ordre usuel $1 \leq 2 \leq 3 \leq 4 \leq 5$ et est faux si un des arguments n'est pas une valeur.
   - — L'interprétation de `max` est une fonction dont les valeurs se déduisent de la grille par exemple $\texttt{max}((1, 1)) = \texttt{max}((2, 1)) = \texttt{max}((3, 1)) = 3$ ou encore $\texttt{max}((4, 5)) = 5$

   L'objet de cette question est de compléter le modèle pour cet exemple de grille particulier.

   (a) Donner l'interprétation du prédicat `adj` de manière implicite sous la forme
   $\{((i_1, j_1), (i_2, j_2)) \mid\ \ 1 \leq i_1, i_2 \leq 4, \text{ et } 1 \leq j_1, j_2 \leq 5 \text{ et } \ldots \text{ compléter } \ldots\}$

   (b) L'interprétation du symbole `zone` est un ensemble de couples de cases. Donner trois couples de cases qui sont dans cet ensemble et trois qui ne le sont pas.

2. Exprimer en langage courant le sens des formules suivantes

   (a) $\forall p, \texttt{case}(p) \Rightarrow \texttt{val}(\texttt{max}(p))$

   (b) $\forall p, \texttt{case}(p) \Rightarrow \exists i, i \leq \texttt{max}(p) \wedge X(p, i)$

   (c) $\neg \exists p\, q, \texttt{zone}(p, q) \wedge \texttt{max}(p) \neq \texttt{max}(q)$

3. Donner une formule en forme normale de négation équivalente à la négation de la formule
   $\forall p\, q\, r, \texttt{zone}(p, q) \wedge \texttt{zone}(q, r) \Rightarrow \texttt{zone}(p, r)$

4. Pour une grille quelconque, définir les propriétés suivantes par des formules en logique du premier ordre qui utilisent la signature proposée :

   (a) Deux cases adjacentes n'ont pas la même valeur.

   (b) Deux cases différentes dans la même zone ne peuvent pas avoir la même valeur.

   (c) Il existe une zone formée exactement de deux cases.

(d) Une case ne peut pas avoir deux valeurs différentes.

Exercice 4.9 *Modélisation, modèles et preuves (examen session 2 2018-19).*
On s'intéresse au problème de coloriage d'un graphe. Il s'agit d'attribuer une couleur à chaque sommet du graphe de manière à ce que deux sommets qui sont reliés par une arête n'aient pas la même couleur.

On se donne un langage dans lequel il y a deux prédicats binaires E et `col` et un prédicat unaire V tels que

— $V(x)$ est vrai exactement lorsque x est un *sommet du graphe*

— $E(x, y)$ est vrai exactement lorsque x et y sont deux sommets *reliés par une arête* du graphe.

— $\texttt{col}(x, c)$ est vrai exactement lorsque le sommet x est *associé à la couleur c.*

D'un point de vue logique, n'importe quel élément de l'univers peut représenter une couleur, y compris les sommets du graphe. Attention, il n'y a pas de prédicat pour l'égalité dans le langage, les formules ne devront donc pas utiliser l'égalité.

1. Formaliser les énoncés suivants qui seront par la suite notés P et Q :
   P : A tout sommet est associé (au moins) une couleur
   Q : Deux sommets reliés par une arête ne sont pas associés à la même couleur
2. Soit la formule $\forall x, V(x) \Rightarrow \neg E(x, x)$ notée R.
   (a) A quelle propriété du graphe de sommets V et d'arêtes E correspond la formule R ?
   (b) En utilisant la méthode de résolution (dont on détaillera les étapes), montrer que R est conséquence logique de P et de Q.
3. Soient un domaine D et des interprétations de E et de V quelconques qui satisfont la propriété R.
   Montrer qu'il existe toujours une interprétation de `col` telle que les formules P et Q sont vraies dans cette interprétation.
4. On modifie le problème en introduisant dans le langage deux constantes `bleu` et `rouge`. On remplace la formule P qui dit qu'à tout sommet est associé une couleur, par la formule P' qui dit qu'à tout sommet est associé une des deux couleurs `bleu` ou `rouge`.
   (a) Donner la formule P'.
   (b) On se place sur le domaine D des entiers naturels, et dans une interprétation dans laquelle le prédicat V est toujours vrai et la relation $E(n, m)$ est vraie si et seulement si $n + m$ est impair. La couleur `bleu` est interprétée 0 et la couleur `rouge` par 1.
      i. La formule R est-elle vraie dans cette interprétation ?
      ii. Trouver une interprétation de `col` qui rend vraies les formules P' et Q.
   (c) Dans un domaine D et une interprétation de E et de V quelconque qui rend vraie la formule R, existe-t-il toujours une interprétation de `col` qui rend vraies les formules P' et Q ? Justifier votre réponse.

