



**SORBONNE
UNIVERSITÉ**

Projet LRC

Logique et représentation des connaissances

Subsumptions en Prolog

Antoine Théologien & Gabriel Zamy

Décembre 2024

Introduction

Ce rapport est consacré au projet de LRC portant sur les subsomptions en prolog. Nous y apporterons les réponses aux différents exercices, principalement aux questions mises en commentaires.

Exercice 1

subs(chat, felin)

- Traduction : Les chats sont des félins
- FL :

$$\text{chat} \sqsubseteq \text{felin}$$

subs(chihuahua, and(chien, pet))

- Traduction : Un chihuahua est à la fois un chien et un animal de compagnie.
- FL :

$$\text{chihuahua} \sqsubseteq (\text{chien} \sqcap \text{pet})$$

subs(and(animal, some(aMaitre)), pet)

- Traduction : Un animal qui a un maître est un animal de compagnie.
- FL :

$$(\text{animal} \sqcap \exists \text{aMaitre}) \sqsubseteq \text{pet}$$

subs(some(aMaitre), all(aMaitre, personne))

- Traduction : Toute entité qui a un maître ne peut avoir qu'un (ou plusieurs) maître(s) humain(s)
- FL :

$$\exists \text{aMaitre} \sqsubseteq \forall \text{aMaitre}.\text{personne}$$

subs(and(all(mange, nothing), some(mange)), nothing)

- Traduction : On ne peut pas à la fois ne rien manger (ne manger que des choses qui n'existent pas) et manger quelque chose
- FL :

$$(\forall \text{mange}.\text{nothing} \sqcap \exists \text{mange}) \sqsubseteq \text{nothing}$$

equiv(carnivoreExc, all(mange, animal))

- Traduction : Un carnivore exclusif est défini comme une entité qui mange uniquement des animaux
- FL :

$$\text{carnivoreExc} \equiv \forall \text{mange}.\text{animal}$$

Exercice 2

1.

subsS1(C, C).

Traduction : Tout concept C est subsumé par lui-même ($C \sqsubseteq_s C$).

subsS1(C, D) :- subs(C, D), C \== D.

Traduction : Si $C \sqsubseteq D$ (c'est-à-dire que $\text{subs}(C, D)$ est explicitement dans la TBox) et que $C \neq D$, alors $C \sqsubseteq_s D$.

subsS1(C, D) :- subs(C, E), subsS1(E, D).

Traduction : Si $C \sqsubseteq E$ et $E \sqsubseteq_s D$, alors $C \sqsubseteq_s D$. (Cette règle exploite la transitivité.)

Pour les requêtes $\text{canari} \sqsubseteq_s \text{animal}$ et $\text{chat} \sqsubseteq_s \text{etreVivant}$, on obtient *true*, *true*, et enfin *false* qui correspond à la fin du parcours de l'arbre.

2.

En utilisant le mode *trace*, on remarque que la requête $\text{chien} \sqsubseteq_s \text{souris}$ semble tomber dans une boucle infinie, qui pourrait être causée par le fait que l'on ne vérifie pas si une subsumption a déjà été testée ou non.

3.

On teste donc les requêtes suivantes :

- Pour $\text{chat} \sqsubseteq_s \text{etreVivant}$, on attend le résultat *true*, on obtient bien *true*.
- Pour $\text{chien} \sqsubseteq_s \text{canide}$, on attend le résultat *true*, on obtient bien *true*.
- Pour $\text{chien} \sqsubseteq_s \text{chien}$, on attend le résultat *true*, on obtient bien *true*.
- Pour $\text{chien} \sqsubseteq_s \text{souris}$, on attend donc *false*, que l'on obtient bien en exécutant la requête. En observant la trace, on remarque qu'à chaque exploration, le concept depuis lequel l'arbre est exploré est bien ajouté dans la liste des concepts visités. Cela permet d'éviter les boucles infinies que nous avons précédemment. Comme aucune règle ne permet de vérifier la requête, on obtient bien le résultat *false*.

4.

En testant la requête $\text{souris} \sqsubseteq_s \exists \text{mange}$, on obtient bien *true*. Cela fonctionne car la règle $\text{subsS1}(C, D) : -\text{subs}(C, D), C \equiv D$. n'oblige pas D à être un concept atomique. Ainsi, comme on a la règle : $\text{subs}(\text{animal}, \text{some}(\text{mange}))$, et qu'une souris est bien un animal, on obtient *true*.

5.

- Pour $\text{chat} \sqsubseteq_s X$, on devrait obtenir :
 - $X = \text{chat}$;
 - $X = \text{felin}$;
 - $X = \text{mammifer}$;
 - $X = \text{animal}$;
 - $X = \text{etreVivant}$;
 - $X = \text{some}(\text{mange})$;
 - *false*;

En testant, on obtient bien ce résultat.

- Pour $X \sqsubseteq_s \text{mammifere}$, on devrait obtenir :

- $X = \text{felin}$;
- $X = \text{canide}$;
- $X = \text{souris}$;
- $X = \text{chat}$;
- $X = \text{chien}$;
- $X = \text{lion}$;
- false ;

En testant, on obtient bien ce résultat.

6.

On travaille donc avec la requête suivante : $\text{lion} \sqsubseteq_s \forall \text{mange}.\text{animal}$. Avant l'ajout, on obtient *false*.

Après l'ajout des règles dans le fichier prolog, on obtient donc *true*, puisque les subsomptions sont maintenant définies grâce aux équivalences.

Exercice 3

1.

Les trois requêtes retournent toutes *true*.