

Programmation Logique avec contraintes - TD/TP 3

Le but de ce TD/TP est de se familiariser avec les contraintes sur un domaine fini en utilisant GNU Prolog.

GNU Prolog

GNU Prolog (<http://gnu-prolog.inria.fr/>) est un Prolog avec un solveur intégré de contraintes sur un domaine fini. GNU Prolog est installé sur nivose et se lance avec la commande `gprolog`. Les prédicats principaux additionnels pour les contraintes sur un domaine fini:

- `fd_domain(?Vars, +Integer1, +Integer2)`
Ce prédicat définit le domaine d'une variable `Vars` ou d'une liste de variables `Vars` d'être entre les deux bornes `Integer1` et `Integer2`.
- `fd_domain(?Vars, +ListeValeurs)`
Ce prédicat définit le domaine d'une variable ou d'une liste de variables `Vars` d'être la liste des valeurs dans `ListeValeurs`
- Les contraintes arithmétiques s'écrivent en utilisant les fonctions habituelles et les prédicats suivant: `#=`, `#\=`, `#<`, `#>`, `#<=`, `#>=`
- `fd_all_different(?ListeVars)`
Ce prédicat décrit la contrainte qui impose que toutes les variables de la liste `?ListeVars` prennent des valeurs différentes.
- `fd_labeling(?Vars, [])`
Ce prédicat est utilisé pour rechercher des solutions des contraintes sur les variables `Vars`

Un programme en GNU Prolog pour résoudre une contrainte s'écrit en trois parties: définir les domaines des variables, décrire la contrainte, résoudre. Exemple:

```
probleme([X,Y,Z]) :- fd_domain(X,0,5),
                    fd_domain([Y,Z],3,7),
                    X+Y #< 2*Z,
                    fd_labeling([X,Y,Z], []).
```

Ensuite:

```
| ?- probleme(L).
L = [0,3,3] ? ;
L = [0,3,4] ? ;
L = [0,3,5] ? ;
etc.
```

GNU Prolog utilise la borne consistence. Un peut demander par exemple:

```
| ?- fd_domain(X,0,5), fd_domain([Y,Z],3,7), X+2*Y #< 2*Z - 2.

X = _#3(0..5)
Y = _#25(3..5)
Z = _#47(5..7)
```

Exercice 1 On considère l'addition suivante :

```
SEND
+ MORE
-----
MONEY
```

où chaque lettre représente un chiffre différent (compris entre 0 et 9). On souhaite connaître la valeur de chaque lettre, sachant que la première lettre de chaque mot représente un chiffre différent de 0. Pour modéliser ce problème on peut considérer que le mot `SEND` a comme valeur $1000*S + 100*E + 10*N + D$, etc.

- La variable M peut être contrainte d'avoir la valeur 1 en utilisant la borne consistance. Expliquez comment.
- Donnez un programme en GNU Prolog qui résout ce problème.

Exercice 2

- Rendez bornes consistantes la contrainte

$$X = 2 * Y + Z \wedge X + Y = 6$$

avec domaines $X : [2..7], Y : [1..5], Z : [1..2]$

- Comparez avec le résultat de GNU Prolog.

Exercice 3 Programmer le problème des 6 reines en GNU Prolog.

Exercice 4 On considère la grille suivante :

		9			1	6	2	
5	7			2	8		3	
3			7					4
8	9			7		4		
	6		5		3		9	
		1		9			7	6
6					7			8
	4		1	3			6	5
	2	7	6			9		

Le problème consiste à remplir la grille de sorte que chaque ligne, chaque colonne et chaque carré contiennent les chiffres 1 à 9.

- Écrire un programme en GNU Prolog pour résoudre ce problème.