5 Définitions récursives, Substitution

Exercice 5.1 *Récurrence sur les formules.*

1. On considère la fonction `simpl` définie par les équations suivantes dans lesquelles a est une variable propositionnelle choisie arbitrairement :

$$\begin{array}{llll}
\texttt{simpl}(\top) & = a \vee \neg a & \texttt{simpl}(P \wedge Q) & = \neg(\neg\texttt{simpl}(P) \vee \neg\texttt{simpl}(Q)) \\
\texttt{simpl}(\bot) & = \neg(a \vee \neg a) & \texttt{simpl}(P \vee Q) & = \texttt{simpl}(P) \vee \texttt{simpl}(Q) \\
\texttt{simpl}(p) & = p \quad p \text{ variable prop.} & \texttt{simpl}(P \Rightarrow Q) & = \neg\texttt{simpl}(P) \vee \texttt{simpl}(Q) \\
\texttt{simpl}(\neg P) & = \neg(\texttt{simpl}(P)) & &
\end{array}$$

— Donner la formule résultat de $\texttt{simpl}(x \wedge y \Rightarrow z)$.
— Que fait la fonction `simpl` en général (comment $\texttt{simpl}(P)$ est relié à P et quelles sont ses propriétés syntaxiques) ?

2. Donner les équations récursives qui définissent une fonction `ht` qui mesure la hauteur d'une formule (nombre maximal de connecteurs emboités)

$$\begin{array}{lll} \texttt{ht}(P) & = \ldots. & \texttt{si } P \texttt{ atomique} \\ \texttt{ht}(\neg P) & = \ldots.. & \\ \texttt{ht}(P \circ Q) & = \ldots\ldots. & \circ \in \{\vee, \wedge, \Rightarrow\} \end{array}$$

3. Démontrer par récurrence sur la hauteur d'une formule P que pour toute formule P, `simpl`(P) ne contient pas le symbole $\Rightarrow$.
4. Pour prouver qu'une propriété $\phi(P)$ est vérifiée pour toutes les formules propositionnelles P, on peut raisonner par récurrence **structurelle** sur les formules, suivant le principe :

   — Si on peut montrer que :
   (a) $\phi(P)$ est vérifiée lorsque P est une formule atomique en particulier $\phi(\top)$ et $\phi(\bot)$;
   (b) pour une formule propositionnelle A quelconque, en supposant que $\phi(A)$ est vérifiée, on peut montrer $\phi(\neg A)$
   (c) pour des formules propositionnelles A et B quelconques, en supposant que $\phi(A)$ et $\phi(B)$ sont vérifiées, on peut montrer $\phi(A \wedge B)$ ainsi que $\phi(A \vee B)$ et $\phi(A \Rightarrow B)$
   — Alors on peut en déduire que pour tout $P \in \textsc{Prop}$, $\phi(P)$ est vérifiée.

   Utiliser ce schéma pour montrer la propriété attendue de $\texttt{simpl}(P)$.
5. (optionnel) Justifier la correction de ce schéma de preuve en le ramenant à une récurrence sur les entiers.
   — On suppose que les conditions (1) à (3) du principe de récurrence sur les formules sont satisfaites par la propriété ϕ. Montrer par récurrence sur n que pour toute formule P telle que $\texttt{ht}(P) \leq n$, on a $\phi(P)$ est vérifiée.
   — En déduire que $\phi(P)$ est vérifiée pour toutes les formules P.

