Faculté des Sciences Économiques et de Gestion de Sfax
Année universitaire: 2014-2015
Module: Méthodes Formelles
Enseignant: Mohamed TOUNSI
Auditoire: 1 année Master - SINT

Travaux Dirigés - 2

Exercice 1

Trouver les occurrences libres et les occurrences liées des variables dans les formules suivantes:

1. $\exists x\ (logicien(x)\ \wedge\ etudiant(x))$
2. $(\exists x\ logicien(x))\ \wedge\ etudiant(x)$

Exercice 2

Un syllogisme d'Aristote fait généralement intervenir des propriétés P, Q des individus et des affirmations telles que:

1. Tous les P sont des Q
2. Certains P sont des Q
3. Aucun P n'est un Q
4. Certains P ne sont pas des Q

Écrivez des formules de la logique du 1er ordre qui retranscrivent ces affirmations.

Exercice 3

1. Remplir la table de vérité suivante:

n	$n \in \mathbb{N}$	$n \in \mathbb{N} \Rightarrow P(n)$	$n \in \mathbb{N} \wedge P(n)$
0			
1			
2			
3			
$\vdots$			
a			
b			
cde			
+			
!			
$\vdots$			

2. À partir de la table ci-dessus, indiquez la différence entre les deux premiers énoncés ci-dessous. De même avec les deux derniers.

   (a) $\exists n.\ (n \in \mathbb{N} \Rightarrow P(n))$
   (b) $\exists n.\ (n \in \mathbb{N} \wedge P(n))$
   (c) $\forall n.\ (n \in \mathbb{N} \Rightarrow P(n))$
   (d) $\forall n.\ (n \in \mathbb{N} \wedge P(n))$

Exercice 4

Définissez les prédicats suivants en logique du premier ordre. On fera l'hypothèse que `x`, `y` et `z` sont des variables de type entier.

1. `pair(x)` qui est vrai si `x` est un nombre pair
2. `div(x,y)` qui est vrai si `x` est un diviseur de `y`
3. `mulpair(x,y)` qui est vrai si `x` est un multiple pair de `y`
4. `pgcd(x,y,z)` qui est vrai si `x` est le PGCD de `y` et `z`
5. `prem(x)` qui est vrai si `x` est un nombre premier (c'est-à-dire divisible par aucun nombre plus petit que lui excepté 1)

Exercice 5

Définissez les prédicats suivants en logique du premier ordre, où `t` est un tableau, `x` est un élément et `i`, `j` sont des variables entières. On supposera l'existence d'un ordre total $\leq$ sur les éléments de `t`.

1. `croissant(t)` qui est vrai si `t` est un tableau trié par ordre croissant. On note `t[i]` l'élément `i` de `t` et `taille(t)` le nombre d'élément de `t`.
2. `notin(x,t,i,j)` qui est vrai si `x` n'appartient pas à la partie du tableau `t` allant des indices `i` à `j`
3. `sup(x,t)` qui est vrai si `x` est supérieur aux éléments de `t`
4. `max(x,t)` qui est vrai si `x` est l'élément maximal du tableau `t`