

Acquisition de connaissance 2

TP 2 - Programmation logique inductive

Installation

Récupérez le programme Aleph à l'adresse :

http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph_toc.html

Garder cette page ouverte, c'est le manuel d'utilisation d'Aleph ; lisez-en l'introduction et le « *getting started* ».

Le compilateur YAP nécessaire à l'exécution d'Aleph est installé et disponible sous /usr/local/bin et est nommé yap-5.1.1.

1 - Prise en main - Les trains de Michalsky

Le répertoire exemple-train contient les fichiers pour Aleph correspondant à l'exemple des trains de Michalsky vu en cours.

- 1) Examinez ces fichiers, notamment l'encodage des informations. Identifiez les exemples, le *background knowledge*, la définition du langage d'hypothèse, et les autres paramètres... Reportez vous à la documentation d'Aleph pour comprendre ces différents éléments.
- 2) Lancez le processus d'inférence : tout d'abord construisez la *bottom clause* de l'exemple 1 (commande *sat(1)*), puis lancez la recherche dans l'espace des hypothèses (*reduce*). Qu'en déduisez vous sur l'opérateur de raffinement par défaut d'Aleph ?

2 – Une affaire de famille

Nous avons vu en cours un exemple de l'utilisation de la PLI pour inférer le concept de 'fille de'.

Modéliser le problème pour Aleph, à partir des informations suivantes :

Ann est la mère de Mary et de Tom. Mary est la mère de Rosy. Tom est le père d'Eve, de Lisa et de Bob.

Indiquez quels sont vos exemples positifs, négatifs, les informations du *background knowledge* et la ou les réponses données par Aleph. La réponse correspond-elle à votre propre définition de fille ? Si non, expliquez ce qu'il se passe et corrigez vos données pour obtenir des règles qui correspondent à une définition correcte du concept de « fille de ».

3 - Les figures du Poker

Le but est de faire apprendre à la machine ce qu'est une paire, une suite, un brelan...

Les données brutes sont fournies, ainsi que le squelette d'un petit script Perl (prepar_poker_data.prl) pour les mettre au bon format pour Aleph.

Complétez ce script, modifier les chemins d'accès à Aleph, et compléter le *background knowledge*

(fichier entete.pl) avec la définition des prédicats qui vous semblent utiles pour faire apprendre figure par figure. Réfléchissez bien à ce qui doit servir d'exemples négatifs.
Commencer par faire apprendre ce qu'est un carré. Pourquoi les autres figures sont plus difficiles à faire apprendre ?

À préparer à l'avance : Une des clefs est d'écrire un prédicat `cartes/2` qui *prend* en premier argument un identifiant de main (jeu), et *renvoie* en deuxième argument l'ensemble (l'ordre n'est donc pas important, contrairement à une liste) des cartes contenues dans la main. On veut donc définir le prédicat suivant :

`cartes(+jeu, -ensemble_de_carte)`.

Pour écrire la définition de ce prédicat, on suppose que l'on dispose d'un prédicat `a_carte(jeu4, carte3)` qui indique que la carte `carte3` est dans le jeu `jeu4`.

Notez bien qu'une définition comme celle-ci :

`cartes(Jeu, [A,B,C,D,E]) :- a_carte(Jeu,A), a_carte(Jeu,B), a_carte(Jeu,C),
a_carte(Jeu,D), a_carte(Jeu,E).`

ne convient pas.