Exercice 5.2 *Définition récursive propositionnelle.*

1. Donner une formule équivalente à $\neg(P \Rightarrow Q)$ qui n'utilise que la conjonction et la négation.
2. On souhaite écrire une fonction qui pour toute formule propositionnelle P calcule une nouvelle formule $\texttt{neg}(P)$ qui est logiquement équivalente à $\neg P$ mais qui au lieu d'ajouter un symbole de négation en tête de la formule, effectue de possibles simplifications et n'ajoute éventuellement un nouveau symbole de négation qu'au niveau d'une formule atomique. Pour cela on utiliserale fait que les formules suivantes sont valides :

$$\neg\bot \Leftrightarrow \top \qquad \neg\neg A \Leftrightarrow A \quad \neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q \qquad \neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$$

   (a) Calculer la valeur attendue pour $\texttt{neg}(A)$ avec A définie comme $(\neg P) \Rightarrow (Q \vee \neg(P \vee R))$. Pour quelle valeur de P, Q et R cette formule est-elle vraie ?
   (b) Donner les équations récursives qui définissent $\texttt{neg}(P)$
   (c) Montrer par récurrence structurelle sur la formule P que $\texttt{neg}(P)$ est logiquement équivalent à $\neg P$.

Exercice 5.3 *Sous-formules.* On dit qu'une formule Q est une sous-formule de P si $Q = P$ ou bien si la formule Q apparait sous un connecteur de P. C'est-à-dire $P = \neg P'$ et Q est une sous-formule de P' ou bien $P = P_1 \circ P_2$ et Q est une sous-formule de P_1 ou bien une sous-formule de P_2 avec $\circ$ un des 3 connecteurs binaires : $\{\vee, \wedge, \Rightarrow\}$.

1. Donner toutes les sous-formules de la formule $\neg(p \vee (q \wedge r)) \Rightarrow (p \wedge q)$
2. Donner les équations qui définissent la fonction `sf` dans $\text{PROP} \rightarrow \wp(\text{PROP})$ qui à une formule propositionnelle P associe l'ensemble de ses sous-formules.
3. Trouver un majorant du nombre de sous-formules d'une formule P qui utilise n connecteurs logiques. Donner un exemple où ce majorant est atteint. Prouver ce résultat par récurrence structurelle sur la formule.
4. (optionnel) Même question pour un minorant du nombre de sous-formules.

Exercice 5.4 *Définition récursive sur les termes*

On considère un langage avec des symboles de fonctions `f` binaire, `g` unaire et une constante `a` ainsi qu'un symbole de prédicat `R` binaire.

Donner les équations récursives qui définissent une fonction `xin` qui étant donnée une formule P du calcul des prédicats sur le langage précédent, et une variable x, renvoie vrai si la variable x est libre dans la formule P et faux sinon.

Exercice 5.5 On considère un langage avec des symboles de fonctions `f` binaire, `g` unaire et une constante `a` ainsi qu'un symbole de prédicat `R` binaire.

Donner les formules résultats des substitutions suivantes :

1. $\mathtt{R}(x, \mathtt{g}(y))[x \leftarrow \mathtt{f}(\mathtt{a}, \mathtt{g}(y)); y \leftarrow x]$
2. $\mathtt{R}(x, \mathtt{g}(y))[x \leftarrow \mathtt{f}(\mathtt{a}, \mathtt{g}(y))][y \leftarrow x]$
3. $\forall x, \mathtt{R}(x, \mathtt{g}(y))[y \leftarrow \mathtt{g}(z)]$
4. $\forall x, \mathtt{R}(x, \mathtt{g}(y))[y \leftarrow \mathtt{g}(x)]$
5. $\forall x, \mathtt{R}(x, \mathtt{g}(y))[x \leftarrow z]$
6. $\forall x, \mathtt{R}(x, \mathtt{g}(y))[x \leftarrow y]$

6 Théories

Exercice 6.1 On se donne un langage avec un symbole de prédicat binaire pour l'égalité et les axiomes de la théorie de l'égalité.

