# Introduction

La représentation des connaissances en utilisant des modèles logiques est largement répandue, toutefois, elle rencontre certaines difficultés reliées soit au temps de décision exponentiel pour les systèmes décidables, soit à la semi ou non décidabilité d’autres logiques. Ainsi, il s’agissait d’améliorer l’expressivité de la représentation ; la rendre plus facile à lire. Ce que font les réseaux sémantiques.

*Un réseau sémantique est un graphe orienté et étiqueté, une sémantique est associée à une relation.*

Dans ce TP nous allons nous intéresser à l’exploitation des réseaux sémantiques grâce à un algorithme de propagation de marqueurs. Nous allons le présenter puis le modéliser et l’implémenter pour pouvoir le tester sur certains exemples.

# Exploitation des réseaux sémantiques et algorithme de propagation de marqueurs

Un réseau sémantique peut être considéré comme une conjonction de formules, dance ce cas, les mêmes méthodes que dans les modèles logiques sont utilisées. Il peut également être considéré comme un graphe, des techniques de propagations sont alors appliquées et c’est ce à quoi nous nous intéressons.

L’algorithme de propagation de marqueurs permet de répondre à une question posée au réseau sémantique. L’algorithme extrait les nœuds pertinents qui peuvent mener à une ou plusieurs réponses. Ensuite, il effectue une propagation de telle sorte à marquer les nœuds en suivant le sens contraire de la relation « is-a » qui représente l’héritage. Le même processus est répété jusqu’à ce qu’il n’y ait plus de propagation possible. Au cours de cette propagation, l’algorithme cherche à trouver une réponse à la question qui lui est posée. Il saura qu’une réponse est obtenue lorsque des nœuds communicants avec la relation exprimée dans la question sont marqués de part et d’autre.

Il est à noter que les différents liens des réseaux sémantiques sont gérés par cet algorithme, notamment les liens d’exceptions.

# Modélisation

Pour modéliser un réseau sémantique nous avons choisis d’implémenter 3 différentes classes :

## Classe

La principale étant la classe « Classe » qui représente un nœud quelconque dans le réseau celui-ci est composé de 3 attributs principaux

**private** String nom;

**private** ArrayList<Lien> liens;

**private** Marquage marquage;

Le nom de la classe (toujours en majuscule), un ensemble de liens et enfin l’attribut marquage que nous verrons par la suite, cet ensemble de lien représente les arcs reliés à la classe « nom » (par souci d’optimisation nous avons décidé de ne représenter que les liens entrant d’une classe c)

## Lien

**private** String libelle;

**private** String source;

**private** String type;

Dans un réseau sémantique un lien est porteur d’une étiquette, l’attribut « labelle » (toujours en minuscule pour nous) représente cette dernière dans notre modélisation, comme nous l’avons précisé plus haut nous n’avons représenté que les liens entrant par rapport à une classe ainsi la chaine « source » nous donne l’information d’où proviens ce lien (cet attribut est le nom d’une autre classe bien évidemment). Enfin vu que nous pouvons avoir plusieurs types de lien comme par exemple *sorte de strict, non sorte de strict* ….. nous avons ajouté l’attribut « type » pour sauvegarder cette information.

Lors de l’affichage du réseau chaque type de lien aura une représentation de sa flèche

Sorte de non strict <|— libellé —

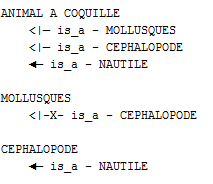
Sorte de strict ◄— libellé —

Exception ◄-- libellé —

Non sorte de strict◄-x- libellé —

Non sorte de non strict <|-x- libellé —

Normal <— libellé —



**Figure 01 : Fichier représentant le réseau Céphalopodes.**

## Marquage

**private** **boolean** marquage;

**private** String type;

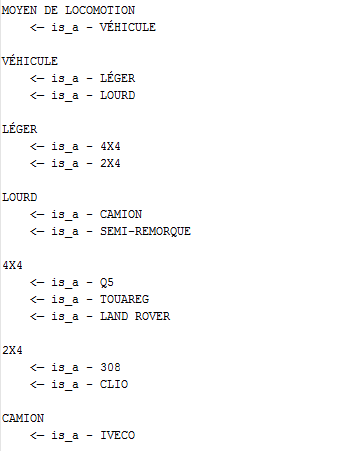
Nous arrivons à la classe Marquage, nous avons décidé d’implémenter marquage au lieu d’un simple booléen pour caractériser un marquage par un nom ou type (exemple : M1) et pouvoir gérer le cas de plusieurs marquages.

Ainsi grâce à cette classe nous pouvons facilement savoir par quelle marquage un nœud donné est marqué (M1 ou M2 ou ….) en utilisant la méthode « by() ».

Ou encore tester si un nœud N est marqué par un marquage précis (exemple: MarkedBy(“M1“) renvoie true si le nœud est marqué par M1).

# Classe principale

Principalement le réseau est décrit dans la classe ReseauSementique celui-ci n’a pour attribut qu’un ensemble de nœuds (classes), les classes constituant le réseau. La classe contient des méthodes pour l’affichage est l’écriture du réseau dans un fichier



**Figure 02 : Représentation du réseau « moyen de locomotion »**

Comme précisé plus haut les noms de classes sont en majuscule les libellés de lien en minuscule, et les liens sont représentés selon une hiérarchie et toujours entrant

(pas de sens ->)

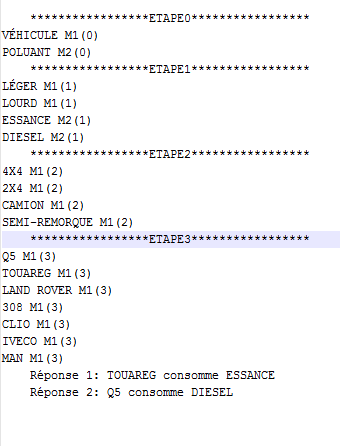
Egalement La méthode de propagation, celle-ci prend en arguments 3 paramètres, la source, le lien et la destination, ainsi par exemple pour poser la question « Quels sont les véhicules qui consomment du carburant poluant » nous donnerons en paramètre :

source = véhicule

lien = consomme

destination = poluant

L’exécution de la propagation nous donnera un fichier détaillant tout le déroulement de l’algorithmes en précisant les étapes et à chaque étape nous obtiendrons les résultats trouvé pour cette itération.



**Figure 03 : Illustration des étapes de la propagation sur le réseau véhicules.**

# Conclusion

Nous concluons ce travail en énonçant les forces et les faiblesses des réseaux sémantiques. Les réseaux sémantiques possèdent des mécanismes qui permettent de structurer la base de connaissances. Nous citons : la généralisation, la partition et l’agrégation. De plus, leur forme graphique facilite la compréhension en augmentant l’expressivité. Le principal inconvénient des réseaux sémantiques est qu’ils génèrent une explosion combinatoire dès que la taille du graphe devient assez grande. Ainsi, nous sommes en présence de certains avantages que l’on peut exploiter et d’inconvénients que l’on doit prendre en considération lorsque nous voulons utiliser les réseaux sémantiques.