INFO501 : logique (et informatique)
TD 2 : logique du premier ordre

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 17, poste : 94 22
email : `Pierre.Hyvernat@univ-smb.fr`
www : `http://www.lama.univ-smb.fr/~hyvernat/`

Exercice 1 : formules

Question 1. Donnez une formules du premier ordre (en introduisant si besoin des fonctions ou des relations) pour les phrases suivantes.

N'oubliez pas de typer explicitement vos quantificateurs si vous estimez que c'est nécessaire.

- "Tout nombre entier peut s'écrire comme la somme de 4 carrés."
- "Le carré d'une somme de deux nombres est toujours égal à la somme des carrés augmentée de leur double produit."
- "Toute chaine de caractères non vide possède une dernier élément."
- "Toute liste non vide possède une premier élément."
- "Tous les oiseaux ne volent pas."
- "Tous les nombres premiers sont impairs, sauf 2."
- "Entre 2 nombres réels, on peut toujours trouver un rationnel."
- "Pour n suffisamment grand, 1.0001^n est plus grand que $n^{10\,000}$."

Question 2. Donnez une formule arithmétique pour le prédicat "être impair" et une autre pour le prédicat "être premier". (Vous n'avez droit qu'aux constantes entières, aux fonctions $+$ et $\times$, et à l'égalité.)

Question 3. On utilise souvent des abréviations comme "$\forall \varepsilon > 0, \dots$", "pour tout nombre premier, ...", "$\exists n < p, \dots$".

Précisez le sens de ces abréviations en donnant des formules correspondantes dans le langage donné en cours.

Question 4. On utilise parfois le quantificateur $\exists^! x, P(x)$ pour "il existe un *unique* x qui vérifie $P(x)$.

Comment peut-on exprimer ce quantificateur en logique du premier ordre ?

Question 5. Donnez une formule du premier ordre pour exprimer :

- "la variable `m` est l'élément minimal parmi `T[0]`, `T[1]`, ... `T[i]`",
- "les i premiers éléments du tableau `T` sont triés dans l'ordre croissant".

Question 6. L'arithmétique de Peano est un langage pour l'arithmétique avec l'égalité, la constante 0 et la fonction unaire s (pour "successeur" : $s(n)$ représente l'entier suivant n, càd $n+1$).

Donnez une formule (un axiome) qui exprime qu'on peut utiliser la récurrence pour démontrer la formule $\forall n, \varphi(n)$: *"pour prouver que φ est vraie pour tous les entiers, il suffit de prouver qu'elle est vraie pour* 0*, et que lorsqu'elle est vraie pour* n*, elle est vraie pour* $n+1$*."*

Question 7. La théorie des ensembles de Zermelo-Fraenkel utilise l'égalité et une relation binaire d'appartenance : "$\_ \in \_$".

Un des axiomes est que pour toute paire d'ensembles x et y, il existe un ensemble u qui est la *réunion* de x et y : cet ensemble contient exactement les éléments de x et de y.

Exprimez cet axiome comme une formule du premier ordre.

Plus difficile, le vrai axiome assure l'existence de la réunion de n'importe quel ensemble d'ensembles : pour tout ensemble X, il existe un ensemble U dont les éléments sont exactement les éléments des éléments de X. Si la formule précédente était trop simple, essayez celle là...

Question 8. Toujours en théorie des ensembles de Zermelo-Fraenkel, l'axiome des parties affirme l'existence de l'ensemble des parties d'un ensemble. Exprimez ceci par une formule.

Question 9. Lors d'une modélisation en logique du premier ordre, on peut parfois choisir entre introduire une relation ou une fonction.
- est-ce que la relation `etre_le_pere_de` peut être remplacée par une fonction ?
- est-ce que la relation `etre_la_soeur_de` peut être remplacée par une fonction ?

On suppose qu'on a également un prédicat `humain`.

Comment pourrait on modéliser la phrase "le père d'un être humain est un être humain" ?

Même question pour "la soeur d'un être humain est un humain" ?

Question 10. Donnez l'arbre des formules suivantes. Identifiez les variables liées (et leurs quantificateurs) ainsi que les variables libres.
- $\forall x, \big(P(x) \Rightarrow Q(y)\big)$
- $\forall x, \big(P(x) \Rightarrow Q(x)\big)$
- $\exists x, P(x) \Rightarrow Q(x)$
- $\big(\exists x, P(x) \land \forall x, Q(x) \Rightarrow \exists x, R(x)\big) \land S(x)$

Renommez les variables liées pour éviter que des variables différentes portent le même nom.

Question 11. Faites les substitutions suivantes
- $f(x) = 0$, avec $x := f(y)$
- $\exists x, f(x) = y$, avec $y := f(x)$
- $t > 0 \Rightarrow \exists n, \forall s, \pi_t(s) > n$, avec $n := t + 1$
- $t > 0 \Rightarrow \forall s, \pi_t(s) > n$, avec $n := t + 1$
- $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$, avec $n := i + j$

Exercice 2 : égalité

Question 1. Montrez que l'on peut déduire la symétrie ("si $u = v$ alors $v = u$") à partir de la réflexivité ("$u = u$") et de la substitutivité ("si $u = v$, alors $F[x := u]$ est équivalente à $F[x := v]$").

Question 2. Montrez maintenant que l'on peut déduire la transitivité.

Exercice 3 : validité, modèles

Question 1. Que pensez vous de la formule $\big(\forall x, F(x)\big) \Rightarrow \big(\exists x, F(x)\big)$?

Question 2. Modélisez la phrase suivante "Dans la promo de L3, il y a un étudiant qui, s'il réussit le cours de logique, tous les étudiants réussissent."