On ne s'intéressera qu'aux modèles dits équationnels dans lesquels l'égalité de la théorie est interprétée par l'égalité du domaine.

1. On s'intéresse aux interprétations sur des domaines qui ont moins de n élements (au sens large). Que faut-il ajouter au langage et aux axiomes pour que ce soit le cas ? C'est-à-dire donner un ensemble d'axiomes dont les modèles équationnels sont exactement les interprétations qui ont un domaine dont le cardinal est inférieur ou égal à n.
2. On s'intéresse aux interprétations sur des domaines qui ont plus de n élements (au sens large). Que faut-il ajouter au langage et aux axiomes pour que ce soit le cas ?

Exercice 6.2 La théorie des entiers de Peano s'appuie pour représenter les entiers sur un langage avec une constante 0, un symbole de fonction unaire S (également des symboles pour l'addition et la multiplication que nous ne considérons pas ici) et un symbole de prédicat binaire pour l'égalité.

Les axiomes pour ces symboles dans cette théorie sont

— Les axiomes pour la reflexivité, symétrie et transitivité de l'égalité

— La stabilité de l'opération S par rapport à l'égalité

$$\forall x\, y, x = y \Rightarrow S(x) = S(y)$$

— Les axiomes de la théorie des entiers pour S

1. $\forall x, \neg 0 = S(x)$
2. l'injectivité de l'opération S :

$$\forall x\, y, S(x) = S(y) \Rightarrow x = y$$

1. Montrer que tout modèle de ces formules a un domaine qui est infini.
2. Est-ce toujours le cas si on retire le premier axiome ?
3. Est-ce toujours le cas si on garde le premier axiome et qu'on retire le second ?
4. On se donne une théorie quelconque et on souhaite se limiter à des interprétations du langage qui ont un nombre infini d'éléments. Que faut-il ajouter au langage et aux axiomes pour que ce soit le cas ?

Exercices d'approfondissement

Exercice 6.3 On se donne un langage avec un symbole de prédicat binaire pour l'égalité et les axiomes de la théorie de l'égalité. On ne s'interessera qu'aux modèles dits équationnels dans lesquels l'égalité de la théorie est interprétée par l'égalité du domaine.

1. On s'intéresse aux interprétations sur des domaines qui ont moins de n élements (au sens large). Que faut-il ajouter au langage et aux axiomes pour que ce soit le cas ? C'est-à-dire donner un ensemble d'axiomes dont les modèles sont exactement les interprétations qui ont un domaine dont le cardinal est inférieur ou égal à n.
2. On s'intéresse aux interprétations sur des domaines qui ont plus de n élements (au sens large). Que faut-il ajouter au langage et aux axiomes pour que ce soit le cas ?
3. Proposer une théorie dont les modèles équationnels sont exactement ceux dont le domaine est un ensemble à 3 éléments.
4. Proposer une théorie dont les modèles équationnels sont exactement ceux dont le domaine a une infinité d'éléments. On donnera de préférence une solution avec un nombre fini de symboles et d'axiomes.
5. Montrer que si une théorie a des modèles finis dont le domaine est de cardinal arbitrairement grand alors elle admet aussi un modèle infini.
   Indication : on pourra étendre la théorie en ajoutant à la signature un nombre infini de constantes ($(c_n)_{n\in\mathbb{N}}$) et des axiomes pour que tout modèle de la nouvelle théorie soit forcément infini. Il faut ensuite justifier l'existence d'au moins un modèle de cette théorie étendue. On utilisera (sans le démontrer) le théorème de compacité qui dit qu'une théorie est satisfiable si et seulement si tout sous-ensemble fini de formules est satisfiable.

Exercice 6.4 *Exercice théorique modèles.* Le but de l'exercice est de montrer que dans un langage contenant uniquement m constantes $c_1, \ldots, c_m$ (avec $m \geq 1$) et pas de fonction, alors une formule $\forall x_1 \ldots x_n, \exists y_1 \ldots y_p, A$ avec A sans quantificateur est valide si elle est vraie dans toute interprétation qui a moins de $n+m$ éléments.

