

Outils formels de Modélisation

11^{ème} séance d'exercices

Dimitri Racordon

08.12.17

Dans cette séance d'exercices, nous allons étudier la logique du premier ordre. Nous tâcherons notamment de traduire des phrases d'un domaine externe à la logique en formule du premier ordre, et inversement.

1 logic.translate.com?fr-fol (★)

Traduisez les phrases suivantes en formules de logique du premier ordre. Créez autant de prédicats et/ou de formules que nécessaire.

1. Il y a un assistant qui n'est pas une femme.
2. Tous les étudiants sont soit des hommes, soit des femmes.
3. Tous les étudiants qui étudient les méthodes formelles sont intelligents.
4. Aucun étudiant n'est meilleur que tous les autres étudiants.

2 logic.translate.com?fol-fr (★★)

Traduisez les formules suivantes en phrases.

1. $\forall a, (Homme(a) \implies Barbe(a)) \wedge (Femme(a) \implies \neg Barbe(a))$
2. $\exists a, \exists b, \exists c, Soeur(a, b) \wedge Soeur(b, c) \wedge Soeur(c, a)$
3. $\forall x, \forall y, BelleSoeur(x, y) \implies \exists z, Femme(x) \wedge Epouse(y, z) \wedge Soeur(x, z)$
4. $\forall x, \forall y, Enfant(x) \wedge Pokemon(y) \implies Aime(x, y)$

3 Un peu de Swift (★★★)

La logique du premier ordre permet d'exprimer des formules sur des ensembles de tailles arbitraires, éventuellement infinies, par l'utilisation du quantificateur existentiel (\exists) et universel (\forall). En Swift, on peut traduire cela par l'utilisation de prédicats sur des séquences:

```
func isWomanName(name: String) -> Bool {
    switch name {
    case "Aline" : return true
    case "Cynthia": return true
    default      : return false
    }
}

let people = ["Aline", "Bernard", "Cynthia"]
let thereAreWoman = people.contains(isWoman)
let thereAreOnlyWoman = people.filter(isWoman).
    count == people.count
```

En partant de ce principe, écrivez en Swift les formules de logique des exercices 1 et 2.