

TD 5 – recap

1 Logique / résolution

Modélisez en logique du premier ordre les assertions suivantes :

1. Les personnes qui ont la grippe doivent prendre du Tamiflu.
2. Les personnes qui ont de la fièvre et qui toussent ont la grippe.
3. Ceux qui ont une température supérieure à 38 degré ont de la fièvre.
4. Pierre tousse et a une température supérieure à 38 degré
5. Pierre doit prendre du Tamiflu

Modélisez les mêmes assertions en logique propositionnelle, et démontrez, en utilisant le méthode de résolution, que la dernière assertion (5.) est une conséquence logique des autres.

2 Représentation à base de graphes

Modélisez les assertions suivantes à l'aide d'un réseaux sémantique et listez les inférences possibles:

1. Les ordinateurs sont des appareils électroniques
2. Les téléphones sont des appareils électroniques
3. Les ordinateurs portables sont des ordinateurs
4. Les smartphones sont des téléphones et des ordinateurs
5. Les ordinateurs ont un CPU et de la mémoire
6. Les ordinateurs portables ont un écran et un clavier
7. Les smartphones ont un écran
8. Mathieu a un ordinateur portable de marque Dell
9. Mathieu a un smartphone de marque Samsung
10. Mary a un smartphone de marque Sansung

Modélisez maintenant ces assertions par une graphe conceptuel et dessinez le graphe conceptuel correspondant à la requête suivante (et listez les résultats) :

Les appareils électroniques avec un clavier et appartenant à Mathieu.

En supposant que le graphe est aussi représenté en RDF, écrivez maintenant la requête ci-dessus en SPARQL.

3 Programmation Logique Inductive

On souhaite trouver les règles qui permettent de reconnaître que quelqu'un est le frère ou la soeur de quelqu'un d'autre à l'aide de la programmation logique inductive.

1. Indiquez les prédicats cibles (entête de règles, `modelh` en Aleph), le type de leur paramètres et si les paramètres seront des variables ou des constantes dans les règles.
2. Indiquez les prédicats utilisables dans le corps des règles (`modelb` en Aleph), le type de leur paramètres et si ceux ci seront des variables ou des constantes dans les règles.
3. Donnez plusieurs exemples pour les connaissances de base, les exemples positifs et les exemples négatifs.
4. Pour un exemple positif, essayez de trouver la saturation (la règle la plus spécifique qui spécialise l'exemple) et une généralisation de cette règle qui couvre le plus possible d'autres exemples positifs.

4 Programmation par contraintes

Considérant le tableau ci-dessous, et que l'on a un budget de 14K€, modéliser le problème qui consiste à trouver l'ensemble des investissements réalisables avec le budget qui maximise les gains.

Investissement	1	2	3	4	5	6
Cash requis	5K€	7K€	4K€	3K€	4K€	6K€
Gain	16K€	22K€	12K€	8K€	11K€	19K€

La modélisation utilise une contrainte d'optimisation (maximiser les gains). Comment est il possible de modifier l'algorithme d'anticipation de façon à prendre en compte cette contrainte. Réaliser quelques étapes de cet algorithme pour tester votre idée?

Note : Il est possible d'ajouter ce type de contrainte dans ORTools en utilisant la fonction `model.Maximise(exp)`.