1. (**Sous-interprétations**) Soit deux interprétations I et J d'une même signature. On dit que J est une sous-interprétation de I si
   — le domaine E de J est inclus dans le domaine D de I
   — l'interprétation f_J d'un symbole f dans J est la restriction à E de l'interprétation f_I du symbole f dans I, c'est-à-dire que l'interprétation des constantes est la même et appartient donc à E et que $\forall e_1, \ldots, e_n \in E, f_J(e_1, \ldots, e_n) = f_I(e_1, \ldots, e_n)$.
   — l'interprétation R_J d'un symbole de relation R dans J est la restriction à E de l'interprétation R_I du symbole R dans I, c'est-à-dire que $R_J = \{(u_1, \ldots, u_n) \in E^n | R_I(u_1, \ldots, u_n)\}$

   Par exemple sur une signature comportant les constantes 0 et 1, les opérations d'addition et de multiplication et la relation d'ordre $\leq$, l'interprétation usuelle des entiers naturels de domaine $\mathbb{N}$ est une sous-interprétation de l'interprétation usuelle des entiers relatifs de domaine $\mathbb{Z}$.

   Soit J une sous-interprétation de I et e un environnement dans J. Montrer les résultats suivants :
   (a) Pour tout terme t, $\texttt{val}_I(e,t) = \texttt{val}_J(e,t)$
   (b) Si A est une formule sans quantificateur alors
      i. $\texttt{val}_I(e,A) = \texttt{val}_J(e,A)$ si A est sans quantificateur
      ii. Si $\texttt{val}_I(e, \forall x_1 \ldots x_n, A) = V$ alors $\texttt{val}_J(e, \forall x_1 \ldots x_n, A) = V$
      iii. Si $\texttt{val}_J(e, \exists x_1 \ldots x_n, A) = V$ alors $\texttt{val}_I(e, \exists x_1 \ldots x_n, A) = V$
      iv. Donner un exemple avec les interprétations $\mathbb{N} \subseteq \mathbb{Z}$ qui montre que les réciproques des deux derniers réultats sont fausses.
2. On se donne un langage contenant uniquement m constantes $c_1, \ldots, c_m$ avec $(1 \leq m)$ et pas de fonction et une formule $B \stackrel{\text{def}}{=} \forall x_1 \ldots x_n, \exists y_1 \ldots y_p, A$ avec A sans quantificateur et on suppose que cette formule est vraie dans toute interprétation de cardinal inférieur à $n+m$.

   Soit $I = (D, F_I, R_I)$ une interprétation. Chaque constante c_i est interprétée par un élément de D. On note C le sous-ensemble de D formé des valeurs des constantes. Soient $d_1, \ldots, d_n \in D$ quelconques, soit J la sous-interprétation de I construite en restreignant I sur le domaine $E \stackrel{\text{def}}{=} \{d_1, \ldots, d_n\} \cup C$.
   (a) Donner une borne du cardinal de E et en déduire que B est vrai dans J.
   (b) En déduire que B est vraie dans I
3. On se place dans un langage avec une constante 0, un symbole de fonction unaire f et un symbole de relation binaire E.

   Montrer que la formule $\forall x, (E(x,x) \wedge E(f(x), f(x)) \Rightarrow E(f(x), x) \vee E(f(f(x)), f(x)) \vee E(f(f(x)), x)$ est vraie dans toute interprétation de moins de 2 éléments. Cette formule est-elle valide en général ?

Exercice 6.5 *Logique monadique*

Les questions de ce problème sont largement indépendantes.

On considère une logique dont la signature est composée de deux prédicats unaires P, Q.

La signature donnée est un cas particulier de signature *monadique* c'est-à-dire avec uniquement des symboles de prédicat unaires et pas de symbole de fonction. Cette logique est décidable et on va justifier dans ce cas particulier que l'on peut toujours éliminer les quantificateurs.