Est-ce que la formule correspondante est valide ?

Question 3. Donnez un exemple de formule de la forme $\forall x, P(x)$ qui soit valide, mais où $\neg P(y)$ est valide *lorsqu'on fait le renommage $x := y$ sans renommer les variables liées.*

Question 4. Toute formule est équivalente à une formule sous forme *prénexe*, càd, où tous les quantificateurs sont au début de la formule :

$$\forall x, \exists y, \ldots, \forall z, \qquad \underbrace{F(x, y, \ldots, z)}_{\text{aucun quantificateur}}$$

Donnez un algorithme qui transforme une formule quelconque en formule prénexe.

Indice, vous pourrez utiliser l'équivalence

$$F \vee \forall x, G(x) \quad \Leftrightarrow \quad \forall x, \big(F \vee G(x)\big)$$

si x n'est pas libre dans F.

Utilisez votre méthode pour donner une formule prénexe équivalente à

$$\Big(\exists x, \exists x', f(x) < 0 \wedge f(x') > 0\Big) \Rightarrow \exists x, f(x) = 0$$

Question 5. Que pensez vous de la modélisation suivante pour la phrase "Richard a deux frères : John et Geoffrey" :

$$\texttt{frere}(\texttt{Richard}, \texttt{John}) \wedge \texttt{frere}(\texttt{Richard}, \texttt{Geoffrey})$$

où :
- `frere` est une relation binaire (qui satisfait des axiomes non spécifiés),
- `Richard`, `John` et `Geoffrey` sont des constantes introduite pour l'occasion.

Question 6. Donnez quelques exemples de formule vérifiant la propriété F n'est pas valide et $\neg F$ n'est pas valide.

Précisez bien quels sont vos fonctions et relations, et donnez des modèles pour justifier.

Question 7. Donnez une formule qui signifie "le modèle a au plus 2 éléments".

Idem pour "au plus 3", "au plus 4", etc.

Cherchez une formule du premier ordre qui signifie "le modèle a un nombre fini d'éléments".

Exercice 4 : déduction naturelle

Rappels : voici les règles de la déduction naturelle pour la logique du premier ordre.

$$\frac{\Gamma \vdash F(x)}{\Gamma \vdash \forall x, F(x)}\,(\forall_c) \qquad \frac{\Gamma \vdash \forall x, F(x)}{\Gamma \vdash F(u)}\,(\text{spécialisation})$$

$$\frac{\Gamma \vdash F(u)}{\Gamma \vdash \exists x, F(x)}\,(\exists_c) \qquad \frac{\Gamma \vdash \exists x, F(x) \qquad \Gamma; F(x_0) \vdash C}{\Gamma \vdash C}\,(\text{utilisation du } \exists)$$

avec
- dans la règle $(\forall_c)$, x est une variable *qui n'est pas libre dans le séquent*,
- dans les règle (spécialisation) et $(\exists_c)$, u est un terme arbitraire,
- dans la règle (utilisation du $\exists$), x_0 est une variable *qui n'est pas libre dans le séquent.*

Question 1. Faites une preuve de la formule

$$\Big(\forall x, \big(P(x) \wedge Q(x)\big)\Big) \quad \Rightarrow \quad \Big(\big(\forall x, P(x)\big) \wedge \big(\forall x, Q(x)\big)\Big)$$

puis de l'implication inverse.

Question 2. Plus difficile, faites une preuve de la formule

$$\Big(\big(\exists x, P(x)\big) \vee \big(\exists x, Q(x)\big)\Big) \quad \Rightarrow \quad \Big(\exists x, \big(P(x) \vee Q(x)\big)\Big)$$

puis de l'implication inverse.

Question 3. Démontrez l'implication suivante :

$$\big(\exists x, \neg\varphi(x)\big) \Rightarrow \neg\big(\forall x, \varphi(x)\big)$$

Question 4. Nettement plus difficile, montrez l'implication inverse :

$$\neg\big(\forall x, \varphi(x)\big) \Rightarrow \big(\exists x, \neg\varphi(x)\big)$$

Exercice 5 : unification, programmation logique

Question 1. Appliquez l'algorithme vu en cours pour unifier les expressions suivantes.
- $f(g(x), y)$ et $f(z, y)$
- $f(g(x, y))$ et $f(g(a, f(x)))$
- $f(g(x, y))$ et $f(y)$
- $g(x, f(y))$ et $g(f(z), h(z))$

Question 2. Appliquez l'algorithme de résolution de Prolog (SLD) sur les exemples suivants

```
%%% exemple 1
aime(juliette, pizza).                          %(1)
aime(romeo, Truc) :- aime(juliette, Truc).      %(2)
-------------------------------------------
?- aime(romeo, pizza).

%%% exemple 2
aime(romeo, anchois).                                         %(1)
aime(juliette, pizza(anchois)).                               %(2)
aime(romeo, Truc) :- nourriture(Truc), aime(juliette, Truc).  %(3)
nourriture(pizza(X)) :- nourriture(X).                        %(4)
nourriture(X) :- aime(romeo, X).                              %(5)
-------------------------------------------
?- aime(romeo, pizza(anchois)).

%%% exemple 3
% mêmes règles que l'exemple 2
-------------------------------------------
 ?- aime(romeo, pizza(crevettes)).
```

Question 3. Que pensez vous de la modélisation suivante pour la phrase “Richard a deux frères : John et Geoffrey” en Prolog

```
frere(richard, john).
frere(richard, geoffrey).
```

où :
- `frere` est une relation binaire (qui satisfait des axiomes non spécifiés),
- `richard`, `john` et `geoffrey` sont des constantes introduite pour l'occasion.