

Modélisation en logique des prédicats

Dans ces exercices, on ne s'intéresse pas à *prouver* des choses, mais seulement à les *exprimer* correctement par des formules logiques bien formées.

Exercice 1. Dans cet exercice, on considère un unique type de base H , dont les membres désignent des individus. On dispose de deux prédicats $\text{ami}(x, y)$ et $\text{ennemi}(x, y)$ d'arité 2 sur H , qui énoncent respectivement que x considère y comme un ami ou un ennemi ; on utilise également une constante d'objet moi , qui désigne la personne qui parle. Les relations d'amitié et d'inimitié ne sont pas a priori supposées symétriques.

Interpréter en langage courant les formules suivantes :

- $\forall x : H, \sim \text{ami}(\text{moi}, x)$
- $\forall x, y : H, \text{ami}(x, y) \rightarrow \text{ami}(y, x)$
- $\sim(\exists x, y : H, \text{ami}(x, y) \wedge \text{ennemi}(x, y))$
- $\forall x : H, \sim \text{ami}(\text{moi}, x) \rightarrow \text{ennemi}(\text{moi}, x)$
- $\forall x : H, \text{ennemi}(x, x)$
- $\exists x : H, \forall y : H, \sim \text{ami}(x, y)$
- $\forall y : H, \exists x : H, \sim \text{ami}(x, y)$
- $\forall x : H, \exists y : H, \text{ennemi}(x, y)$
- $(\forall x, y : H, \text{ami}(x, y)) \rightarrow (\exists z : H, \text{ami}(\text{moi}, z))$
- $\forall x, y : H, \text{ami}(x, y) \rightarrow \exists z : H, \text{ennemi}(x, z) \wedge \text{ennemi}(y, z)$

Exercice 2. On reprend le cadre précédent ; on suppose de plus qu'on dispose du symbole d'égalité (vu comme un prédicat sur les paires d'individus, mais qu'on notera sous forme infixe $x = y$ au lieu de préfixe $=(x, y)$). Traduire sous forme de formule logique chacune des affirmations suivantes :

- Les ennemis de mes amis sont mes ennemis.
- Tout le monde a des ennemis. (À prendre au sens de "au moins un ennemi")
- Personne n'a que des amis (personne n'est l'ami de tout le monde).
- Les amis de mes amis ne sont pas forcément mes amis.
- J'ai un et un seul ami. (La difficulté est de formuler le "un seul" ; il faut utiliser l'égalité)

Exercice 3. Dans cet exercice, les objets sont des entiers naturels (type nat). On suppose qu'on dispose de deux symboles de fonction somme et prod, de type $\text{nat} \times \text{nat} \rightarrow \text{nat}$, et qui désignent respectivement les opérations de somme et de produit ; ainsi que d'un prédicat $\text{inferieur}(n, m)$ qui exprime que n est (strictement) inférieur à m . On pourra utiliser les notations usuelles, par exemple

écrire $a \times b < c$ au lieu de *inferieur*(produit(a, b), c). On s'autorise à utiliser des constantes pour les entiers sous forme décimale (0, 1, etc.).

- Écrire une formule pour définir un prédicat *divise*(n, m), qui exprime que n est un diviseur de m .
- Écrire une formule dépendant d'un entier n et qui exprime que, si cet entier est divisible séparément par deux entiers, il est divisible par leur produit. (Affirmation qui est en général fausse; on s'intéresse ici à l'expression de la propriété)
- Un entier n est premier s'il n'est égal ni à 0 ni à 1, et qu'il a pour seuls diviseurs 1 et n . Écrire une formule pour un prédicat *premier*(n) qui exprime que n est premier.
- Deux entiers sont premiers entre eux si leur seul diviseur commun est 1. Écrire une formule pour un prédicat *premiersRelatifs*(n, m) qui exprime que n et m sont premiers entre eux.
- Écrire une formule qui exprime le fait qu'il existe une infinité d'entiers premiers (trouver un moyen détourné d'exprimer cette notion d'infini!)

Exercice 4. On considère deux types D (dont les membres sont des programmeurs) et L (dont les membres sont des langages de programmation). On dispose d'un prédicat C sur $D \times D$ ($C(x, y)$ signifiant que le programmeur x connaît le programmeur y) et d'un prédicat P sur $D \times L$ ($P(x, \ell)$ signifiant que le programmeur x pratique le langage ℓ), ainsi que deux prédicats $=_D$ et $=_L$ sur $D \times D$ et $L \times L$ respectivement, qu'on interprète comme exprimant l'égalité sur D et sur L respectivement. Enfin, on a une fonction $F : D \rightarrow L$, qu'on interprète comme suit : $F(x)$ désigne le langage favori du programmeur x .

Traduire en formule les affirmations suivantes :

- Tout programmeur pratique au moins un langage de programmation.
- Tout langage est pratiqué par au moins un programmeur.
- Il existe un langage qui n'est le favori d'aucun programmeur.
- Il existe un programmeur qui ne pratique pas son langage favori.
- Il existe un programmeur qui ne pratique que son langage favori.
- Il existe un programmeur qui pratique tous les langages.
- Il existe un programmeur qui pratique tous les langages, sauf son langage favori.
- Tout programmeur a parmi ses connaissances un programmeur qui pratique chaque langage. (Attention, ambiguïté possible)
- Il existe un programmeur qui ne pratique aucun langage que pratique une de ses connaissances.
- Quand on connaît un programmeur, on les connaît tous.
- Il existe un langage dont tous les praticiens se connaissent entre eux.