Soit A une formule et I une interprétation du langage dont le domaine est $\mathcal{D}$. A tout élément du domaine $d \in \mathcal{D}$, on associe un couple de booléens qui sont les valeurs de vérité de $P(x)$ et $Q(x)$ dans un environnement dans lequel x a la valeur d, c'est-à-dire le couple $(\texttt{val}(x \mapsto d, P(x)), \texttt{val}(x \mapsto d, Q(x)))$. On note $\tau(d)$ ce couple.

1. Dans cette question on considère une interprétation particulière N dont le domaine est l'ensemble des entiers naturels, dans lequel le prédicat P est interprété par la propriété "être un entier pair" et le prédicat Q par la propriété "être une multiple de 3". Quelles sont les valeurs de $\tau(1)$, $\tau(2)$, $\tau(3)$, $\tau(4)$, $\tau(5)$, $\tau(6)$.
2. Combien de valeurs différentes prend $\tau(d)$ dans l'interprétation N de l'exemple précédent ? Combien de valeurs différentes peut prendre $\tau(d)$ dans le cas d'une interprétation I quelconque (donner un maximum et un minimum) ?
3. On introduit une relation binaire $d \simeq d'$ entre les éléments de $\mathcal{D}$ définie par $d \simeq d'$ si et seulement si $\tau(d) = \tau(d')$, c'est-à-dire que les valeurs de vérité des prédicats sont les mêmes pour d et d'. Il est facile de voir que $d \simeq d'$ est une relation d'équivalence qui a un nombre fini de classes d'équivalence.

   On dit que deux environnements ι et ι' sont équivalents si pour toute variable x, on a $\iota(x) \simeq \iota'(x)$. On note également $\iota \simeq \iota'$ cette équivalence entre environnements.

   Montrer que pour toute formule A et pour n'importe quels environnements ι et ι', si $\iota \simeq \iota'$ alors $\texttt{val}(\iota, A) = \texttt{val}(\iota', A)$ (on pourra raisonner par récurrence sur la formule A et traiter juste le cas de formules qui ne contiennent que le quantificateur $\forall$, et les connecteurs de négation et de conjonction).
4. On construit à partir de I une nouvelle interprétation I' en choisissant comme domaine un sous-ensemble $\mathcal{D}' \subseteq \mathcal{D}$ formé exactement d'un élément de $\mathcal{D}$ dans chaque classe d'équivalence. C'est-à-dire que
   — pour tout $d \in \mathcal{D}$, il existe $d' \in \mathcal{D}'$ tel que $d \simeq d'$
   — pour tout $x, y \in \mathcal{D}'$, si $x \simeq y$ alors $x = y$

   (a) Justifier le fait que $\mathcal{D}'$ est un ensemble fini et donner une borne M sur sa taille.
   (b) Construire en suivant cette méthode une interprétation N' pour l'interprétation N de la question 1 (on indiquera le domaine et l'interprétation des deux prédicats sur ces valeurs).
   (c) Montrer que pour tout environnement ι de domaine $\mathcal{D}$, il existe un environnement ι' sur le domaine $\mathcal{D}'$ tel que $\iota \simeq \iota'$.
   (d) Montrer dans le cas général que pour n'importe quelle formule A et n'importe quel environnement ι' sur le domaine $\mathcal{D}'$ de l'interprétation I', la valeur de A dans l'environnement ι' est la même dans l'interprétation I et dans l'interprétation I', c'est-à-dire $\texttt{val}_I(\iota', A) = \texttt{val}_{I'}(\iota', A)$.
5. On introduit M nouvelles constantes $a_1, \ldots, a_M$ dans la signature.
   (a) Expliquer comment on peut transformer toute formule close A en une formule close A_M sans quantificateur telle que A est valide si et seulement si A_M est valide.
   (b) Appliquer votre méthode aux formules $(\forall x, P(x)) \Rightarrow (\exists x, P(x))$ et $(\exists x, P(x)) \Rightarrow (\forall x, P(x))$. Dire si les formules obtenues sont ou non valides.
6. En déduire une méthode pour décider de la validité d'une formule dans le langage.